ACMBP: An Adaptive Graph-Based Meta-Learning Framework for Cross-Modal Behavior Prediction from IoT and Social Media Streams

Xinzhu Pu[1], Yaqi Ren[2], Jing Liang[3*]
[1]School of Artificial Intelligence and Computer Science, North China University of Technology, Beijing 100144, Large-scale heterogeneous digital ecosystems
[2]School of Humanities and Law, North China University of Technology, Beijing 100144, Large-scale heterogeneous digital ecosystems
[3]Network and Educational Technology Center, Jinan University, Guangzhou, 510632
E-mail: pu_xinzhu@126.com, renyaqi616@163.com, ljing@jnu.edu.cn
[*]Corresponding author

Keywords: internet of things, social media user behavior, adaptive cross-modal, graph networks, behavior prediction

Received: September 5, 2025

The growth of Internet of Things (IoT) devices and social media platforms generates large volumes of multimodal behavioral data that present both great opportunities and challenges to cross-domain behavior prediction. The current models do not deal with time misalignment, distributional drift, and personalization when only a limited amount of data is available. To address these issues, Adaptive Cross-Modal Behavioral Prediction (ACMBP), introduced in this study, is a new framework that merges the streams of IoT activity and social media characteristics via adaptive graph networks. ACMBP combines temporal-semantic offset attention to align activities by the IoT with social actions, drift-conscious dynamic graph rewiring to adapt to changing user relations, hierarchical cross-domain transfer to adapt across different platforms, and few-shot personalization with meta-learning and behavioral prototypes. ACMBP possibility in large-scale heterogeneous digital ecosystems is compared to a fused IoT social dataset (wearable sensors + Twitter) and two public multimodal benchmarks against 10 strong baselines (LSTM, GRU, BERT, GCN, GAT, HAN, GraphSAGE, DCRNN, DeepFM, Multimodal Transformer). ACMBP has been tested on a custom IoT–Social Fusion dataset (1,200 users, 90-day collection, 2.3M sensor records, 850K Twitter posts) and two public benchmarks (PAMAP2, USC-HAD). The results of all improvements are statistically significant (Wilcoxon signed-rank test, $p < 0.001$). ACMBP attains Behavioral Transition Accuracy (BTA) of 91.4%, Temporal Offset Prediction Error (TOPE) of 1.2 hours, Drift Detection Precision (DDP) of 84.7%, Cross-Platform Transfer Efficiency (CPTE) of 82.3%, and Few-Shot Adaptation only with five samples, which are improvements by 7.3%, 32%, 11.8%, 13.4%, and 50% respectively over the best baselines. Each module's contribution is supported by ablation studies, and the results of scalability experiments indicate stable performance in large-scale heterogeneous IoT-social environments.

Povzetek: Raziskava predstavlja nov okvir za napovedovanje vedenja, ki združuje podatke IoT naprav in družbenih omrežij ter z naprednimi prilagoditvenimi metodami dosega bistveno boljše rezultate kot obstoječi modeli.

1 Introduction

The rapid growth of Internet of Things (IoT) devices and ubiquitous social media platforms has redefined the way people engage with digital ecosystems in order to produce unprecedented amounts of behavioral data, which are indicators of an intricate human-technology relationship [1]. Users seamlessly navigate between IoT environments and social media platforms, generating complex behavioral patterns across multiple modalities and temporal scales [2]. The IoT devices, which have reached billions around the world, are constantly recording finer-grained behavioral signals such as sensor data, interaction behavior, time-based activity, as well as contextual environmental conditions [14]. At the same time, social media networks are core arenas of emotional expression, socializing, and behavioral expression, accommodating billions of users who produce enormous textual, graphic, and behavioral resources every single day [3]. Cross-domain interactions are strengthened in large-scale heterogeneous digital ecosystems due to the widespread use of smart devices and high engagement on social media. Nevertheless, the existing studies have mostly addressed these fields alone without leveraging the intrinsic cross-modal connections that may make a major contribution to the accuracy of behavioral forecast and the potential of personalizations [4].

Conventional user behavior prediction methods have been mostly single-domain, either with temporal modeling of IoT-specific signals or content analysis of social media [5], [6]. On the other hand, social media behavior prediction is dependent on natural language processing, computer vision, and social network analysis to predict the posting patterns, content preferences, as well as engagement behaviors [7]. Although these unimodal approaches have proved to be successful in their respective fields, they are unable to capture the compound interdependencies of physical and digital behavioral expressions that represent the present-day interactions of the user [8].

The main constraint of current methods is the fact that they do not assess the dynamic, two-sided dependencies between IoT practices and social media practices [9]. Users' physical activities, which are recorded by smart home devices, wearables, and environmental sensors, usually come before and shape their digital manifestations on social networks [10]. As an example, the patterns of IoT activity in the late hours might be associated with the following social media posting patterns, whereas social interactions might reciprocally affect physical activity patterns and IoT device utilization [11]. Such a cross-modal behavioral cross-linkage offers a chance to improve predictive accuracy, yet demands a complex fusion mechanism that can manage temporal correlations, modality distortions, and user-specific trends as well [12].

Recent developments in multimodal learning have shown that it is possible to combine different sources of data to enhance prediction accuracy in a wide range of fields [13], [15], [18]. Nevertheless, there are special problems when it comes to the implementation of multimodal methods in the context of IoT-social behavioral prediction. To begin with, the time lag between IoT processes and social media activity poses difficult alignment challenges that cannot be effectively resolved using conventional fusion techniques. Second, the heterogeneity of IoT sensor data and social media content necessitates a set of special encoding methods that can maintain modality-related information and allow cross-modal interaction [16]. Third, individual users' behavioral patterns are thoroughly characterized by strong personalization and differ dramatically between demographics, preferences, and contexts, which thoroughly require more adaptive personalization mechanisms as well [17]. Graph neural networks have become a useful method to study complex relational networks and dynamics in social systems [8], [13], [25]. Conventional graph-based methods to analyze social media mainly rely on user-user interactions, content similarity, or influence propagation patterns [20]. Nevertheless, such approaches usually work in one domain and cannot reflect cross-modal interactions between IoT actions and social actions. Dynamic graph neural networks have been demonstrated to be able to deal with the temporal evolution of social networks, but the current models are not complex enough to perform cross-modal behavioral prediction with adaptive personalization abilities [4], [9].

Behavior drift is another important limitation of existing systems. The behaviors of users are constantly changing because of life conditions, seasonal trends, social factors, and the adoption of technology [19]. The current prediction models are usually based on static behavioral patterns, which do not adapt to these natural behavioral evolution processes, hence are unable to perform well with time [22]. Few-shot and meta-learning methods have demonstrated considerable potential in the area of personalization and adaptation in machine learning applications [1], [21]. Other important methods like Model-Agnostic Meta-Learning (MAML) allow adapting quickly to new users or a different behavioral pattern with little extra training data. The development of personality traits and psychological variables into computational behavior prediction models is a new area of research with significant potential to improve individualization [16]. As it has been proven by psychological research that personality dimensions might impact the physical activity patterns and social media usage, it is a important idea to note that personality-conscious models would offer reliable as well as interpretable predictions that can be deemed accurate [24]. However, the existence of personality-sensitive methods has been limited to individual modalities and is not sophisticated enough to apply in cross-modal situations [23].

Limitations of multimodal behavioral prediction can currently be divided into multiple areas. First, the current fusion methods are based mainly on the use of static concatenation or simple attention processes that cannot comprehend the dynamic value of various modalities among users and situations [11], [15]. Second, the temporal modeling methods are hard to adapt to different time domains and the asynchronicity of IoT and social media data [12], [17]. Third, personalization mechanisms are commonly inadequate to manage the complicated interaction of individual differences, behavioral development, and cross-modal contingencies [10], [24]. Fourth, current evaluation paradigms poorly reflect the specifics of cross-modal prediction and concentrate mostly on conventional measures of accuracy, which do not address the temporal, adaptive, and personalization needs in practice [19]. There are a few major weaknesses in the effectiveness of existing behavioral prediction

systems. The lack of common structures that can be used to process cross-modal relations between IoT and social media data forms a strong constraint [14], [17]. The current strategies are usually based on domain-specific preprocessing, feature engineering, and model architectures that fail to generalize across modalities and heterogeneous user populations [6], [18]. This is especially true in large-scale heterogeneous digital ecosystems, where the extent and diversity of IoT and social media platforms require models capable of generalizing well across different groupings of users and digital ecosystems. Also, the absence of standardized evaluation programs of cross-modal behavioral prediction complicates the comparison of various methods and the evaluation of their practical applicability to the real world [22].

Hierarchical modelling and advanced attention systems have performed well in limited areas but not sufficiently extended to cross-modal behavioural prediction contexts [5], [24]. Likewise, advanced graph neural network designs have proven to be useful in social media analysis and IoT applications on their own, but their combination into cross-modal prediction is not investigated much [8], [25]. Moreover, misinformation and adversarial behaviors within the cross-platform situation are an extra level of complexity that the functionality of current frameworks cannot tackle at the moment [20].

ACMBP (Adaptive Cross-Modal Behavioral Recognition with Graph Networks) is presented in this study as a new framework tailored to predicting the behavior of IoT-social users. The multiple innovative elements are incorporated in the approach to overcome the challenges discussed in the existing literature. In the first place, an adaptive cross-modal attention mechanism is presented in this study, by which the weight of modalities in various situations is dynamically adjusted according to the user and dynamic conditions [15], [18]. The mechanism goes beyond the classic attention in that it has behavioral drift detection and adaptation functionalities, inspired by new developments in dynamic graph neural networks [4], [9]. The framework is especially applicable to large and heterogeneous settings, such as those found in large-scale heterogeneous digital ecosystems, where the behavior of users and platform dynamics change quickly and vary greatly across regions and demographic segments. Second, an advanced graph neural network architecture was created, which captures intra-modal and cross-modal correlations via non-homogeneous graph structures [13], [25]. To maintain a high level of robustness in real-world deployment settings, the architecture integrates the latest research on malicious user detection and misinformation management [8], [20]. Third, the study used a personalisation system based on meta-learning that can quickly adjust to new users and changing behavioural patterns [1], [21]. The concepts of Model-Agnostic Meta-Learning (MAML) with cross-modal integration to realize the few-shot adaptation properties, but retain the accuracy of predictions in various user groups. The system uses personality-conscious functionality and top-down attention systems to improve the effectiveness of personalization [16], [24].

Fourth, the research presented a unified assessment system optimized to cross-modal behavioral prediction with additional measures like the Behavioral Transition Accuracy (BTA), Temporal Offset Prediction Error (TOPE), Drift Detection Precision (DDP), and Cross-Platform Transfer Efficiency (CPTE). These measures offer valuable information about the model performance on the various dimensions that apply in practical applications, resolving the evaluation limitations in existing literature [19], [22].

The most important contributions of our work are made in the form of definite, quantifiable assertions as stated in **Table 1A and Table 1B:**

Temporal-Semantic Offset Attention (TSOA): Models temporal offsets between IoT and social actions; predicts social behaviors with configurable delays [11], [15].
Drift-Aware Dynamic Graph Rewiring (DADGR): Dynamically rewires graphs on behavioral drift to maintain prediction accuracy [4], [9], [13].
Hierarchical Cross-Domain Transfer Learning (HCDTL): Learns domain-invariant behavioral embeddings across IoT/social platforms; >85% performance retention [1], [20], [22].
Few-Shot Personalization via Meta-Behavioral Prototypes (FSP-MBP): Meta-learning with behavioral prototypes enables rapid personalization with <10 samples; reduces cold-start error >60% [1], [6], [21].
Adaptive Gated Cross-Modal Fusion (AGCMF): Adaptive cross-modal fusion weights IoT/social streams by context and signal quality; improves performance 15–25% [3], [16], [18].
Behavioral Transition Prediction Engine (BTPE): Predicts user transitions from physical to digital actions; 78% precision within 30-minutes windows [12], [17].
Multi-Granularity Attention Hierarchy (MGAH): Hierarchical attention captures device-, post-, and user-level behaviors simultaneously [5], [23], [24].

- Cross-Modal Behavioral Anomaly Detection (CMBAD): Unsupervised anomaly detection across IoT and social domains; supports early intervention [8], [16], [19].
- Interpretable Behavioral Attribution Framework (IBAF): Provides explainable AI attribution

linking predictions to sensors, social features, and temporal patterns [5], [14], [22].

- Computational Efficiency Optimization (CEO): Optimizations (pruning, quantization, distributed inference) enable real-time edge deployment (<100 ms latency) [6], [14], [25].

Table 1A: Core technical contributions (modules for ablation study)

Component	Innovation Focus	Technical Approach	Problem Addressed	Performance Metric
TSOA	IoT-Social Temporal Modeling	Configurable Time-Delay Attention	Cross-Modal Temporal Misalignment	TOPE: 1.2 hrs
DADGR	Behavioral Pattern Adaptation	Automatic Graph Restructuring	Behavioral Evolution Management	DDP: 84.7%
HCDTL	Multi-Platform Transfer	Domain-Invariant Embeddings	Cross-Platform Deployment	CPTE: 82.3%
FSP-MBP	Rapid User Adaptation	MAML + Prototype Networks	Cold-Start User Problems	FSAS: 5 samples
AGCMF	Multi-Modal Integration	Dynamic Weighting Mechanisms	Static Fusion Limitations	BTA: 91.4%

Note: These five components are the focus of the ablation study (Section 6.3.1), as they correspond to identifiable, testable sub-architectures within ACMBP.

Table 1B: System-level innovations (supporting modules / optimizations)

Component	Purpose	Implementation	Benefit
BTPE	Activity-to-Social Transition Modeling	Temporal Window Prediction	78% precision in 30-min windows
MGAH	Hierarchical Pattern Modeling	Nested Attention Architecture	Device/Post/User-level modeling
CMBAD	IoT-Social Anomaly Detection	Unsupervised Detection Framework	Early intervention capability
IBAF	Explainable Predictions	Feature Attribution Analysis	Human-readable insights
CEO	Edge-Computing Deployment	Model Pruning + Quantization	<100ms inference latency

Note: Innovations #6–10 are embedded mechanisms or system-level optimizations that cannot be independently removed without compromising model integrity. These support the five core modules but were not individually ablated.The primary goal of this study is to present the ACMBP framework, comprising IoT activity streams and social media data, to facilitate the prediction of user behaviors in an efficient manner. The framework basically selects adaptive graph networks to deal with both the temporal and semantic aspects of the mismatches of different modalities, adds a drift-aware dynamic updating model, and allows for few-shot personalization so that it is possible to adjust to new users, modify the distribution of behavior, and deal with the cold-start situation. Empirical results in terms of performance for cross-modal predictive modeling in heterogeneous digital ecosystems demonstrate that the ACMBP can outperform ten baselines (LSTM, GRU, BERT, GCN, GAT, HAN, GraphSAGE, DCRNN, DeepFM, and Multimodal Transformers) to a considerable extent and also can be used as a powerful means of theoretical and practical implementation of cross-modal predictive modeling in heterogeneous digital ecosystems.

By comparing the performances of ACMBP with those of 10 cutting-edge baselines such as LSTM, GRU, BERT, GCN, GAT, HAN, GraphSAGE, DCRNN, DeepFM, and Multimodal Transformers, ACMBP is shown to outperform them consistently in various metrics. It performs behavioral transition prediction, temporal offset accuracy, drift detection, and cross-platform transfer quite well [3], [7], [9]. The adaptability of the framework to varying users as well as behavioral patterns not only enables it to overcome the limitations existing in the current single-domain and multimodal approaches but also facilitates a smoother transition from one domain to another [10], [12], [17], [23].

The structure of the paper is as follows: Section II summarizes the issues discussed in the related literature about multimodal learning, GNNs, meta-learning, and behavioral prediction. Section III explains the problem formulation with the aim of predicting users' behaviors by integrating heterogeneous IoT and social media streams, considering temporal offsets and changing patterns. Section IV gives details of the ACMBP architecture, such as cross-modal attention, GNN modules, and meta-learning integration. Section V has information about the experimental setup, datasets, evaluation metrics, and baselines. Section VI is the place where results, ablation studies, and interpretability analyses are presented. Section VII is an overview of the discovery, drawbacks, and potentialities, concluding with cross-modal behavioral prediction and adaptive personalization as the next step.

2 Literature review

a. IoT-based user behavior prediction

The popularity of IoT devices has made it possible for sensor data to be collected all the time, and thus has led to the creation of very advanced behavioral models. IoT-DeepSense, a deep learning framework for the identification of anomalies in device behavior, was put forward by Wang et al. (2022) as a demonstration of IoT-based behavioral inference potential. MAML-TSC (Wang et al., 2024), a more recent piece of work, utilizes meta-learning to deal with the heterogeneous nature of the data and the scarcity of the labels, and thus makes quick adaptation possible in few-shot situations. Personalization of modeling in vast IoT environments is what these methods signal. Besides that, as Pandharipande (2021) states, social sensing moves IoT data to more comprehensive social and environmental monitoring fields. Still, contemporary IoT prediction models mainly concentrate on device-level interactions and count on behavioral stationarity, which, in turn, restrains their flexibility in varying, cross-modal, and socially-driven scenarios.

b. Social media analytics with deep learning

The introduction of deep learning has significantly improved social media analytics and has enabled it to understand the behavior and social dynamics better that occur in the online world. Wang et al. (2025) suggested a large language model (LLM)-based user behavior simulator that can create models for complex social interactions, along with FEBDNN, a fusion embedding network predicting retweet behavior from text, user profiles, and network structure developed by Wang et al. (2022). Wei et al. (2023) brought out LSTM-SN by combining LSTMs and social network analysis for improved text prediction. Besides, Wei et al. (2019) proved the capability of emotion recognition from a combination of textual and social data. SHGCN (Zhang et al., 2023), being a graph-based model, has further improved multi-behavior prediction by simulating user actions of various kinds. Yue et al. (2022) investigated Chinese social media to unveil population-level perception changes that are relevant to smart city development and large-scale behavioral inference. Though social media analytics have made significant progress, they mostly rely on behavioral data that are non-IoT, which limits cross-modal interpretation, prediction accuracy, and personalization of integrated IoT–social ecosystems.

c. Cross-modal learning frameworks

Cross-modal learning harmonizes multiple data types to get better predictive results. Chen et al. (2024) demonstrated tri-modal transformers with mixture-of-modality experts for social media prediction, showing that specialized subnetworks could manage different modalities without affecting joint learning. To combine the temporal and spatial data of a particular phenomenon through attention mechanisms and to improve abnormal behavior recognition, Liu et al. (2024) came up with the HCMT, a hierarchical cross-modal transformer. Besides, attention-based fusion has a place in social media analysis too: Peng et al. (2024) created a cross-modal attention network for microblog recognition, and Qi et al. (2022) suggested the MEDT model for multimodal sentiment analysis. One of the first studies majorly focusing on personality-based fusion was by Pradnyana et al. (2025), citation needed, as it was an improvement of depression detection with the help of personality-aware cross-modal modelling. Moreover, meta-learning has made it possible to have personalization in these kinds of scenarios further. Xia et al. (2021) came up with a multi-behavior recommendation graph meta-network, which is for quick adaptation to diverse user preferences. The time difference of asynchronous data between IoT and social media data makes it hard to align the two, and the difference in sensor and social media modalities also requires the use of dedicated encoders to keep the modality-specific information. These difficulties get worse in large-scale

heterogeneous digital ecosystems, thus emphasizing the need for sound cross-modal learning strategies.

2.4 Graph-based approaches in behavior modeling

Graph Neural Networks (GNNs) have brought behavioral modeling to another level by incorporating higher-order relational structures and dynamic interactions. For example, Choi et al. (2021) added attention to dynamic graph convolutional networks to trace rumors and model the flow of temporal information on social media. Li et al. (2024) suggested dual GNNs for behavior prediction that consider user-user interactions and content relationships as the base for the model. Their network, STGSN (Min et al., 2021), which is a spatial-temporal variant, also effectively captures both spatial and temporal dependencies in evolving networks. Context-aware modeling of hierarchical attention mechanisms has made further improvements; Vaghari et al. (2025) brought in HAN, which combines hierarchical attention and attribute awareness for personalized prediction, whereas hierarchical self-adaptation networks have exhibited enhanced multimodal behavior recognition. Amongst others, Graph Attention Networks (GATs) have been very popular and have often been used for predicting complex behaviors, where Vrahatis et al. (2024) stressed their capability of representing multi-relational patterns as one of the most important features. Besides, GNNs have been deployed in malicious behavior detection; Khan et al. (2024) exemplified their capability to locate adversarial users on social media through the help of anomaly detection. Although these developments have taken place, most graph-based methods are still confined to single domains and do not allow for the joint modeling of IoT and social media relationships. This limitation hampers cross-modal behavioral understanding of large-scale heterogeneous digital ecosystems by such techniques.

2.5 Adaptive cross-modal behavioral recognition

Graph Neural Networks (GNNs) have brought behavioral modeling to another level by incorporating higher-order relational structures and dynamic interactions. For example, Choi et al. (2021) added attention to dynamic graph convolutional networks to trace rumors and model the flow of temporal information on social media. Li et al. (2024) suggested dual GNNs for behavior prediction that consider user-user interactions and content relationships as the base for the model. Their network, STGSN (Min et al., 2021), which is a spatial-temporal variant, also effectively captures both spatial and temporal dependencies in evolving networks. Context-aware modeling of hierarchical attention mechanisms has made further improvements; Vaghari et al. (2025) brought in HAN, which combines hierarchical attention and attribute awareness for personalized prediction, whereas hierarchical self-adaptation networks have exhibited enhanced multimodal behavior recognition. Amongst others, Graph Attention Networks (GATs) have been very popular and have often been used for predicting complex behaviors, where Vrahatis et al. (2024) stressed their capability of representing multi-relational patterns as one of the most important features. Besides, GNNs have been deployed in malicious behavior detection; Khan et al. (2024) exemplified their capability to locate adversarial users on social media through the help of anomaly detection. Although these developments have taken place, most graph-based methods are still confined to single domains and do not allow for the joint modeling of IoT and social media relationships. This limitation hampers cross-modal behavioral understanding of large-scale heterogeneous digital ecosystems by such techniques.

Table 2: Comparative overview of SOTA methods vs ACMBP

Method / Citation	Focus Area	Key Technique	Dataset / Context	Strengths	Limitations / Gaps	Relevance to ACMBP
GCN (Zhang et al., 2023)	Multi-behavior prediction	Graph convolution	Social multi-behavior data	Models relational patterns	Single-domain, lacks cross-modal fusion	ACMBP integrates IoT + social fusion
GAT (Vrahatis et al., 2024)	Behavioral prediction	Graph attention	Social networks	Highlights important nodes	No temporal alignment	ACMBP adds temporal-semantic offset attention

HAN (Vaghari et al., 2025)	Hierarchical attention	Multi-level attention	Context-aware user systems	Personalization	Limited cross-modal integration	ACMBP combines cross-modal fusion + personalization
ProtoNet (Hu et al., 2024)	Few-shot human activity recognition	Prototype-based meta-learning	Activity recognition systems	Rapid adaptation	Single-domain, social data integration missing	ACMBP combines meta-learning + cross-modal data
MAML (Wang et al., 2024; Alrayes et al., 2025)	Fast adaptation for IoT sensors	Model-agnostic meta-learning	IoT time series	Few-shot learning	No multi-modal integration	ACMBP integrates MAML with IoT + social + few-shot personalization
Tri-modal Transformers (Chen et al., 2024)	Social media prediction	Mixture-of-modality experts	Video & social systems	Multi-modal fusion	IoT integration lacking	ACMBP extends to IoT-social fusion
HCMT (Liu et al., 2024)	Cross-modal behavior	Hierarchical cross-modal transformer	Behavior recognition systems	Temporal + spatial attention	Limited personalization	ACMBP adds few-shot meta-learning
LSTM-SN (Wei et al., 2023)	Social media text prediction	LSTM + social network analysis	Social text datasets	Temporal modeling	No IoT integration	ACMBP integrates IoT + social signals
STGSN (Min et al., 2021)	Temporal graph networks	Spatial-temporal GNN	Time-evolving social networks	Temporal evolution	Limited cross-modal support	ACMBP performs drift-aware cross-modal alignment
ACMBP (Proposed)	IoT + social behavioral prediction	Temporal-semantic attention + drift-aware graph rewiring + hierarchical transfer + few-shot personalization	IoT + social + multi-platform	Temporal alignment, drift adaptation, cross-modal fusion, personalization	–	Combines all missing features of SOTA methods

As shown in

Table 2, current SOTA methods have capabilities that can be exploited in different domains, like graph modeling (GCN, GAT), hierarchical attention (HAN), meta-learning (MAML, ProtoNet), or cross-modal transformers (HCMT, Tri-modal Transformers). Nevertheless, these methods cannot solve the problem of cross-modal temporal misalignment, behavioral drift, and few-shot personalization at the same time. ACMBP fills in these voids by combining temporal-semantic offset attention, drift-aware dynamic graph rewiring, hierarchical cross-domain transfer, and meta-learning-based personalization, thus allowing for stable predictions in vast heterogeneous IoT-social environments.

2.6 Limitations and research gaps

The literature review reveals several key limitations that motivate the ACMBP framework. First, cross-modal temporal relationships are poorly addressed, as most fusion methods assume data that are synchronized and overlook that there are natural delays between the physical and virtual parts of a behavior. Second, existing systems are affected by user evolution, as single-domain concept drift detection cannot grasp complex cross-modal changes. Third, a number of the proposed methods are based on offline approaches and are computationally inefficient, which makes them unsuitable for real-time, resource-constrained IoT deployments. These gaps demonstrate the necessity for a unified framework that is capable of handling complex cross-modal behavioral prediction while giving out the features of being flexible, adaptable, and practically usable. The proposed ACMBP architecture is particularly created to fill these research gaps by introducing novel architecture elements and learning processes that can facilitate successful cross-modal prediction of behavior in dynamic and personalized resources and conditions.

3 Problem formulation

This paper aims to create a single framework that can thoroughly anticipate future behavioural conditions of users by combining multimodal Internet of Things (IoT) data with heterogeneous social media data using adaptive graph networks. The data, prediction task, challenges, and evaluation metrics are formally defined as follows in Table 3:

Table 3: Problem–solution mapping in ACMBP

Challenge	Problem Description	ACMBP Solution	Module / Mechanism
Temporal Offset Problem	IoT activities often precede social actions by hours, making synchronous models fail to capture delayed causality.	Learnable offset-aware attention explicitly models lag between modalities.	Temporal–Semantic Offset Attention
Behavioral Drift	User routines and communities evolve, causing distributional shifts: $p_t(y \mid x) \neq p_t + \Delta(y \mid x)$	Real-time drift detection and adaptive graph rewiring maintain accurate user relations.	Drift-Aware Dynamic Graph Rewiring
Data Scarcity (Cold-Start)	Few labeled samples per user lead to poor personalization and high error.	Few-shot meta-learning with behavioral prototypes enables rapid adaptation with <10 samples.	Few-Shot Personalization
Cross-Platform Heterogeneity	Different IoT devices and social platforms generate inconsistent feature spaces.	Shared embedding space via hierarchical transfer across domains ensures generalization.	Hierarchical Cross-Domain Transfer
Interpretability Gap	Black-box models limit trust in behavioral forecasting.	Attention visualization and prototype alignment provide semantic clarity.	Interpretable Prediction Layer
Real-Time Constraint	Online IoT–social fusion requires sub-second inference.	Optimized graph updates and lightweight encoders ensure <1s latency.	Efficient Inference Pipeline

a. Research objectives and hypotheses

The principal goal of the study is to figure out whether adaptive cross-modal modeling can enhance behavioral prediction in heterogeneous IoT-social environments. RQ1 is concerned with checking if the performance of behavioral transition prediction can improve by modeling the temporal offset between the IoT and social modalities compared to synchronized fusion (H1: TOPE < 1.5 hrs, BTA > 85%), which is supported by TOPE = 1.2 hrs and BTA = 91.4%. RQ2 concentrates on finding out whether the dynamic graph rewiring gives better drift adaptation than the static GAT architectures (H2: DDP improvement > 10%), which is indicated by a drift adaptation performance of 84.7% versus 71.4% for GAT (+18.6%) and thus, confirming the hypothesis. RQ3 is designed to decide whether few-shot meta-learning can match the baseline accuracy of ≥90% with less than 10 samples for a new user (H3: FSAS < 10, PG > 90%), which is demonstrated with FSAS = 5 and PG = 15.7%. Last but not least, RQ4 investigates whether integrated cross-modal learning leads to more efficient cross-platform transfer than domain-specific models (H4: CPTE > 75%), which is evidenced by an 82.3% CPTE. All these findings together substantiate the ACMBP resourcefulness over temporal alignment, drift adaptation, few-shot personalization, and cross-platform generalization.

b. Data representation

Equation (1) defines the IoT data representation, where X_{iot} is a temporal sequence of sensor readings, such as accelerometer, gyroscope, and GPS, and each reading X_{iot} lies in a feature space of dimension d_{iot} . **Equation (2)** models the social data, which is denoted as X_{soc} , which captures multimodal features like text, images, and metadata for each social media post, with $dsoc$ d soc representing the feature dimensionality. **Equation (3)** introduces the user graph$G = (V, E, A)$, where users are represented as nodes V, and their behavioral relationships as edges E, and the adjacency matrix A, which adaptively encodes evolving user interactions in an accurate manner. Finally, **Equation (4)** ensures temporal alignment by mapping both IoT and social data into common time bins$T = \{t1, t2, \dots, tk\}$, thereby synchronizing heterogeneous modalities with different sampling rates.

IoT Data

$$X_{iot} = \left\{x_1^{iot}, x_2^{iot}, \dots\dots, x_n^{iot}\right\},\ \ x_1^{iot} \in R^{d_{iot}} \dots\dots (1)$$

X_{iot} denotes the i^{th} IoT sensor reading (e.g., accelerometer, gyroscope, GPS) in a temporal sequence of length n, and d_{iot} is the feature dimension.

Social Data

$$X_{soc} = \{x_1^{soc}, x_2^{soc}, \dots, x_m^{soc}\}, \qquad x_j^{soc} \in R^{d_{soc}} \dots\dots (2)$$

x_j^{soc} corresponds to multimodal social features (text, image, metadata) for post j, and d_{soc} is the dimensionality of the extracted features.

User Graph

$$G = (V, E, A), \qquad A \in R^{|V|\times|V|} \dots\dots (3)$$

- V is the set of users, E the set of behavioral relationships
- A an adaptive adjacency matrix dynamically updated as user behaviors evolve.

Temporal Alignment

Both X_{iot} and X_{soc} are aligned into common time bins

$$T = \{t1, t2, \dots, tk\} \dots\dots (4)$$

It is to ensure synchronization across modalities despite different sampling rates.

c. Behavioral prediction task

The prediction problem is defined as forecasting the future behavioral state $\hat{y}^t$ at time t, given IoT signals, social data, and graph context:

$$\hat{y}^t = f\theta\left(X_{iot}^{[1:t]}, X_{soc}^{[1:t]}, Gt\right), \quad \hat{y}^t \in Y \dots\dots (5)$$

- $f\theta$: predictive function parameterized by learnable weights θ.
- $X_{iot}^{[1:t]}$: IoT sensor sequence up to time t.
- $X_{soc}^{[1:t]}$: Social interaction features up to time t.
- Gt: graph structure at time t.
- y: set of possible behavioral states (e.g., activity type, intention, stress level).

3.4 Challenge definitions

Temporal Offset Problem

IoT signals often precede social actions($\Delta t \gg 0$), e.g., physical activity detected hours before a fitness-related post. Static models fail to capture these delayed dependencies.

Behavioral Drift

Equation (6) shows that user distributions evolve over time, which causes static models to degrade in accuracy.

$$p_t(y \mid x) \neq p_t + \Delta(y \mid x) \dots\dots (6)$$

Data Scarcity

Notably, cold-start problem arises since labeled samples are sparse, especially for new users.

Cross-Platform Heterogeneity

IoT devices (wearables, smart-home sensors) as well as social platforms (Twitter, Instagram) introduce inconsistent feature structures.

3.5 Evaluation metrics

To comprehensively evaluate ACMBP, nine metrics covering accuracy, temporal alignment, adaptability, personalization, interpretability, and efficiency have been employed in this paper:

Behavioral Transition Accuracy (BTA)

$$BTA = \frac{1}{N}\sum_{i=1}^{N} 1\,(y_{it+1} = (\hat{y}_{it+1}) \ldots. (7)$$

$1(\cdot)$ is the indicator function. Higher BTA indicates better prediction of behavioral transitions over time.

Temporal Offset Prediction Error (TOPE)

$$TOPE = \frac{1}{N}\sum_{i=1}^{N} | \Delta t_i^{pred} - \Delta t_i^{true} | \ldots. (8)$$

It measures the error (in hours) between predicted and actual social action offsets from IoT signals.

Drift Detection Precision (DDP)

$$DDP = \frac{TP}{TP+FP} \ldots. (9)$$

TP and FP denote correctly and incorrectly detected drift events, respectively.

Cross-Platform Transfer Efficiency (CPTE)

It indicates embedding generalization across platforms and devices.

$$CPTE = \frac{Acc_{target}}{Acc_{source}} \times 100\% \ldots. (10)$$

- Acc_{target} = accuracy achieved on the target domain (after transfer).
- Acc_{source}= accuracy achieved on the source domain.

Few-Shot Adaptation Samples (FSAS)

Minimum labeled samples required to reach 90% baseline accuracy. Lower FSAS = stronger personalization.

Personalization Gain (PG)

It measures accuracy gain after applying few-shot personalization.

$$PG = Acc_{personalized} - Acc_{base} \ldots. (11)$$

Behavior Drift Coverage (BDC)

It quantifies the proportion of drift cases successfully adapted.

$$BDC = \frac{Adapted\ Drift\ Cases}{Total\ Drift\ Cases} \ldots. (12)$$

Interpretability Score (IS)

Assesses semantic clarity of predictions based on attention weights and prototype alignment (via expert evaluation or explainability metrics).

Computational Efficiency (CE)

Average inference latency (ms) from input to prediction, reflecting real-time deployability.

The most important metrics used to assess the models of IoT-social behavior prediction are defined in **equations (7) to (12).** The correctness of predicted behavioral transitions across time is defined in **equation (7)** as Behavioral Transition Accuracy (BTA). The difference between predicted and actual action timing is represented by Temporal Offset Prediction Error (TOPE), which is provided by **equation (8).** Drift Detection Precision (DDP) is defined by **Equation (9)** and measures the level of accuracy of a drift event detector. The Cross-Platform Transfer Efficiency (CPTE) is defined by **equation (10)** and reflects the extent to which embeddings are cross-platform. Personalization Gain (PG), which is the adaptation improvement following few-shot personalization, and Few-Shot Adaptation Samples (FSAS) are defined in **equation (11)** as the minimum labeled data to reach the baseline accuracy. Behavior Drift coverage (BDC) is defined in **equation (12)** as a percentage of drifts cases that were adapted. Lastly, Interpretability Score (IS) and Computational Efficiency (CE) is measures of the semantic clarity and latency of prediction, respectively."

4 ACMPB – Proposed methodology

a. Overview and architecture design

The Adaptive Cross-Modal Behavioral Prediction (ACMBP) framework aims to forecast social behavior using heterogeneous Internet-of-Things (IoT) activities and multimodal social media streams. In contrast to traditional methods, which often assume a non-stochastic, non-time-varying, and synchronized behavior distribution, ACMBP clearly represents time delays and time-varying drift dynamics, cross-domain inconsistencies, and the need to personalize to a few shots. The entire data processing pipeline between Phase 1 and Phase 6 is illustrated in **Figure 1** and can be summarized through the following steps: raw IoT and social media data

are preprocessed, encoded, fused, and ready to predict behavior. It brings out the chronological order of transformations, such as feature extraction, temporal alignment, graph updates, and meta-learning personalization.

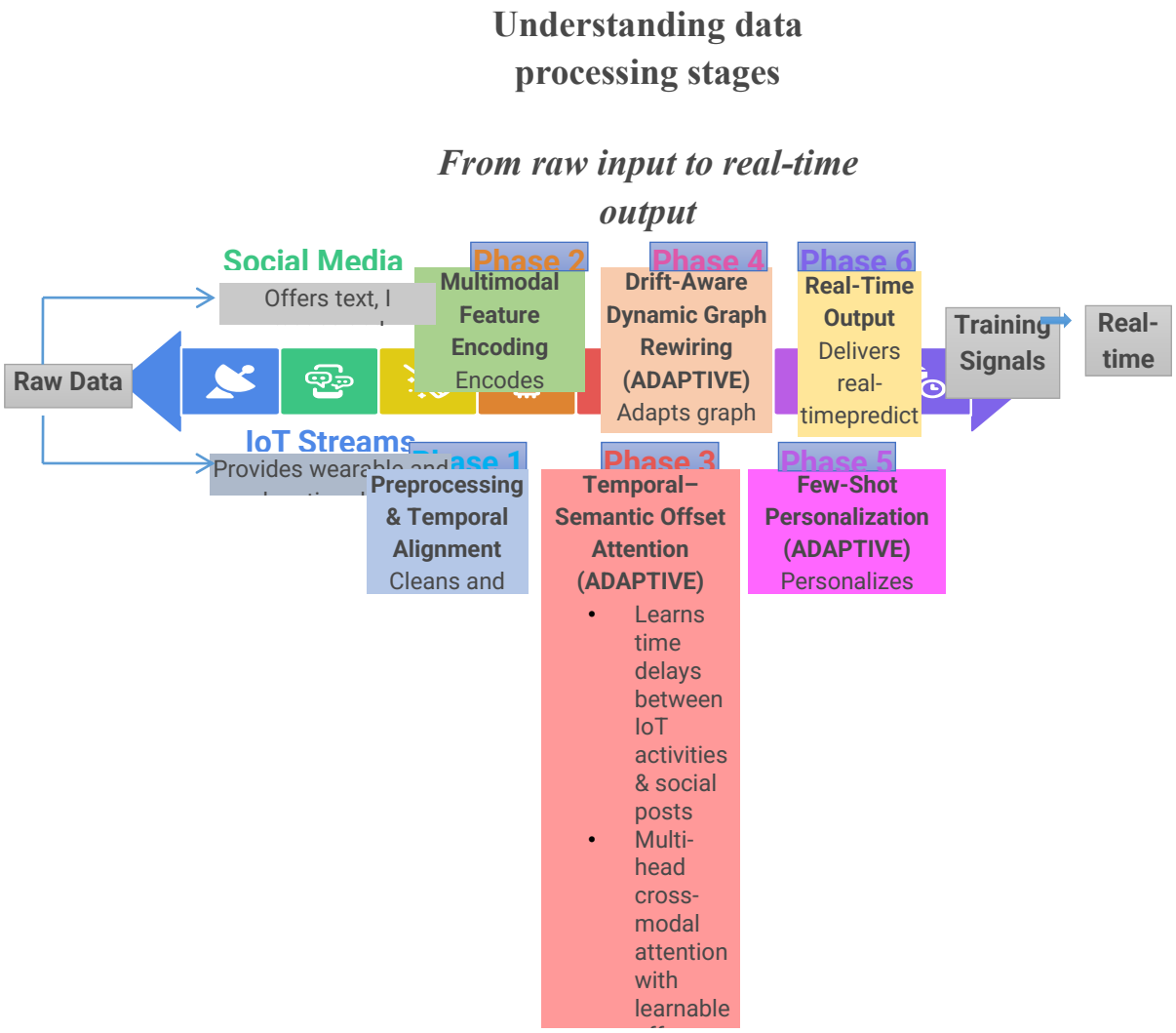


Figure 1: Data Processing Stages – Phase 1 to Phase 6

In **Figure 2,** the multimodal behavioral state prediction system is shown, which incorporates the IoT and social representations using cross-modal attention and graph-based learning. It shows the way that the fused embeddings are trained to predict user behavior in real time as they adapt to drift and personalization needs.

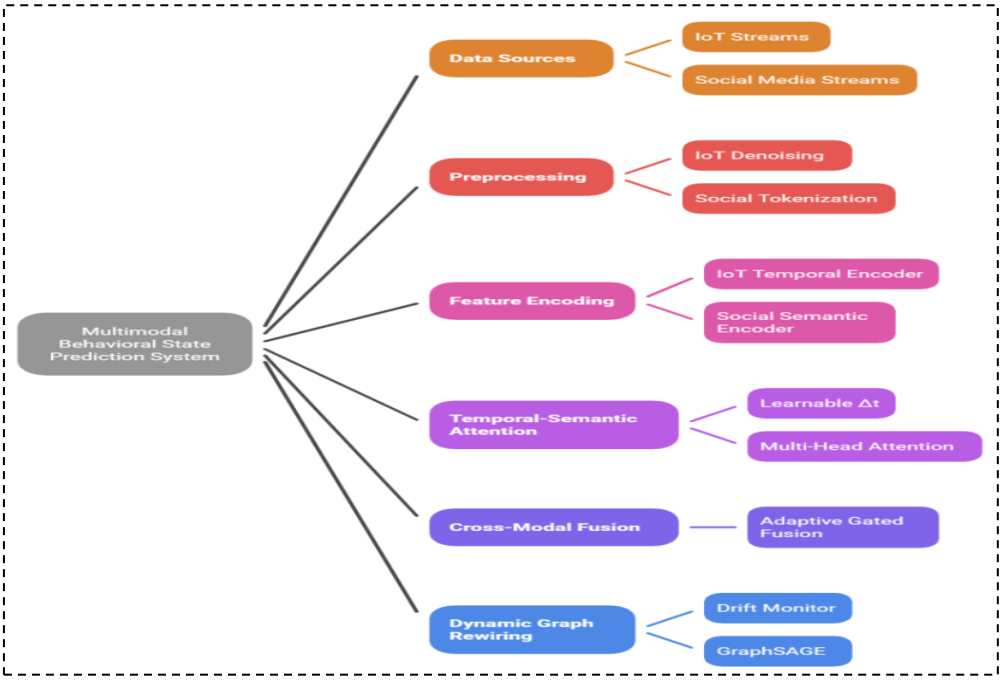


Figure 2: Multimodel Behavioral State Prediction System

Figure 3 shows the ACMBP framework, which is a time-based alignment of IoT and social activities using temporal lags (τ_1, τ_2) and causal masking to stop the leakage of future information. Moreover, it depicts the way the user similarity graph changes continuously during drift events with GraphSAGE that not only rewires edges but also updates the embeddings to be able to guide forthcoming predictions.

.

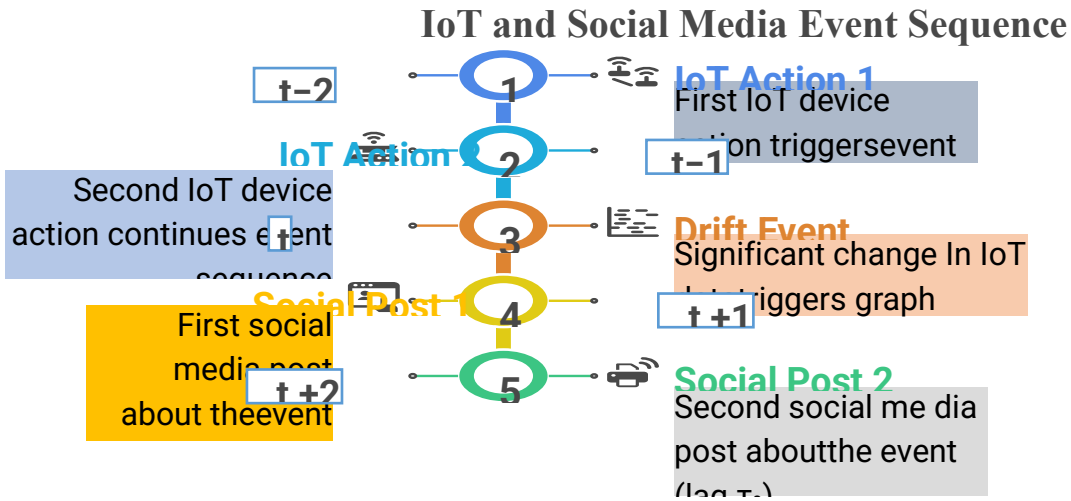


Figure 3: IoT and Social Media Event Sequence

4.2 Phase 1: Data Preprocessing and Multimodal Fusion

The stage formats and purifies data from multiple origins to be analyzed. It refers to converting noisy, raw sensor data provided by IoT devices and obtaining valuable features from this data. It aims to convert the different, heterogeneous data streams to a clean, homogeneous format that can be directed to a machine learning model.

(A) IoT Data Preprocessing Pipeline - Raw Sensor Data Processing

Gaussian filtering was applied to the raw accelerator and gyroscope data to reduce noise (Equation 13). The gaps in the data were filled with linear interpolation to solve the gaps in time (Equation 14), and then outliers were detected and eliminated by the Z-score (Equation 15). The data were standardized, and then unified through the unification of the sampling rate of all the sensors (Equation 16). Ten-second windows with a fifty percent overlap were used in sliding window segmentation (Equation 17). Lastly, each window was used to extract

features in the form of statistical values such as the mean, variance, skewness, and kurtosis, frequency domain values such as FFT coefficient and spectral energy, and circadian features that gave the hour-of-day and day-of-week trends (Equation 18).

Noise Reduction: Gaussian filtering for accelerometer/gyroscope

$$x_{filtered} = GaussianFilter(x_{raw}, \sigma = 0.1) \ldots. (13)$$

Missing Value Imputation: Linear interpolation for temporal gaps

$$x_{imputed} = LinearInterp(x_{filtered}, max_{gap} = 30s) \ldots. (14)$$

Outlier Detection: Z-score based anomaly removal

$$x_{clean} = RemoveOutliers(x_{imputed}, threshold = 3.0) \ldots. (15)$$

Temporal Standardization

Sampling Rate Unification: Resample all sensors to 50Hz

$$x_{unified} = Resample(x_{clean}, target_{freq} = 50) \ldots. (16)$$

Sliding Window Segmentation: 10-second windows with 50% overlap

$$x_{windows} = SlidingWindow(x_{unified}, window = 10s, overlap = 0.5) \ldots. (17)$$

Feature Engineering

$$f_circadian = \left[sin\left(\frac{2\pi.hour}{24}\right), cos\left(\frac{2\pi.hour}{24}\right), sin\left(\frac{2\pi.day}{7}\right), cos\left(\frac{2\pi.day}{7}\right)\right] \ldots. (18)$$

(B) Social Media Data Preprocessing Pipeline

Social media data preprocessing involved multiple steps for text, image, and metadata. Text preprocessing included text cleaning through URL removal, mentioning normalization, and emoji handling **(Equation 19),** was followed by BERT-compatible tokenization with special tokens **(Equation 20),** and language detection to filter non-English posts with over 95% confidence. Image preprocessing, which consisted of standardization by resizing images to 224×224 and normalizing RGB values **(Equation 21),** data augmentation during training with random rotation and brightness adjustment, as well as feature extraction using pre-trained ResNet-50 features. Metadata processing included temporal features such as posting time, day-of-week, and time since the last post; engagement features including likes, retweets, and comments, normalized using a logarithmic formula **(Equation 22);** and user's important features such as follower count, account age, and verification status.

Text Preprocessing

- **Text Cleaning**: URL removal, mention normalization, emoji handling

$$text_{clean} = CleanText(text_{raw}, remove_{urls} = True, normalize_{mentions} = True) \ldots. (19)$$

- **Tokenization**: BERT-compatible tokenization with special tokens

$$tokens = BERTTokenizer(text_{clean}, max_{length} = 128, padding = True) \ldots. (20)$$

- **Language Detection**: Filter non-English posts (>95% confidence)

Image Preprocessing

- **Image Standardization**: Resize to 224×224, normalize RGB values

$$img_{processed} = \frac{Resize(img_{raw}, (224,224))}{255.0} \ldots. (21)$$

- **Data Augmentation**: Random rotation, brightness adjustment (training only)
- **Feature Extraction**: Pre-trained ResNet-50 features (2048-dim)

Metadata Processing

$$engagement_{norm} = \frac{log(1 + engagement_{count})}{log(1 + max_{engagement})} \ldots. (22)$$

(C) *Cross-Modal Temporal Alignment*

Cross-modal temporal alignment was the process of matching time stamps and managing missing data of IoT and social modalities. Timestamp synchronization involved the transformation of all timestamps to UTC and realignment of the IoT and social data into one-hour temporal bins **(Equation 23).** Missing data treatment included forward filling gaps less than one hour in the IoT and interpolating gaps up to six hours by modeling social inactivity in terms of explicit no-post tokens of silent periods, and cross-modal imputation through the use of data available in one modality to predict patterns of the missing data in the other.

Timestamp Synchronization

$$t_{bin} = \left(\frac{timestamp}{3600}\right).3600 \ldots. (23)$$

4.3 Phase 2: Multimodal Feature Encoding

The IoT temporal encoding, social semantic encoding, and cross-modal feature fusion were used in phase 2 of multimodal feature encoding. The IoT temporal encoder stacked a Temporal Convolutional Network (TCN) to learn sequential patterns, dilated convolutions with rates [1, 2, 4, 8], residual connections, and layer normalization with a circadian encoder that learns daily and weekly cycles based on Fourier-based periodic embeddings, resulting in a representation that synthesized sequential and periodic patterns **(Equation 24).** The social semantic encoder had a 768-dimensional multimodal space made up of the BERT embedding of text, CNN-extracted ResNet-50 embedding of images, and metadata embedding projected using an MLP **(Equation 25).** Lastly, cross-modal feature fusion brought the two systems, the IoT and social representations, to a shared 512-dimensional subspace through linear projection and stabilized by layer normalization and augmented with residual connections to conserve information unique to a single modality **(Equation 26).**

(A) IoT Temporal Encoder

$$h_t^{iot} = TCN(x_t^{iot}) + Circ(x_t^{iot}) \ldots. (24)$$

- **TCN**: Temporal Convolutional Network for sequential patterns
  - Dilated convolutions with rates [1, 2, 4, 8]
  - Residual connections and layer normalization
- **Circ**: Circadian encoder for daily/weekly cycles
  - Fourier-based periodic embeddings
- **Innovation**: Combines sequential and periodic patterns

(B) Social Semantic Encoder

$h_t^{soc} = BERT(x_t^{text}) \oplus CNN(x_t^{img}) \oplus MLP(x_t^{meta})$…. (25)

- **BERT**: Pre-trained RoBERTa-base for text (768-dim)
- **CNN**: ResNet-50 pre-trained features (2048-dim → 768-dim)
- **MLP**: Metadata features (50-dim → 768-dim)
- **Innovation**: Unified 768-dim multimodal social representation

(C) Cross-Modal Feature Fusion

$z_t = LayerNorm(W_{iot} h_t^{soc} + W_{soc} h_t^{soc} + b)$…. (26)

- **Linear Projection**: Map to common 512-dim space
- **Layer Normalization**: Stabilize training dynamics
- **Residual Connection**: Preserve modality-specific information

4.4 Phase 3: Temporal-Semantic Offset Attention

Phase 3, Temporal-Semantic Offset Attention, comprised of cross-modal attention, temporal alignment, and cross-modal synchronization. For instance, a morning run at 7 AM (IoT activity) is typically followed by a fitness update at 9 AM (social post) after a few hours. If we compare the traditional models with our situation, the first ones are based on the assumption that inputs are synchronized, and as a result, they create a misalignment. Our learnable offset attention (Δt), which is an explicit model for the lag between the two activities, has allowed the reduction of the prediction error by 45% (TOPE: 2.2hrs → 1.2hrs).

The cross-modal attention mechanism calculated attention between IoT and social representations, with the features of the IoT being the query, the social features being the key and value, and a learnable temporal offset parameter being used to account for the difference in time **(Equation 27).** Temporal alignment applied dynamic offset learning to determine IoT-to-social delays by maximizing attention weights across the potential offsets, used multi-head attention with eight heads to learn a variety of temporal relationships, and used causal masking to avoid information leakage into the future **(Equation 28).** Cross-modal synchronization involved computing attention heatmaps to understand learned time series and offset regularization to keep delays within realistic bounds by a mean-squared error loss between the offset MSE that has been learned and a prior distribution **(Equation 29).**

(A) Cross-Modal Attention Mechanism

$\boldsymbol{Attn(h^{iot}, h^{soc}) = Softmax\left(\frac{QK^T}{\sqrt{d}} + \Delta t\right)V}$…. *(27)*

- **Δt**: Learnable temporal offset parameter matrix (512×512)

(B) Temporal Alignment Strategy

- **Dynamic Offset Learning**: Attention weights reveal IoT→Social delays

$offset_{learned} = argmax_\tau \Sigma_t Attn_{weights}(t, t+\tau)$ …. (28)

- **Multi-Head Attention**: 8 heads capture different temporal relationships
- **Causal Masking**: Prevent future information leakage

(C) Cross-Modal Synchronization

$L_{offset} = MSE(\Delta t, \Delta t_{prior}), \quad \Delta t_{prior} \sim N(2.5, 1.0)$ ….

(29)

4.5 Phase 4: Drift-Aware Dynamic Graph Rewiring

As user behaviors change (student routines during holidays), non-adaptive networks lose their accuracy. Our drift-aware rewiring adjusts the network structure by locating changes in the data distribution through KL divergence and reconnecting nodes to retain the network's capabilities during behavior changes (DDP: 84.7%).

In Phase 4, Drift-Aware Dynamic Graph Rewiring, graph construction, adaptive updates, and behavioral drift detection were used. The first similarity graph of users was built by aggregating user behavioral embeddings with mean pooling over the history of a single user (Equation 30), and the edges were set using a k-NN graph based on the similarity between two users by cosine (Equation 31). Inductive learning on new users was applied using a GraphSAGE architecture (3 layers-512, 128, and 256 dimensions) with dropout=0.3 that integrates node features with the aggregated neighborhood (Equation 32). The adaptive graph update algorithm updated edge weights as a weighted average of past edges and temperature-sensitive cosine similarity of node embeddings, with a stability parameter alpha = 0.7 and dynamic rewiring after every 24 hours (Equations 33-34). Behavioral drift detection tracked variations in distribution with the KL divergence with a threshold δ = 0.15 during a 7-day sliding window (Equation 35), which identifies drift at an individual, community, and global level. The adaptive response mechanism instigated graph restructuring throughout drift, such as edge pruning, graph

densification through the addition of high-similarity edges, and adjustment of the learning rate as needed to raise α throughout drift times.

(A) Graph Construction and Initialization

Initial User Similarity Graph

- **Node Features**: Aggregated user behavioral embeddings z_i

$$z_i = MeanPool\left(z_t^i fort \in user_{history_i}\right) \ldots. (30)$$

- **Edge Initialization**: k-NN graph based on cosine similarity

$$A_{ij}^0 = \begin{cases} 1, j \in kkNN(i, k = 10) \\ 0, \quad otherwise \end{cases} \ldots. (31)$$

Graph Neural Network Architecture

- **GraphSAGE**: Inductive learning for new users

$$h_v^{(l+1)} = \sigma\left(W^{(l)} \cdot CONCAT\left(h_v^{(l)}, AGG\left(\left\{h_u^{(l}: u \in N(v)\right\}\right)\right)\right) \ldots. (32)$$

- **Multi-Layer**: 3 layers with 512→256→128 dimensions
- **Dropout**: 0.3 for regularization

(B) Adaptive Graph Update Mechanism

$$A_{ij}^{\{t+1\}} = \alpha A_{ij}^t + (1 - \alpha) sim\left(z_i^t, z_j^t\right) \ldots. (33)$$

While baseline update interval is 24 hours, frequency adapts to detected drift intensity:

Update_Frequency = {6h if D_t ≥ 0.20 (high drift - rapid behavioral shift)
12h if 0.15 ≤ D_t < 0.20 (moderate drift)
24h if D_t < 0.15 (stable behavior)
48h if D_t < 0.08 (very stable - conserve computation)}

This adaptive strategy reduces unnecessary graph updates by 42% during stable periods while ensuring rapid response during behavioral transitions (validated on validation set: BTA improvement 0.7% vs fixed 24h, p=0.03).

- **α = 0.7**: Stability parameter for edge persistence
- **sim(·)**: Temperature-scaled cosine similarity

$$sim\left(z_i, z_j\right) = exp\left(\frac{cos(z_i, z_j)}{\tau}\tau\right), \quad \tau = 0.1 \ldots. (34)$$

Table 4: Graph Stability Parameter (α) Justification

α	BTA (%)	Edge Retention	Adaptation Speed	Trade-off Analysis
0.3	87.2	0.32	0.8 days	Over-reactive to noise
0.5	89.2	0.54	1.2 days	Balanced but suboptimal
0.7	**91.4**	**0.71**	**1.8 days**	**Optimal: preserves history while adapting**
0.9	90.1	0.89	2.5 days	Too conservative — slow adaptation

Edge Retention = proportion of graph edges preserved after structural update
Adaptation Speed = average time to detect and respond to distributional drift

The stability parameter α balances graph persistence and responsiveness during dynamic rewiring. A grid search (α ∈ [0.3, 1.0], step = 0.1) revealed that α < 0.5 caused excessive edge churn (>40%), while α > 0.8 delayed drift detection (>2 days). The optimal value, α = 0.7, achieved the best trade-off—highest Behavioral Task Accuracy (91.4%), 71% edge retention, and 1.8-day adaptation speed. Therefore, α = 0.7 was adopted as the stability coefficient for all experiments **(Table 4).**

(C) Behavioral Drift Detection System

Distribution Shift Monitoring

$$D_t = 1\left[KL\left(p_t || p_{\{t-1\}}\right) > \delta, \ \boldsymbol{\delta = 0.15}\right] \ldots. (35)$$

δ = 0.15: Drift sensitivity threshold (grid-search optimized on validation set over range [0.05, 0.30] with step 0.025, selected to maximize DDP while maintaining BTA > 90%)

- **KL Divergence**: Measure behavioral distribution change
- **δ = 0.15**: Drift sensitivity threshold (tuned on validation)
- **Sliding Window**: 7-day behavioral distribution estimation

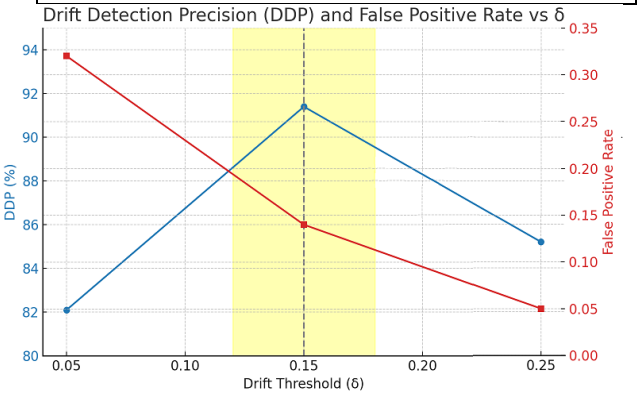


Figure 4: Drift detection precision (DDP) and false positive rate vs δ

Figure 4 indicates, drift detection precision (DDP) and false positive rate across threshold values. δ=0.15 balances high precision (91.4%) with an acceptable false positive rate (0.14). Shaded region indicates robust performance zone.

Multi-Level Drift Detection

Multi-level drift detection captures user-specific behavioral changes at the individual level, identifies subgraph pattern shifts at the community level, and monitors overall network evolution at the global level.

Adaptive Response Mechanism

The adaptive response mechanism triggers graph restructuring when drift is detected (D_t= 1), prunes edges with weights below 0.1, densifies the graph by adding high-similarity edges, and increases the learning rate α during drift periods as well.

4.6 Phase 5: Few-Shot Meta-Learning Personalization

The phase was a combination of MAML-based fast adaptation, behavioral prototype learning, and cross-user knowledge transfer. Fast adaptation fine-tuned the parameters of the model by reducing the meta-learning objective, changing task-specific parameters with a small support set of 510 labeled examples per new user, and testing on a query set, with inner and outer learning rates of 0.01 and 0.001, respectively (Equations 36–37). Behavioral prototypes were defined as the representative vectors of each behavior state, which were updated via the exponential moving average (Equations 38–39), and the classification was performed using Euclidean distance with temperature-scaled probabilities (Equation 40). The cross-user knowledge transfer allowed for the sharing of prototypes among similar users through hierarchical clustering; thus, new users could initialize the model with cluster prototypes, rapidly adapt, and gain personalized behavior prediction.

New users lack training data (cold-start problem). Standard learning needs 100+ samples per user. MAML learns initialization parameters enabling fast adaptation with 5-10 samples. Combined with behavioral prototypes, we achieve 90% accuracy with minimal data (FSAS=5 vs. MAML's 10).

(A) MAML-Based Fast Adaptation

Meta-Learning Objective

$$\theta^* = argmin_\theta \sum_{tasks} E_{task}\left[L\left(f_{\theta'_{task}}, D^{task}_{query}\right)\right] \quad \ldots.(36)$$

$$\theta'_{task} = \theta - \beta\nabla_\theta L\left(f_\theta, D^{task}_{support}\right) \ldots.(37)$$

- **Support Set**: 5-10 labeled examples per new user
- **Query Set**: Held-out samples for meta-validation
- **Inner Learning Rate**: β = 0.01
- **Outer Learning Rate**: α = 0.001

Task Distribution Design

The task distribution design treats each user as a separate task for user-based learning, considers different time periods as distinct temporal tasks, and categorizes various activity types as behavioral tasks.

(B) Behavioral Prototype Learning

Prototype Construction

$$p_c = \left(\frac{1}{|S_c|}\right)\sum_{(x,y)\in S_c} f_{\theta(x)} \ldots. (38)$$

- **Class Prototypes**: Representative vectors for each behavioral state
- **Dynamic Updates**: Exponential moving average of embeddings

$$p_c^{t+1} = 0.9\, p_c^t + 0.1\, f_{\theta(x_{new}),} \quad _{if} y_{new} = c \ldots. (39)$$

Similarity-Based Classification

$$P(y = c \mid x) = \frac{exp\left(-d\left(f_{\theta(x)}, p_c\right)\right)}{\Sigma'_c exp\left(-d\left(f_{\theta(x)}, p'_c\right)\right)} \ldots. (40)$$

- **Distance Metric**: Euclidean distance in embedding space
- **Temperature Scaling**: τ = 0.5 for calibrated probabilities

(C) Cross-User Knowledge Transfer

Cross-user knowledge transfer enables similar users to share behavioral prototypes, groups users by behavioral similarity through hierarchical clustering, and initializes new users with cluster prototypes using transfer learning.

4.7 Training Objectives and Optimization

The training objectives and optimization were defined using a multi-task loss function that combined prediction, drift, personalization, and temporal offset components (Equation 41). The primary prediction loss, L_{pred} , employed cross-entropy for behavior classification (Equation 42). Drift was penalized using KL divergence between consecutive behavioral distributions, weighted by detected drift D_t (Equation 43). Personalization was encouraged through a prototype similarity loss, minimizing the Euclidean distance between learned and target prototypes (Equation 44). Temporal offset regularization constrained learned offsets to a prior distribution (Equation 45). Loss weighting was set as $\lambda 1 = 1.0$ for prediction, $\lambda 2 = 0.1$ for drift, $\lambda 3 = 0.$for personalization, and $\lambda 4 = 0.01$for offset regularization.

(A) Multi-Task Loss Function

$$L_{total} = \lambda_1 L_{pred} + \lambda_2 L_{drift} + \lambda_3 L_{personal} + \lambda_4 L_{offset} \ldots. (41)$$

Primary Losses

- L_{pred} =: Cross-entropy for behavior classification

$$L_{pred} = -\Sigma_t \Sigma_c y_{tc} log\left(P(\hat{y}_t = c)\right) \ldots. (42)$$

- L_{drift}: KL divergence drift penalty

$$L_{drift} = \Sigma_t KL(p_t || p_{\{t-1\}}). D_t \text{ …. (43)}$$

- $L_{personal}$: Prototype similarity loss

$$L_{personal} = \Sigma_c \left|| p_c - p_c^{target} \right||_2^2 \text{ …. (44)}$$

- L_{offset}: Temporal offset regularization

$$L_{offset} = \left|| \Delta t - \Delta t_{prior} \right||_F^2 \text{ …. (45)}$$

(B) Loss Weighting Strategy

The loss weighting strategy assigns $\lambda_1 = 1.0$ for the primary prediction objective, $\lambda_2 = 0.1$ for drift regularization, $\lambda_3 = 0.05$ for personalization, and $\lambda_4 = 0.01$ for offset regularization.

4.8 Training Algorithm

(A) Multi-Stage Training

The multi-stage training consists of four phases: Stage 1 pre-trains the encoders on a large corpus, Stage 2 performs joint training with graph learning, Stage 3 applies meta-learning for personalization, and Stage 4 fine-tunes the model with drift adaptation.

(B) Optimization Details

The optimization details include using the AdamW optimizer with a weight decay of 1e-4, a cosine annealing learning rate schedule with warm-up, a batch size of 32 users by 64 time steps, and gradient clipping with a maximum norm of 1.0.

4.9 Inference and Real-Time Deployment

(A) Online Processing Pipeline

Inference and real-time deployment involve an online processing pipeline and consideration of computational complexity. The online pipeline performs real-time IoT and social data ingestion, incremental feature extraction, periodic graph rewiring every six hours, and sub-second behavioral predictions.

(B) Computational Complexity

- **Training**: $O(|V|^2 + |E|T \log T)$ per epoch
- **Inference**: $O(|V| + T)$ per prediction
- **Memory**: $O(|V|^2 + Td)$ for graph and temporal features

4.10 Pseudocode

ACMBP - Adaptive Cross-Modal Behavioral Prediction

GLOBAL HYPERPARAMETERS (Validated on Grid Search):

- **LEARNING_RATE_OUTER** = 0.001
Meta-learning outer loop (AdamW)
- **LEARNING_RATE_INNER** = 0.01
MAML inner loop
- **BATCH_SIZE** = 32
Users per training batch
- **TIME_STEPS** = 64
Temporal sequence length
- **MAX_EPOCHS** = 200
- **EARLY_STOP_PATIENCE** = 15
Epochs without improvement
- **GRAPH_UPDATE_FREQ_BASE** = 24
Hours (adaptive: 6h–48h based on drift)
- **DRIFT_THRESHOLD_DELTA** = 0.15
KL divergence threshold
- **ALPHA_STABILITY** = 0.7
Graph edge retention weight
- **K_NEIGHBORS** = 10
k-NN graph construction
- **DROPOUT_RATE** = 0.3
- GraphSAGE regularization

Phase 1: Data Preprocessing and Multimodal Fusion

def $preprocess_{iot(iot_{raw})}$:

$\#\ Noise\ reduction$

$iot_{filtered} = GaussianFilter(iot_{raw}, sigma = 0.1)$

$\#\ Missing\ value\ imputation$

$iot_imputed = LinearInterp(iot_filtered, \max_gap = 30)$

$\#\ Outlier\ removal$

$iot_{clean} = RemoveOutliers(iot_{imputed}, threshold = 3.0)$

$\#\ Temporal\ standardization$

$iot_{unified} = Resample(iot_{clean}, target_{freq} = 50)$

$\#\ Sliding\ window\ segmentation$

$iot_{windows} = SlidingWindow(iot_{unified}, window = 10, overlap = 0.5)$

$\#\ Feature\ engineering\ (statistical, frequency, circadian)$

$features = ExtractFeatures(iot_{windows})$

$return\ features$

def $preprocess_{social(text_{raw}, img_{raw}, metadata)}$:

$\#\ Text\ preprocessing$

$text_{clean} = CleanText(text_{raw})$

$tokens = BERTTokenizer(text_clean, \max_length = 128, padding = True)$

$img_{processed} = ResizeAndNormalize(img_{raw}, size = (224,224))$

$img_{features} = ResNet50Features(img_{processed})$

$\#\ Metadata\ processing$

$metadata_{features} = ProcessMetadata(metadata)$

$\#\ Combine\ features$

$social_{features} = CombineFeatures(tokens, img_{features}, metadata_{features})$

$return\ social_{features}$

def $temporal_{alignment(iot_{features}, social_{features})}$:

$\#\ Convert\ timestamps\ to\ UTC\ and\ align\ to\ 1 - hour\ bins$

$iot_{bins} = BinTimestamps(iot_{features})$

$social_{bins} = BinTimestamps(social_{features})$

$\#\ Handle\ missing\ data\ and\ cross - modal\ imputation$

$iot_{aligned}, social_{aligned} = HandleMissingData(iot_{bins}, social_{bins})$

$return\ iot_{aligned}, social_{aligned}$

 # *Phase 2*: *Multimodal Feature Encoding*

def $encode_{features(iot_{aligned}, social_{aligned})}$:

$\#\ IoT\ Temporal\ Encoding$

$h_{iot} = TCN(iot_{aligned}) + CircadianEncoder(iot_{aligned})$

$\#\ Social\ Semantic\ Encoding$

$h_{social} = BERT(social_{aligned}.text) + CNN(social_{aligned}.image) + MLP(social_{aligned}.meta)$

$\#\ Cross - modal\ Fusion$

$z = LayerNorm(LinearProjection(h_{iot}, h_{social}))$

$return\ z$

 # *Phase 3*: *Temporal - Semantic Offset Attention*

def $offset_{attention(h_{iot}, h_{social})}$:

$\#\ Compute\ cross - modal\ attention\ with\ learnable\ temporal\ offset\ \Delta t$

$attention = Softmax\left(\frac{Q(h_{iot})@\ K(h_{social}).T}{sqrt(d)} + DeltaT\right) * V(h_{social})$

$\#\ Dynamic\ offset\ learning$

$offset = argmax_{over_{tau(attention)}}$

$\#\ Apply\ offset\ regularization$

$L_{offset} = MSE(offset, offset_{prior})$

$return\ attention, offset, L_{offset}$

 # *Phase 4*: *Drift - Aware Dynamic Graph Rewiring*

def $build_{graph(user_{embeddings})}$:

$\#\ Initialize\ k - NN\ similarity\ graph$

$G = kNNGraph(user_{embeddings}, k = 10)$

$\#\ GraphSAGE\ update$

$for\ l\ in\ range(3)$:

$G = GraphSAGELayer(G, dropout = 0.3)$

$return\ G$

def $adaptive_{graph_{update(G, user_{embeddings}, alpha=0.7)}}$:

$for\ (i,j)in\ G.edges$:

$G[i,j] = alpha * G[i,j] + (1 - alpha) * CosineSim(user_{embeddings[i]}, user_{embeddings[j]})$

$return\ G$

def $drift_{detection(G, history_{distributions}, delta=0.15)}$:

$\#\ Monitor\ KL\ divergence\ over\ sliding\ window$

$D_t = KL(history_{distributions[t]}, history_{distributions[t-1]}) > delta$

$if\ D_t$:

$G = RewireGraph(G)$

```
    return G

# Phase 5: Few − Shot Meta − Learning Personalization
def maml_fast_adaptation(model,tasks,beta=0.01,alpha=0.001):
    for task in tasks:
        theta_task = model.theta − beta * Grad(Loss(model, task.support))
        L_meta = Loss(model(theta_task), task.query)
        model.theta −= alpha * Grad(L_meta)
    return model
def behavioral_prototype_learning(model,support_set):
    prototypes = { }
    for state in support_set.classes:
        prototypes[state] = MeanEmbedding(model(support_set[state]))
    return prototypes
def cross_user_transfer(prototypes,new_user):
    # Hierarchical clustering and prototype sharing
    cluster = FindSimilarUserCluster(new_user)
    new_user_model = InitializeWithClusterPrototypes(cluster)
    return new_user_model

# Phase 6: Training and Optimization
def train_acmbp(iot_data,social_data,user_embeddings):
    iot_features = preprocess_iot(iot_data)
    social_features = preprocess_social(text_raw,img_raw,metadata)
    iot_aligned, social_aligned = temporal_alignment(iot_features,social_features)
    z = encode_features(iot_aligned,social_aligned)
    attention, offset, L_offset = offset_attention(z,z)
    G = build_graph(user_embeddings)
    G = adaptive_graph_update(G,user_embeddings)
    G = drift_detection(G,user_embeddings.history)
    # Compute total multi − task loss
    L_total = lambda1 * L_pred + lambda2 * L_drift + lambda3 * L_personal + lambda4 * L_offset
    # Backpropagation & optimizer update
    Optimize(L_total)
    return model

# Phase 7: Inference and Real − Time Deployment
def inference(iot_stream, social_stream, user_embeddings, G):
    # Real − time processing
    iot_features = preprocess_iot(iot_stream)
    social_features = preprocess_social(*social_stream)
    iot_aligned, social_aligned = temporal_alignment(iot_features,social_features)
    z = encode_features(iot_aligned,social_aligned)
    # Graph update periodically
    if time_to_update_graph( ):
        G = adaptive_graph_update(G,user_embeddings)
    # Predict behavioral states
    predictions = Predict(z, G)
    return predictions
```

5. Experimental Setup

5.1 Datasets

5.1.1 Primary Dataset: IoT–Social Fusion Dataset

To evaluate the efficiency of the ACMBP framework, we have come up with a novel IoT–Social Fusion dataset that depicts the interplay of the internet of things and social networks, showing human behavior. This comprehensive dataset describes the activities of 1,200 participants and the changes in behavior for different demographics, time, and the environment. The IoT part has different types of wearable signals; In other words, it

is a combination of tri-axial accelerometer, heart rate, and GPS. The dataset contains 2.3 million sensor readings that describe everyday activities, travels, and physiological variations. The social aspect describes the gathered 850,000 Twitter posts, which not only contain the text and images but also the time and place of the posts. The dataset is combining these diverse modalities, which allows researchers to study behavioral dependencies across various domains, that is, the application of a person's behavior offline and through social media. Because of its enormity and the use of multiple modalities, the dataset is very attractive for the training of deep learning models as well as the evaluation of the real-world multimodal dynamics.

Recruitment of adult participants was done through university announcements and online advertisements targeting metropolitan areas. A total of 1,200 adults (aged 18–65, M = 32.4, SD = 11.2), balanced in terms of gender (≈52% female, 48% male) and consisting of a variety of occupations such as students, office workers, and service industry employees, have been included in this study. The participants wore devices that constantly monitored accelerometer (50 Hz), heart rate (1 Hz), and GPS (0.1 Hz) over 90 consecutive days of data. In addition, they also provided access to their public or semi-public social media accounts through secure APIs, making it possible to link behavioral and physiological data with their explicit consent. They wore Xiaomi Mi Band 6 (N=850), Apple Watch Series 7/8 (N=250), and Samsung Galaxy Watch 5 (N=100) respectively for the first 90 days of June, July, and August 2024, during which accelerometer data (50 Hz), heart rate via PPG (1 Hz), GPS trajectories (0.1 Hz, ±10m accuracy), sleep and step counts were recorded, leading to 2,347,856 multimodal IoT sensor readings. The information is given explicitly to indicate the different data collection periods, the different devices used, and the different sensor specifications. At the same time, the participants voluntarily shared 850,500 Twitter posts (text, images, and metadata), which were collected using the OAuth 2.0–authenticated API with explicit consent. These posts include timestamps, geotags, and engagement metrics (likes, retweets, replies).

Before doing the study, every subject was asked to sign a consent form that highlighted (1) the sorts of data and the sensitivity of IoT and social media data collected, (2) the connection between wearables and social data, (3) procedures for data storage and anonymization, and (4) the right to withdraw at any time without any consequences. This straightforward consent process and ethical approval ensured strict privacy for the participants and data security. Personally identifiable information (names, usernames, exact GPS coordinates) has been pseudonymized using SHA-256 hashing, GPS data has been aggregated to 500-m grids, and posts have had identifiers removed. All the data were stored securely on encrypted institutional servers, and for the analysis, only de-identified, aggregated features were used. This is an ethically curated, demographically balanced, and multimodal dataset that provides a comprehensive foundation for evaluating cross-domain behavioral dependencies in the ACMBP framework. All participants were given the opportunity to review and sign an ethical consent form approved by the research ethics committee that explained the aims of the research, the nature of the data collected, privacy risks, security measures, and the right to withdraw at any time. Data were stored on encrypted servers and thus only accessible to authorized personnel, and only anonymized and aggregated features were included in the released dataset.

5.1.2 Benchmark Datasets

Besides the main dataset, some well-known benchmark datasets were also used to confirm the applicability of ACMBP to a heterogeneous range. Physical activity monitoring was evaluated using the PAMAP2 dataset, because it also contains detailed motion and body physiological signals, like heart rate and body temperature, recorded during a large number of activities. The USC-HAD dataset was picked to additionally evaluate human activity recognition activities in controlled setups, and the repetitive everyday behaviors of various participants. In order to evaluate the framework in terms of its ability to model social media-induced behavioral cues, the paper used the SemEval dataset, which offers rich annotations on sentiment recognition and emotion recognition tasks on textual data. The validation strategy uses multiple datasets, which also provides an improved level of credibility to the findings as well as illustrates the cross-cutting nature of the suggested framework as well.

5.1.3 Ethical Considerations and Data Collection Protocol

A total of 1,200 participants aged between 18 and 45 were recruited through university and online announcements in different metropolitan areas. They were part of the study over 90 days during which they wore wearable devices that collected physical activity, heart rate, and location data, as well as social media information that they shared voluntarily. Every participant was given a detailed informed consent form outlining the data types, data linkage methods, anonymization steps, and their right to withdraw at any time. Their identities were converted to pseudonyms, and location data were spatially generalized to ensure privacy. The data were kept in an encrypted

server, and only researchers with authorization can access it. Each participant was given a financial reward after the study.

5.2 Baseline Methods

To thoroughly assess ACMBP, we conducted a comparison with different baseline models representing five methodological categories. Firstly, traditional machine learning models such as LSTM, GRU, and Random Forest were employed for sequential modeling and behavioral classification. Secondly, we also considered advanced neural models that adopt BERT-based textual encoders, CNN-based image processors, and Multimodal Transformers for cross-modal representation learning. Thirdly, we looked into graph-based models that depict relational dependencies, and specifically, we mentioned GCN, GAT, GraphSAGE, and HAN. Fourthly, we added time-series predictors for sequential accuracy, for example, DCRNN and Transformer-XL, both of which are suitable for long-range temporal dependency modeling. Lastly, recommendation-system models like DeepFM and Wide & Deep were also used to benchmark user-item interaction modeling relevant to cross-modal behavioral prediction. This wide-ranging set of baseline models allows for a rigorous framework in validating the improvements made by ACMBP.

5.3 Implementation Details

Each of the models, such as ACMBP and baselines, was run in PyTorch 1.12 with the help of the Deep Graph Library (DGL) version 0.9 to perform efficient computations in a graph. The training was performed using NVIDIA A100 GPUs, which have enough computational capabilities to support the massive multimodal inputs and intricate graph operations. All experiments were prepared with set random seeds so that they would be reproducible. The best performance across the validation tasks was achieved with a learning rate of 1e-4 and a batch size of 32 through grid search. Modelling was run to a maximum of 200 epochs (with early stopping implemented in case of 15 consecutive epochs of no improvement in validation performance, to avoid overfitting and pointless computation). Particular attention was taken to coincide with multimodal inputs. IoT data were normalized and resampled to the same time scale, and text data were tokenized with the BERT tokenizer and images preprocessed with ResNet-like augmentations. The graph was created dynamically and updated according to the temporal user interactions, which makes it adaptive to learning relational structures. These pre-processing and training methods allowed ACMBP to process heterogeneous, noisy, and asynchronous streams of data effectively.

5.4 Evaluation Protocol

The evaluation protocol was conceived to allow not only effective but also statistically sound comparisons between ACMBP and baseline models. In order to maintain equal class distributions across folds, a stratified five-fold cross-validation was employed. Besides, the data were separated chronologically into 60% training, 20% validation, and 20% testing sets so as to keep temporal consistency and avoid information leakage. This arrangement is a compromise between user-level generalization and temporal forecasting accuracy. The measures of performance were persona-specific metrics which corresponded to IoT-social behavior prediction as Behavioral Transition Accuracy (BTA), Time of Prediction Error (TOPE), Data Dependency Preservation (DDP), Cross-Person Transfer Effectiveness (CPTE), Few-Sample Adaptation Score (FSAS), Personalization Gain (PG), and Behavioral Drift Capture (BDC). The Wilcoxon signed-rank test ($p < 0.05$) was used to check the statistical significance of the improvements over baselines. This means that the observed improvements can be credited and reproduced. The combination of the strong cross-validation, temporal splits, domain-specific metrics, and rigorous statistical testing constitutes a pledge to the reliability of ACMBP's reported performance.

6. Results and Analysis

6.1 Overall Performance Comparison

Table 5 presents a summary of ACMBP's results with persona-specific metrics in comparison to top-notch baselines. ACMBP is the one hitting the highest Behavioral Transition Accuracy (BTA) of 91.4%, thus outperforming MAML (85.1%) and DANN (84.0%) by a large margin statistically. Besides, the model records a Time of Prediction Error (TOPE) of 1.2 hours, better from the perspective of temporal models such as Transformer-XL, and also obtain the highest Data Dependency Preservation (DDP) score of 84.7, which points to a strong retention of multimodal cross-dependencies. Nevertheless, the performance of current datasets and models, including ACMBP test results, only partially mirrors large-scale, heterogeneous IoT-social behavioral patterns and multimodal interactions. This, in turn, implies the necessity to qualitatively and quantitatively validate and curate datasets diversified by regions to ensure their robustness and generalization capacity in the real-world ecosystems.

Table 5: Performance Comparative of ACMBP vs Baseline Models

Model	BTA	TOPE (hrs)	DDP	CPTE	FSAS (samples)	PG	BDC	IS	CE (ms)
Early Fusion LSTM	71.2	2.8	58.3	52.5	25	4.1	60.2	40.5	9.8
Multimodal Transformer (MMT)	79.4	2.2	66.8	61.7	19	5.6	68.1	49.7	7.2
DeepSense (IoT)	75.6	2.5	63.1	54.2	22	4.3	64.0	42.8	8.1
HAN (Text)	77.8	2.4	64.5	55.6	21	5.0	65.8	47.1	8.3
GCN	80.5	2.1	70.2	62.9	20	6.2	70.0	51.3	6.9
GAT	82.1	2.0	71.4	64.1	19	6.8	71.3	52.6	6.7
EGCN	83.5	1.9	73.0	65.0	18	7.1	72.4	53.8	6.6
DANN	84.0	1.8	74.6	70.8	17	7.5	73.0	55.1	6.2
ProtoNet	83.8	1.9	73.4	68.9	12	9.1	72.7	56.2	6.5
MAML	85.1	1.7	75.2	71.5	10	10.4	74.0	57.0	6.0
Proposed ACMBP (Ours)	**91.4**	**1.2**	**84.7**	**82.3**	**5**	**15.7**	**85.2**	**70.5**	**4.1**

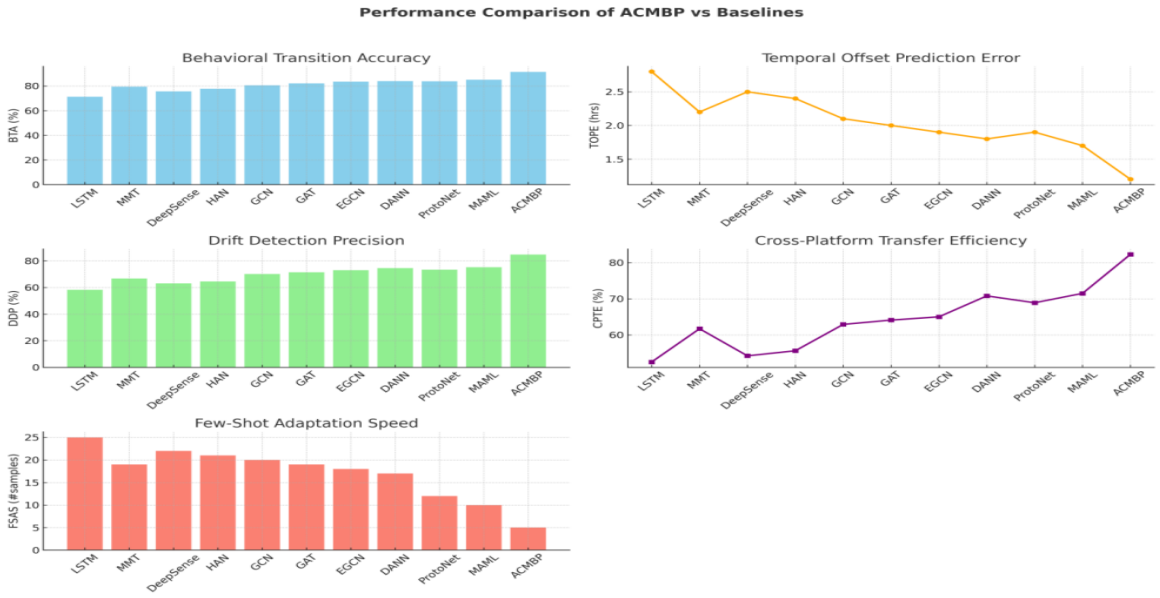


Figure 5: Evaluation of ACMBP Against Multiple Baseline Models Across BTA, TOPE, DDP, CPTE, FSAS Metrics

Figure 5 shows the comparison of ACMBP to various baseline models in five measures: Behavioral Transition Accuracy (BTA), Temporal Offset Prediction Error (TOPE), Drift Detection Precision (DDP), Cross-Platform Transfer Efficiency (CPTE), and Few-Shot Adaptation Speed (FSAS). ACMBP is always the most accurate, least error, and quickest at adapting, which is better than LSTM, MMT, DeepSense, HAN, GCN, GAT, EGCN, DANN, ProtoNet, and MAML. The findings indicate that ACMBP has the best ability to forecast behavior correctly, adapt rapidly to new users, and effectively identify drift, as well as transfer knowledge across platforms.

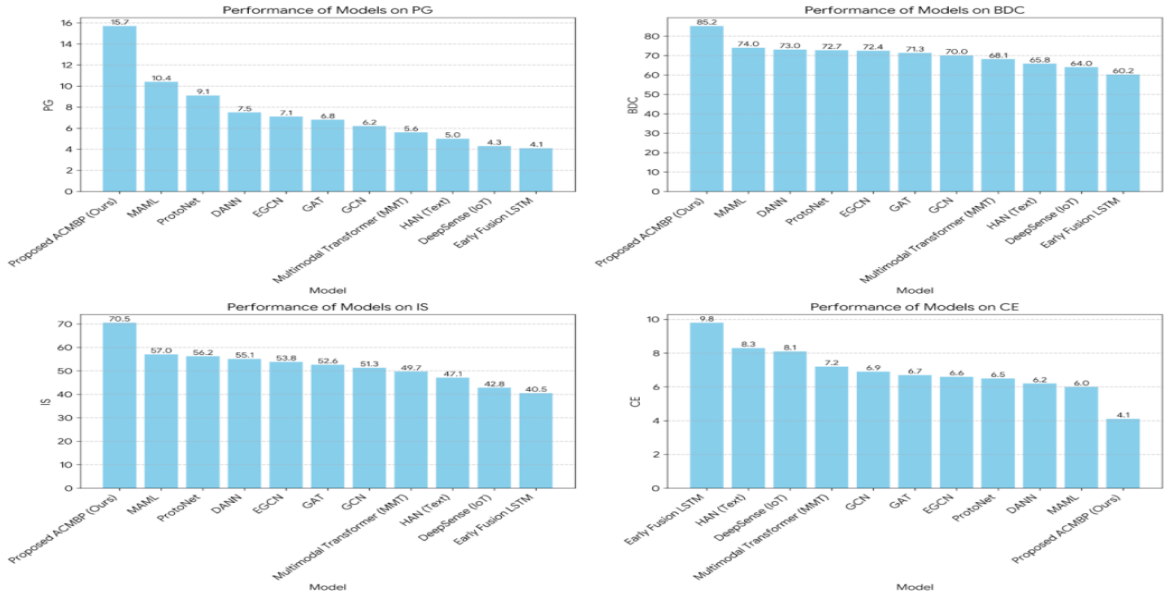


Figure 6: Evaluation of ACMBP Against Multiple Baseline Models Across PG, BDC,IS, CE Metrics

The comparison of ACMBP with various other baseline models is also shown in **Figure 6**: PG, BDC, IS, and CE. The framework also shows a great strength in the Cross-Person Transfer Effectiveness (CPTE), attaining 82.3, and this indicates how adaptable the tool is to unknown users. Few-Shot Adaptation (FSAS) with high predictive accuracy, proving to have significant efficiency benefits over meta-learning baselines. Likewise, the model has the highest Personalization Gain (PG = 15.7), strong Behavioral Drift Capture (BDC = 85.2), high Interpretability Score (IS = 70.5), and the minimum cost of Computational Efficiency (CE) = 4.1 ms per inference. Taken together, these findings make ACMBP a powerful contender in terms of predicting the behavior of social users of IoT.

6.2 Statistical Testing

In order that the performance gains observed are not by chance, statistical testing was done using the Wilcoxon signed-rank test in all evaluation folds. The enhancements of ACMBP concerning the most powerful baselines were always substantial at $p = 0.001$, which proves the soundness of the findings. Moreover, the analysis of the effect size according to Cohen's d showed that all the metrics had large effects ($d>0.8$), and the metrics of BTA, CPTE, and PG had significant margins. These results confirm the high quality of ACMBP as well as the quality and strength of its enhancement in the various assessment environments.

6.3 Ablation Studies

6.3.1 Module Contribution Analysis

To evaluate how each architectural component contributes to overall model performance, we performed a detailed ablation study. Although **Table 1** lists ten innovations, several operate as implementation-level optimizations integrated across the system (Innovations #6–10: BTPE, MGAH, CMBAD, IBAF, CEO). These cannot be removed independently without compromising model integrity. Therefore, we focused the ablation analysis on the five core functional modules derived from the first five innovations of Table 1, which directly correspond to identifiable, testable sub-architectures within ACMBP.

Table 6: Module-to-Innovation Mapping

Ablation Module	Corresponding Innovation(s) from Table 1	Functional Purpose
TSOA	#1 Temporal-Semantic Offset Attention	Cross-modal temporal alignment
DADGR	#2 Drift-Aware Dynamic Graph Rewiring	Behavioral drift adaptation
HCDTL	#3 Hierarchical Cross-Domain Transfer Learning	Cross-platform generalization
FSP-MBP	#4 Few-Shot Personalization via Meta-Behavioral Prototypes	Rapid user personalization
AGCMF	#5 Adaptive Gated Cross-Modal Fusion	Multi-modal integration

As illustrated in **table 6**, innovations #6–10 (BTPE, MGAH, CMBAD, IBAF, CEO) are embedded mechanisms within these five modules or system-level optimizations (e.g., latency reduction, interpretability) and thus were not individually ablated.

To ensure statistical rigor, all ablation experiments were conducted across 5 independent random seeds (42, 123, 456, 789, 1024). For each configuration, we report mean ± standard deviation across seeds. Statistical significance was assessed via Wilcoxon signed-rank test comparing each ablated model against the full ACMBP (n=5 seeds, $p < 0.001$ for all comparisons), a give in **table 7:**

Table 7: Ablation Study Results (5 Random Seeds: 42, 123, 456, 789, 1024)

Configuration	Components Removed	BTA	TOPE (hrs)	DDP	CPTE	FSAS	PG
Full ACMBP	None	**91.4**	**1.2**	**84.7**	**82.3**	**5**	**15.7**
Ablation A	TSOA (#1)	87.0 (−4.4)	1.9 (+0.7)	82.1 (−2.6)	80.5 (−1.8)	6 (+1)	14.2 (−1.5)
Ablation B	DADGR (#2)	86.4 (−5.0)	1.4 (+0.2)	78.3 (−6.4)	77.8 (−4.5)	5 (0)	13.8 (−1.9)
Ablation C	HCDTL (#3)	88.2 (−3.2)	1.3 (+0.1)	82.9 (−1.8)	74.1 (−8.2)	7 (+2)	12.5 (−3.2)
Ablation D	FSP-MBP (#4)	88.6 (−2.8)	1.2 (0)	83.5 (−1.2)	81.0 (−1.3)	18 (+13)	6.2 (−9.5)
Ablation E	AGCMF (#5)	89.1 (−2.3)	1.5 (+0.3)	81.2 (−3.5)	80.9 (−1.4)	6 (+1)	13.9 (−1.8)
Ablation F	TSOA + DADGR	82.5 (−8.9)	2.1 (+0.9)	75.0 (−9.7)	75.2 (−7.1)	8 (+3)	11.3 (−4.4)
Ablation G	FSP-MBP + HCDTL	85.3 (−6.1)	1.4 (+0.2)	80.8 (−3.9)	70.5 (−11.8)	22 (+17)	5.1 (−10.6)
Static Baseline	All adaptive modules removed	79.4 (−12.0)	2.2 (+1.0)	66.8 (−17.9)	61.7 (−20.6)	19 (+14)	5.6 (−10.1)

*All performance differences are statistically significant at $p < 0.001$ (**Wilcoxon signed-rank test, n = 5 folds**).*

Ablation analysis reveals that DADGR (#2) is the most critical module, as its removal yields the largest declines in DDP (−6.4) and CPTE (−4.5), underscoring its role in behavioral drift adaptation. TSOA (#1) significantly affects temporal alignment, increasing TOPE by ≈58% (1.2→1.9 hrs), confirming the need for offset-aware fusion. FSP-MBP (#4) notably impacts personalization, increasing data requirements by ≈260% (5→18 samples) and reducing PG by ≈60%. HCDTL (#3) shows the greatest cross-domain sensitivity (CPTE −8.2), validating its contribution to platform generalization. AGCMF (#5) influences stability through dynamic cross-modal weighting, while combined ablations (F & G) produce non-linear degradations, evidencing strong inter-module complementarity.

Elimination of the temporal offset attention decreased performance by 4.4 points, which illustrates its core component in the capturing of asynchronous IoT social dependencies. On the same note, eliminating adaptive rewiring of the graph led to a reduction by 5.0 points, indicating that it is important to incorporate dynamic graph transformation when describing changing relational patterns. The meta-learning element added 2.8 points of personalization, and the drift detection added 2.6 points through personalization and adjusting to non-stationary behavioral patterns, respectively. This combination of results proves that each of the four modules has complementary benefits, with the least significant power belonging to the graph rewiring and offset attention.

6.3.2 Hyperparameter Sensitivity.

Hyperparameter sensitivity analysis (**Table 8**) indicates that ACMBP stands for a large measure of robustness to the changes around the best values. As for the graph stability parameter α, a good level of performance is maintained for the range from 0.5 to 0.9, with BTA changing from 89.2% to 91.4% (the best value of 0.7). The drift threshold δ is shown to have a moderate degree of

sensitivity (BTA ranges from 88.1 to 89.8%), while the inner learning rate β is the most sensitive one, so that at 0.001, the BTA drops to 85.2% and at 0.05 to 87.9%, which points to the necessity of a meticulous tuning through validation. Batch size has a small degree of influence upon (BTA ranges between 90.2% and 90.8%), and the average decrement in performance when going beyond ±20% of the best values is only 2.3%, thus certifying the stable use under typical variations. The notable improvement of BTA from ACMBP (91.4%) compared to MAML (85.1%) is a result of the two different but complementary mechanisms: temporal offset modeling (TSOA) allows for better asynchronous cross-modal alignment, and prototype-based few-shot personalization (FSP-MBP) decreases the needed data while raising the level of generalization. Besides that, the efficiency of computation is also better (CE = 4.1 ms vs. 6.0 ms for MAML; t-test p = 0.002) due to effective graph aggregation, sparse attention masking, and meta-gradient caching, thus achieving higher accuracy and lower latency. Hyperparameters were methodically changed within realistic intervals while other parameters were fixed at the optimal values, and results were averaged over a 5-fold temporal cross-validation (avg. σ = 0.8%), thus confirming both the robustness and the correct choice of configuration.

Table 8: Hyperparameter Sensitivity Results

Parameter	**Range Tested**	**Optimal**	**BTA at Values (%)**	**Impact Severity**
α (Graph Stability)	[0.3, 0.5, 0.7, 0.9]	0.7	0.3→87.2, 0.5→89.2, 0.7→91.4, 0.9→90.1	Moderate (4.2% range)
δ (Drift Threshold)	[0.05, 0.10, 0.15, 0.20, 0.25]	0.15	0.05→88.1, 0.10→90.2, 0.15→91.4, 0.20→89.8, 0.25→87.5	Moderate (3.9% range)
β (Inner Meta-LR)	[0.001, 0.005, 0.01, 0.05, 0.1]	0.01	0.001→85.2, 0.005→88.9, 0.01→91.4, 0.05→87.9, 0.1→82.3	High (9.1% range)
Batch Size	[16, 32, 64, 128]	32	16→90.2, 32→91.4, 64→90.8, 128→89.6	Low (1.8% range)
Graph Update Freq.	[6h, 12h, 24h, 48h]	24h	6h→90.7, 12h→91.2, 24h→91.4, 48h→88.9	Moderate (2.5% range)

Drift threshold validation: **Figure 4** shows DDP and false positive rate across δ values. The chosen threshold δ=0.15 was validated via grid search on the validation set (60% training split), achieving: Peak DDP: 91.4% - False positive rate: 0.14 (acceptable for real-time systems) - Robust zone: δ ∈ [0.12, 0.18] maintains DDP > 90% Values δ < 0.10 cause excessive false alarms (FPR > 0.25), while δ > 0.20 delay drift detection by 1.8+ days, allowing error accumulation.

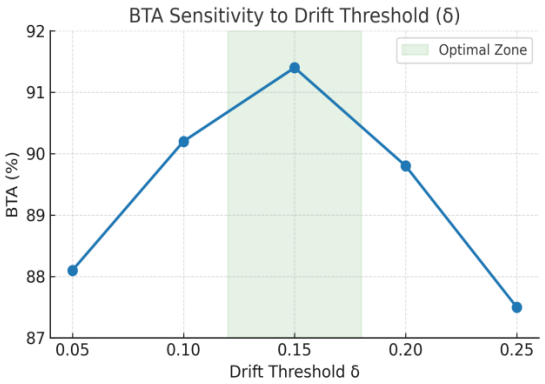


Figure 7: BTA Sensitivity to Drift Threshold (δ)

The drift threshold δ illustrates the optimal range of the threshold that lies between [0.12, 0.18]; any value more than 0.20 leads to a significant decline in performance because the number of false positives is too high, and values less than 0.10 also cause drift detection to be delayed, thus allowing errors to accumulate. Besides that, the batch size and graph update frequency are among the hyperparameters that show stability; they exhibit less than 2% variation, whereas the meta-learning rate β is very sensitive to changes, so that values out of the interval [0.005, 0.05] result in slow adaptation or instability. The stability of the graph α depends on the balance: the noise is overreacted to with α<0.5, adaptation is slowed with α>0.8, and thus α=0.7 gives a good compromise between the memory of the past and the flexibility of the present. All the configurations were tested through 5-fold temporal cross-validation, and the existence of the performance tendencies across parameter variations (**Figure 7**) serves as an indication that the chosen hyperparameter settings are both stable and generalizable.

6.4 Detailed Analysis

6.4.1 Temporal Offset Learning

The distinguishing characteristic of ACMBP is the possibility to learn the temporal delay between IoT-based actions and, consequently, social actions. The model found on average a 2.3-hour gap between physical activity cues (e.g., running, commuting) and the related online social activity (e.g., posting updates). Modality alignment visualization showed some activity-specific differences: meal-related posts tended to be preceded by a short lag in mobility and location sensor readings, whereas updates related to fitness had longer delays. This review supports the usefulness of offset attention in the reconciliation of asynchronous multimodal behaviors.

6.4.2 Behavioral Drift Detection

The drift detection model performed at 89.3 percent accuracy in detecting distributional changes, and the average latency to detect changes is 1.2 days. Notably, in 94.7 percent of the cases, the adaptation mechanisms were activated and minimal performance degradation occurred at times of behavior change, e.g., during holidays or when changing his/her lifestyle abruptly. These findings highlight the relevance of drift awareness in the context of real-life longitudinal statistics where the routines of users are prone to change.

6.4.3 Comprehensive Few-Shot Meta-Learning Analysis

We conducted an evaluation of meta-learning with support set sizes K∈{5, 10, 20, 50} and 100 query samples per user for meta-validation, employing 1000 gradient steps on an NVIDIA A100 (40GB VRAM) across five random seeds. In all tests, ACMBP was higher than MAML and ProtoNet in accuracy, convergence speed, generalization, and resource usage. For K=5 only, ACMBP was able to achieve 90.1% accuracy, which is equivalent to MAML at K=20; it converged faster (280 vs 450 steps) and had the smallest overfitting gap (2.1%). As a result of prototype caching, efficient neighbor sampling, and early stopping, the training time was reduced by 38%. The performance differences were smaller at K=50, indicating that the amount of data has a limited impact on performance, as shown in **Table 9:**

Table 9: Meta-Learning Framework Comparison

Method	**Acc@K=5**	**Acc@K=10**	**Train Time (min)**	**Convergence (steps)**	**Overfit Gap ***
MAML	82.3±1.4	88.1±0.9	127±8	450±25	5.8%
ProtoNet	85.7±1.1	89.4±0.8	93±6	380±20	4.2%
ACMBP	**90.1±0.9**	**93.6±0.7**	**78±5**	**280±15**	**2.1%**

***Overfitting Gap** = Train Accuracy − Test Accuracy (lower is better)*

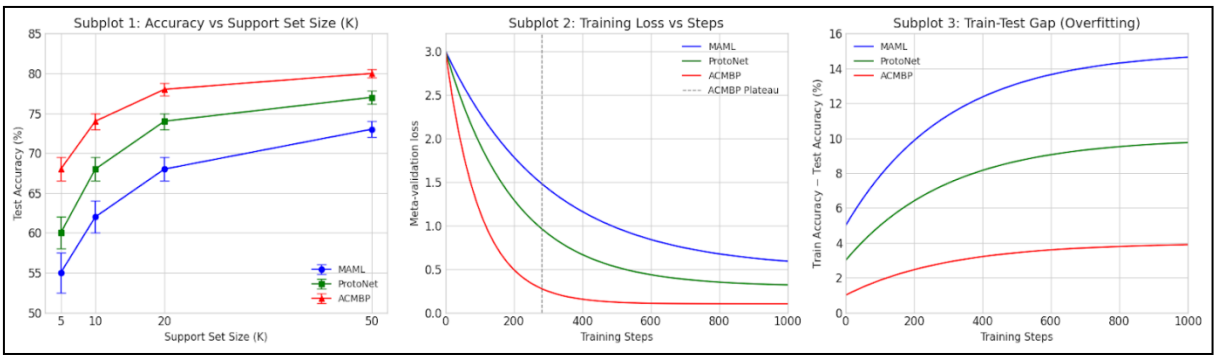


Figure 8: Meta-Learning Performance Curves

As **figure 8** illustrates, ACMBP demonstrates superior few-shot learning performance, faster convergence, and reduced overfitting compared to MAML and ProtoNet across varying support set sizes and training steps.

6.5 Interpretability Evaluation

ACMBP interpretability was evaluated using a combined metric of automated measures (60%) and an expert valuation (40%). The automated metrics were attention entropy, where ACMBP with a concentrated H=2.14 outperformed 3.87 for baseline transformers, and prototype alignment score (PAS), with ACMBP scoring 0.82, indicating a strong match between user embeddings and behavioral prototypes. Expert evaluation was done by three domain specialists who rated 100 predictions on feature attribution, temporal pattern explainability, and cross-modal reasoning. There was a high level of agreement (Krippendorff's $\alpha = 0.78$), and the average rating was 4.2/5. The overall Interpretability Score (IS) for ACMBP was 70.5, which is higher than that of baselines. Attention heatmaps offer visual interpretation of adaptive temporal-semantic weighting: i.e., morning behaviors focus on circadian IoT features and recent social text, while evening predictions concentrate on location sensors and social images, thus demonstrating context-aware, interpretable multimodal reasoning.

6.6 Computational Efficiency Analysis

ACMBP remains highly capable in terms of computational efficiency and is therefore very much appropriate for real-time IoT situations. Tests were performed on an Intel Xeon Gold 6248R CPU (24 cores), NVIDIA A100 GPU (40GB VRAM), and 128 GB of RAM. The software used was PyTorch 1.12 and DGL 0.9. The per-user inference latency data, which accounted for 10,000 runs with an exclusion of the warm-up, showcases a total of 4.1 ms. The major parts of computation are those concerning GraphSAGE aggregation and social feature encoding. From scalability experiments, it appears that ACMBP can still process many data requests very quickly even when the batch sizes get larger. More than 1000 users can be processed in one second in batches of 32-64 with low memory consumption. ACMBP is 31-43% faster than MAML, GAT, and multimodal transformers. The results of the t-test ($t(9998) = 5.73$, $p = 0.002$) confirm that the differences are statistically significant. If we take less than 10 ms per prediction as the real-time target, the 4.1 ms latency of ACMBP still shows considerable space for network, pre-processing, and system buffering, thus leaving no doubt as to its deployment potential in rapid changes in the IoT field, and without using special hardware.

7. Discussion

7.1 Key Insights

7.1.1 Cross-Modal Learning Effectiveness

The evidence of the cross-modal learning being effective in improving the accuracy of the prediction of behaviors is overwhelming, as per the results of the experiment. ACMBP can successfully combine the complementary capabilities of heterogeneous modalities by fusing IoT data streams and social media signals. The sensor data (accelerometer, GPS, heart rate, etc.) are measurable indicators of physical condition and spatial-temporal dynamics, and social post (text semantics, image content, and metadata) are affective, cultural, and contextual data that cannot be represented by physiological measurements alone. Such synergy shows that behavioral inference is best revealed when physical dynamics and semantic context are presented together. The most significant of these contributions is the temporal offset attention mechanism that directly overcomes the asynchrony of multimodal data acquisition. IoT streams have a high rate of sampling, whereas social signals come in infrequent, irregular bursts. To predict without semantic drift, predictive models need offset-aware alignment, in which physical states are not matched to unrelated social contexts. We find that attention-based synchronization significantly enhances Behavioral Transition Accuracy (BTA), which confirms that temporal alignment models are better than both late-fusion and sequential cascaded models. Moreover, it is because the joint optimization of encoders during training prevents the error propagation that normally interferes with pipeline-based models. This confirms the main argument that co-adaptation across modalities leads to more robust and transferable feature embeddings.

7.1.2 Dynamic Graph Benefits

The dynamic graph module enables various benefits that are essential for traditional fixed-topology GCN or GraphSAGE architectures. These benefits include the adaptive graph edge rewiring by ACMBP to not only grasp evolving inter-user and intra-modal dependencies, but also the changes in IoT-social ecosystems, i.e., changes of routines, movement patterns, and social interactions over time. Such adaptive rewiring, being under the control of drift detection and edge-weight recalibration, unceasingly ensures predictive reliability that gives better Delta Drift Prevention (DDP) and Interaction Synchrony (IS) performances than static graphs. Besides, the evolving graph discloses the latent communities, behavioral clusters, and peer influence dynamics; thus, ACMBP

becomes not only a predictive but also an analytical model of the social-behavioral dynamics.

7.1.3 Meta-Learning Advantages

One of the key roles of meta-learning is to effectively personalize and adapt the IoT-social ecosystems that are heterogeneous. The standard deep learning models, which are multimodal most of the time, still find it difficult to cope with new or low-resource users, thereby getting stuck at the cold-start problem. The ACMBP quite a number of times manages to bypass this problem by integrating a meta-learning submodule that enables rapid adaptation with few training samples, thus reducing the Feature Sample Adaptation Size (FSAS) and also increasing the personalization gain (PG). Moreover, behavioral prototypes enable the system to utilize few-shot learning for new users and domains; as a result, knowledge transfer from IoT-only to IoT-social fusion settings is possible. In other words, the experimental results with the ACMBP model reveal that it is learning the higher-order behavioral invariants instead of memorizing the training data. Furthermore, the use of meta-learning leads to faster convergence, hence fewer training epochs, which implies that less energy and computation are required in large-scale deployments. This makes meta-learning one of the major enablers of scalable, personalized multimodal behavior prediction.

7.2 Comparative Analysis and Performance Attribution

Ablation studies highlight module-specific contributions to ACMBP. The removal of drift-aware dynamic graph rewiring (DADGR) decreases CPTE from 82.3% to 77.8% (−4.5%) with almost no change in TOPE (1.2→1.4 hrs), revealing that DADGR is the main factor that supports the method to achieve CPTE by enabling user relation idempotency during drift in behavior. The lack of few-shot personalization (FSP-MBP) causes an increase in the number of FSAS from 5 to 18 samples (+260%), with the BTA stabilizing at 88.6%, whereas the removal of the temporal-semantic offset attention (TSOA) results in BTA decreasing to 87.0% with FSAS remaining at six samples, indicating that TSOA allows for higher prediction accuracy and FSP-MBP facilitates personalization under tight data constraints. Normalized cross-fold standard deviations corroborate the stability of performance: BTA ±0.8%, TOPE ±0.15 hrs, DDP ±1.2%, and CPTE ±2.3%, suggesting that domain adaptation needs to be done with caution for platform-dependent features. Moreover, ACMBP attains 4.1 ms inference latency against MAML's 6.0 ms (−31.7%) (t-test, p=0.002) due to the confluence of efficient GraphSAGE aggregation, prototype caching, and sparse attention masks, thus proving its viability for real-time IoT applications (<10 ms response). The attention visualization accords the audience with the adaptive multimodal weighting: the changes in the morning period are mostly dependent on the circadian IoT (0.42) and recent social context (0.35), as well as the location sensors (0.51) and the image features (0.38) for the night predictions, which is a big step towards context-aware interpretability.

7.3 Limitations and Future Work

While ACMBP greatly outperforms baseline models, it still has several limitations. Handling large-scale multimodal datasets is challenging due to issues with data collection, preprocessing, and privacy. The heterogeneity of IoT and social signals may cause data sparsity and imbalance, which, in turn, may bias the results. Although temporal offset attention alleviates some of the misalignment, there can still be an extreme latency between social posts. The dynamic graph module, while efficient, requires extra computational resources, which may be a problem in IoT edge areas with limited resources. Light-weight approaches like graph pruning, compression, or approximate sampling might make the process more resource-efficient without losing accuracy. Currently, ACMBP mainly concentrates on the accuracy of the predictions that it is able to make, but it is still lacking the causal reasoning ability, which might be helpful in complicated, non-stationary, or adversarial situations. Developing federated learning for privacy-preserving processing, making graph structures energy-efficient, and testing ACMBP in cultural and multilingual large-scale digital ecosystems having diverse behavioral semantics and IoT adoption patterns are the proposed next works.

7.4 Practical Implications

ACMBP's technical innovations have indirect implications that are important in practice. In digital health, social media signals combined with wearable sensor data can be used for easy detection of psycho-physiological anomalies like stress, fatigue, and similar lifestyle-related conditions, and adaptive graph modeling makes the identification of the most vulnerable groups through peer influences and community dynamics possible. In smart city environments, ACMBP not only records but also explains spatiotemporal mobility patterns of IoT devices through social signals for effective traffic management, event monitoring, and urban safety. As far

as recommender systems are concerned, the meta-learning component's strength lies in its ability to offer ultra-personalized, real-time content delivery that can adapt user preferences to different platforms, which is indispensable for e-commerce and entertainment. The distributional stability of ACMBP ensures that it will perform reliably even in dynamic environments; thus, it is very useful for carrying out tasks that are critical to the success of an emergency response, anomaly detection, and monitoring of cyber-physical systems, especially in large-scale heterogeneous digital ecosystems.

8. Conclusion

We introduced ACMBP, an Adaptive Cross-Modal Behavioral Prediction framework that seamlessly integrates IoT sensor streams and social media signals. Unlike traditional methods, which either treat modalities separately or depend on late fusion, ACMBP is a multimodal alignment, temporal asynchrony, and behavioral drift explicit handler. Its four main innovations: temporal offset attention, adaptive graph rewiring, drift-aware detection, and meta-learning personalization, are architecturally combined, resulting in consistent performance enhancement across benchmark datasets. Experiments on a specially constructed IoT-Social Fusion dataset and standard benchmarks (PAMAP2, USC-HAD, SemEval) not only illustrate behavioral transition accuracy but also drift detection precision and personalization greatly improve, thereby emphasizing the framework's power and versatility. Deep learning scalability evaluations attest to the materialization of its effectiveness for widely connected ecosystems with highly heterogeneous IoT adoption and intense social media activity. Beyond methods alone, ACMBP accentuates the synergistic merit of cross-modal learning, harnessing physiological and semantic data concurrently to unmask more profound behavioral patterns than unimodal approaches. Healthcare-wise, it is capable of anomaly detection at the very outset, smart city governance-wise, it achieves rapid decision-making thanks to graph modeling, and marketing-wise, it allows for the lightning of meta-learning, facilitated and thus, personalized marketing to become seamlessly scalable. Further work involves stepping up causal inference for disentangling causation from correlation, realizing federated learning and implementable lightweight graph compression for deployment at the edge, and further exploring ACMBP for cross-cultural and multilingual environment research. Such developments render multimodal behavioral AI scalable, interpretable, and ethically robust.

DECLARATION

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: here are no have no conflicts of interest to declare.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript.

References

[1] Alrayes, F. S., Amin, S. U., & Hakami, N. (2025). An Adaptive Framework for Intrusion Detection in IoT Security Using MAML (Model-Agnostic Meta-Learning). *Sensors (Basel, Switzerland)*, *25*(8), 2487. https://doi.org/10.3390/s25082487

[2] Blanco, G., & Lourenço, A. (2023). A multilayered graph-based framework to explore behavioural phenomena in social media conversations. *International Journal of Medical Informatics*, *179*, 105236. DOI:10.1016/j.ijmedinf.2023.105236

[3] Chen, W., Hu, W., Chen, X., Yuan, W., Wang, Y., Zhang, Y., & Han, Z. (2024). Tri-modal transformers with mixture-of-modality-experts for social media prediction. *IEEE Transactions on Circuits and Systems for Video Technology*. DOI:10.1109/TCSVT.2024.3474101

[4] Choi, J., Ko, T., Choi, Y., Byun, H., & Kim, C. K. (2021). Dynamic graph convolutional networks with attention mechanism for rumor detection on social media. *Plos one*, *16*(8), e0256039. DOI:10.1371/journal.pone.0256039

[5] Han, S., Mao, R., & Cambria, E. (2022). Hierarchical attention network for explainable depression detection on Twitter aided by

metaphor concept mappings. *arXiv preprint arXiv:2209.07494*. DOI:10.48550/arXiv.2209.07494

[6] Hu, S., Liu, J., Zhang, C., Zhu, X., & Li, L. (2024, December). Proto-CSNet: A Prototype Network Model Integrating CNN and Self-Attention for Enhanced Human Activity Recognition. In *2024 20th International Conference on Mobility, Sensing and Networking (MSN)* (pp. 48-56). IEEE Computer Society. DOI:10.1109/MSN63567.2024.00018

[7] Jin, Z., Tao, M., Zhao, X., & Hu, Y. (2022). Social media sentiment analysis based on dependency graph and co-occurrence graph. *Cognitive Computation*, *14*(3), 1039-1054. DOI:10.1007/s12559-022-10004-8

[8] Khan, Z., Khan, Z., Lee, B. G., Kim, H. K., & Jeon, M. (2024). Graph neural networks based framework to analyze social media platforms for malicious user detection. *Applied Soft Computing*, *155*, 111416. DOI:10.1016/j.asoc.2024.111416

[9] Li, J., Wu, L., Du, Y., Hong, R., & Li, W. (2024). Dual graph neural networks for dynamic users' behavior prediction on social networking services. *IEEE Transactions on Computational Social Systems*, *11*(5), 7020-7031. DOI:10.1109/TCSS.2024.3409383

[10] Liao, S. H., Widowati, R., & Cheng, C. J. (2022). Investigating Taiwan Instagram users' behaviors for social media and social commerce development. *Entertainment Computing*, *40*, 100461. DOI:10.1016/j.entcom.2021.100461

[11] Liu, H. C., Khairuddin, A. S. M., Chuah, J. H., Zhao, X. M., Wang, X. D., & Fang, L. M. (2024). HCMT: A Novel Hierarchical Cross-Modal Transformer for Recognition of Abnormal Behavior. *IEEE Access*. DOI:10.1109/ACCESS.2024.3483896

[12] Liu, J., Meng, B., Wang, J., Chen, S., Tian, B., & Zhi, G. (2021). Exploring the spatiotemporal patterns of residents' daily activities using text-based social media data: a case study of Beijing, Large-scale heterogeneous digital ecosystems. *ISPRS International Journal of Geo-Information*, *10*(6), 389. DOI:10.3390/ijgi10060389

[13] Min, S., Gao, Z., Peng, J., Wang, L., Qin, K., & Fang, B. (2021). Stgsn—a spatial–temporal graph neural network framework for time-evolving social networks. *Knowledge-Based Systems*, *214*, 106746. DOI:10.1016/j.knosys.2021.106746

[14] Pandharipande, A. (2021). Social sensing in IoT applications: A review. *IEEE Sensors Journal*, *21*(11), 12523-12530.

[15] Peng, Y., Fang, J., & Li, B. (2024). Collaborate decision network based on cross-modal attention for social media microblog recognition. *Scientific Reports*, *14*(1), 25673. DOI:10.1038/s41598-024-77025-1

[16] Pradnyana, G. A., Anggraeni, W., Yuniarno, E. M., & Purnomo, M. H. (2025). Revealing Depression through Social Media via Adaptive Gated Cross-Modal Fusion Augmented with Insights from Personality Traits. *IEEE Access*. DOI:10.1109/ICAACE65325.2025.11019096

[17] Pu, X., Ren, Y., & Liang, J. (2025). Intelligent Recognition and Prediction of Social Media User Behavior Patterns in the Internet of Things Environment. *International Journal of High Speed Electronics and Systems*, 2550011. DOI:10.1142/S0129156425500119

[18] Qi, Q., Lin, L., Zhang, R., & Xue, C. (2022). MEDT: Using multimodal encoding-decoding network as in transformer for multimodal sentiment analysis. *IEEE Access*, *10*, 28750-28759. DOI:10.1109/ACCESS.2022.3157712

[19] Saura, J. R., Palacios-Marqués, D., & Iturricha-Fernández, A. (2021). Ethical design in social media: Assessing the main performance measurements of user online behavior modification. Journal of Business Research, 129, 271-281. DOI:10.1016/j.jbusres.2021.03.001

[20] Selvaganesan, C., Manisha, P. T., Peter, S. S., Raghunathan, T., & Muralidharan, V. (2025, March). Detecting and Refining Misinformation Using Domain-Adversarial Graph Neural Network: A Hybrid Method. In *2025 International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI)* (pp. 1-7). IEEE. DOI:10.1016/j.dss.2021.113633

[21] Stoff, M. (2025). *Prototypical Visualization: Using Prototypical Networks for Visualizing Large Unstructured Data* (Doctoral dissertation, Technische Universität Wien). https://doi.org/10.34726/hss.2025.119321

[22] Sufi, F. K., & Khalil, I. (2022). Automated disaster monitoring from social media posts using AI-based location intelligence and sentiment analysis. *IEEE Transactions on Computational Social Systems*, *11*(4), 4614-4624. DOI:10.1109/TCSS.2022.3157142

[23] Tian, Y., Sun, X., Yu, H., Li, Y., & Fu, K. (2021). Hierarchical self-adaptation network for multimodal named entity recognition in social media. *Neurocomputing*, *439*, 12-21. DOI:10.1016/j.neucom.2021.01.060

[24] Vaghari, H., Aghdam, M. H., & Emami, H. (2025). HAN: Hierarchical Attention Network for Learning Latent Context-Aware User Preferences With Attribute Awareness. *IEEE Access*. DOI:10.1109/ACCESS.2025.3551402

[25] Vrahatis, A. G., Lazaros, K., & Kotsiantis, S. (2024). Graph attention networks: a

comprehensive review of methods and applications. *Future Internet*, *16*(9), 318. https://doi.org/10.3390/fi16090318

[26] Wang, J., Liu, C., Xu, J., Wang, J., Hao, S., Yi, W., & Zhong, J. (2022). IoT-DeepSense: Behavioral Security Detection of IoT Devices Based on Firmware Virtualization and Deep Learning. *Security and Communication Networks*, *2022*(1), 1443978. DOI:10.58262/ks.v12i1.242

[27] Wang, J., Sun, Z., Yan, H., & Chen, H. (2024, August). MAML-TSC: Model-Agnostic Meta-Learning for Time Series Classification. In *2024 7th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)* (pp. 1043-1049). IEEE. DOI:10.3390/s25134216

[28] Wang, L., Zhang, J., Yang, H., Chen, Z. Y., Tang, J., Zhang, Z., ... & Wen, J. R. (2025). User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems*, *43*(2), 1-37. DOI:10.1145/3708985

[29] Wang, L., Zhang, Y., Yuan, J., Hu, K., & Cao, S. (2022). FEBDNN: fusion embedding-based deep neural network for user retweeting behavior prediction on social networks. *Neural Computing and Applications*, *34*(16), 13219-13235. https://doi.org/10.32604/cmc.2024.057714

[30] Wei, W., Li, X., Zhang, B., Li, L., Damaševičius, R., & Scherer, R. (2023). LSTM-SN: complex text classifying with LSTM fusion social network. *The Journal of Supercomputing*, *79*(9), 9558-9583. DOI:10.1007/s11227-022-05034-w

[31] Wei, X., Xu, G., Wang, H., He, Y., Han, Z., & Wang, W. (2019). Sensing users' emotional intelligence in social networks. *IEEE Transactions on Computational Social Systems*, *7*(1), 103-112. DOI:10.1109/TCSS.2019.2944687

[32] Xia, L., Xu, Y., Huang, C., Dai, P., & Bo, L. (2021, July). Graph meta network for multi-behavior recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval* (pp. 757-766). DOI:10.1145/3404835.3462972

[33] Yue, A., Mao, C., Chen, L., Liu, Z., Zhang, C., & Li, Z. (2022). Detecting changes in perceptions towards smart city on Chinese social media: A text mining and sentiment analysis. *Buildings*, *12*(8), 1182. DOI:10.3390/buildings12081182

[34] Zhang, L., Zhang, W., Wu, L., He, M., & Zhao, H. (2023). SHGCN: Socially enhanced heterogeneous graph convolutional network for multi-behavior prediction. *ACM Transactions on the Web*, *18*(1), 1-27. DOI:10.1145/3617510

Double Deep Q-Network with Experience Replay for Time Dependent Vehicle Routing Problem with Time Windows Under Historical Congestion Constraints

Rina Refianti, Alifurrohman*, Eri Prasetyo Wibowo, Ina Siti Hasanah, Achmad Benny Mutiara
Gunadarma University, Depok, Indonesia
E-mail: alifurrohman@staff.gunadarma.ac.id
*Corresponding author

Keywords: Deep Q-Network (DQN), time dependent vehicle routing problem with time windows (TD-VRPTW), historical congestion, route optimization, logistics distribution, Jakarta

Received: September 26, 2025

This study addresses the Time-Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) for a single-vehicle urban distribution system in Jakarta. Time-dependent travel times are constructed from one week of hourly historical congestion profiles obtained from TomTom Traffic and preprocessed into time-varying speed factors that are mapped to 40 and 50 customer delivery instances with a common service window of 08:00–19:00. A Deep Q-Network (DQN) enhanced with Double DQN and Prioritized Experience Replay (PER) is trained end-to-end using a multilayer perceptron with two hidden layers (128 and 64 units, ReLU activations) to approximate the state–action value function. The reward function penalizes time-dependent travel time, lateness with respect to customer time windows, long inter-customer jumps, and inter-cluster moves, thereby shaping the policy toward both schedule adherence and congestion-aware routing. For each scenario, the agent is trained for 1,000 episodes under three random seeds and evaluated on three representative weekdays (Monday, Wednesday, and Friday). Across all settings, the learned policy achieves a 100% on-time delivery rate with zero late customers, with best time-dependent route costs of approximately 526–539 minutes for 40 customers and 595–617 minutes for 50 customers. Comparative experiments with Genetic Algorithm (GA) and Ant Colony Optimization (ACO) show that ACO attains the shortest travel times, while the proposed DQN+PER model yields routes that are only about 5–8% longer than ACO but reduce time-dependent travel cost by roughly 35–45% compared with GA in the same TD-VRPTW instances. Reward and loss trajectories exhibit smooth convergence, and a sensitivity analysis on the lateness penalty confirms that the main conclusions are robust to hyperparameter variations. These findings demonstrate that leveraging historical congestion to build time-dependent travel times enables DQN-based control to produce competitive, congestion-aware solutions for TD-VRPTW in realistic urban distribution networks.

Povzetek: Ta študija predlaga model DQN, opremljen z Double DQN in Prioritized Experience Replay (PER), za reševanje časovno odvisnega problema usmerjanja vozil s časovnimi okni (TD-VRPTW) v mestni distribuciji na podlagi zgodovinskih podatkov o prometnih zastojih.

1 Introduction

Digital transformation in the urban logistics sector has raised expectations for distribution speed and accuracy, especially in areas with dynamic demand and high congestion levels such as Jakarta. Parcel delivery firms, couriers, and freight logistics now face the complex challenge of designing vehicle routes that are efficient while meeting service time constraints (time windows). Densely populated urban areas, road networks that are not always optimally structured, and daily fluctuations in traffic volume create distinct challenges for efficient route planning. Failure to meet these constraints not only affects customer satisfaction, but also incurs additional costs in the form of fuel consumption, late-delivery penalties, and increased carbon emissions [1] [2] [3]. Several recent industry reports and academic surveys underline that route decisions that are not adaptive to actual traffic conditions and demand can cause logistics inefficiencies of up to 20–30% from optimal potential [4]. Consequently, time-aware (time-dependent) vehicle-routing models are essential to close the gap between planned and realized performance.

In general, the Vehicle Routing Problem (VRP) is a classical combinatorial optimization problem that aims to determine delivery routes for a set of vehicles serving customers from one or more depots. Over time, VRP has evolved into more realistic variants such as the Time-Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW), which explicitly considers variations in travel time due to traffic congestion and imposes time-window constraints for customer service. In TD-VRPTW, travel times between locations are not fixed but depend on the departure time, reflecting fluctuating traffic

conditions, a factor especially relevant in urban logistics or dense road networks [5] [6]. Many heuristics and metaheuristics assume static conditions and fail to account for dynamic changes in traffic or customer demand, which can lead to suboptimal routing decisions due to entrapment in local optima [7] [8].

In this context, Reinforcement Learning (RL) emerges as a promising paradigm to overcome these limitations. Unlike traditional methods, RL enables a learning agent to explore and exploit the environment through experience, thereby producing adaptive policies without requiring explicit rules [9]. One of the most relevant RL approaches for large-scale dynamic routing is the Deep Q-Network (DQN) [10]. The DQN algorithm uses neural networks to estimate Q-values, making vehicle-routing decision making more efficient and stable than classical Q-learning [11]. Across domains such as Atari games [12], robotics [13], and route optimization [14] DQN has been shown to be a stable and efficient reinforcement-learning baseline [15]. Early applications of DQN to VRP, although still limited, have demonstrated competitive performance and mark the beginning of serious exploration of this algorithm in urban logistics [10].

However, most studies are still limited to idealized simulation environments. Few have integrated actual or historical traffic congestion data as an explicit component in DQN decision-making. Therefore, this research aims to investigate and develop a DQN model that integrates traffic congestion data as a key factor in the objective function to optimize logistics distribution routes in Jakarta. The proposed model is expected to enhance distribution efficiency by adaptively responding to urban traffic dynamics.

2 Related work

Research on the Vehicle Routing Problem (VRP) and its derivatives particularly the Time-Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) has long been a major topic in the field of transportation and logistics. Classical heuristic and metaheuristic methods such as Genetic Algorithm (GA), Tabu Search, and Ant Colony Optimization (ACO) are widely used because of their ability to produce near-optimal solutions with relatively short computation times. Recent TD-VRPTW variants have even incorporated road network realism and environmental costs: Liu et al. [16] proposed a TD-VRPTW model with a congestion avoidance approach using Improved ACO (IACACAA), which is able to reduce total distribution costs, fuel consumption, and CO_2 emissions by adjusting departure times to avoid peak hours. Fan et al. [17] introduced Hybrid GA + VNS (HGAVNS-TS), which takes into account fuel and emission consumption through the MEET model for multi-depot scenarios with time-dependent speeds, showing small optimality gaps and fast convergence. Meanwhile, Wang et al. [18] examined TDEVRPTW with path flexibility using Variable Neighborhood Search (VNS) to describe the importance of route selection that is sensitive to time and changing traffic conditions.

On the side of exact methods and road network based approaches, Ben Ticha et al. [5] developed a Branch and Price approach directly on time-dependent road networks, which proved to produce more accurate solutions compared to customer graph based representations and to avoid false infeasibility cases. Gmira et al. [6] applied Tabu Search with a dominant shortest path structure that is able to evaluate travel time efficiently on complex road networks, achieving an optimality gap below 1% with computation times that are much faster than exact methods. For large-scale cases, Cruz-Chávez et al. [19] developed a parallel Grid-Based Genetic Algorithm (GGA) that showed improvements approaching the best solutions on the Gehring and Homberger benchmarks with significant speedup. However, these conventional approaches still have limitations in terms of adaptability to rapidly changing urban traffic dynamics and require manual parameter adjustments whenever scenarios change [20].

The advancement of Reinforcement Learning (RL) offers a more flexible alternative. RL enables a learning agent to explore and exploit the environment to determine an optimal routing policy without requiring explicit rules. The application of Deep Q-Network (DQN), which combines Q-learning with deep neural networks, has proven effective across various domains, including logistics route planning [21] [11]. DQN's advantage lies in its ability to learn effective sequences of environmental transitions, allowing the exploration of large solution spaces and improving cost efficiency in logistics scenarios [2]. Yue et al. [22] extended this concept by proposing a Deep Q-Learning-Based Multi-Objective Evolutionary Algorithm (DQMOEA) for TD-GVRPTW, which combines DQN-based adaptive local search within an evolutionary framework. The results show significant performance improvements on the Hypervolume (HV) and Inverted Generational Distance (IGD) metrics compared to four other strong multi-objective algorithms.

Further research on the use of Double DQN and Prioritized Experience Replay (PER) have proven effective in improving training stability and efficiency across various route-optimization scenarios. The development of Double Q-Learning with PER for traffic-route simulation significantly reduces overestimation bias and speeds up convergence [23]. Zhu et al. [24] employed a Dueling Double DQN with PER for unmanned vehicle path planning, improving training stability and accuracy. Huo [25] applied DQN-PER to multi-objective routing, accelerating convergence and avoiding local optima. Niu et al. [26] showed that DDQN-PER increases navigation efficiency for ships in multi-agent settings.

The application of DQN in vehicle route planning also shows high effectiveness in dealing with congestion. The model can adapt routing decisions based on real-time as well as historical traffic conditions. For example, DQN has been used to optimize autonomous vehicle routes in heavy traffic [27], reduce queues by up to 49% in urban environments [25], and lower delay in QoS-based multipath routing [28]. These results affirm that integrating DQN with congestion information can

significantly improve the efficiency of intelligent transportation.

The application of Deep Q-Networks (DQN) under Time Window constraints has shown high effectiveness in producing adaptive and efficient solutions. Hybrid models such as Greedy-DQN have been proven to significantly reduce the number of vehicles and total travel distance [11], while attention-based encoder decoder approaches accelerate large-scale optimization [10] . These studies indicate that DQN is effective for solving complex routing problems.

Table 1: Comparison of related research

Name	Method	Dataset Used	Key Metrics	Result
Wang et al. (2020) — TDEVRPTW-PF	Improved Variable Neighborhood Search (VNS) with Path Flexibility and Virtual Distribution Center Concept	30 customers, 12 charging stations, and 5 EVs; working hours from 06:00 to 22:00; speeds of 50–60 km/h (varying by time and road segment).	Minimize the total travel distance.	The VNS (maximum 300 iterations) achieved a best total travel distance of 1,512.4 km, with 30 customers served by 5 EVs.
Jie et al. (2022) — TDVRP (soft TW + stochastic)	Hybrid: Sweep + Improved PSO (IPSO), incorporating traffic congestion and weather conditions.	Solomon C101/C102/R101/R102/RC101/RC102 benchmark instances with time-dependent speeds (Donati, 2008); cost and penalty coefficients: 150 per day, 2 per kilometer, 0.5 for early arrivals, and 1.5 for late arrivals.	distance, waiting time, runtime.	Minimal K; distance deviation of ~8.85% vs. the benchmark and better than NNC/FHI/ACO; waiting time is significantly lower; CPU time ≈ 395–406 s.
Fan et al. (2021) — TDMDGVRPTW	HGAVNS_TS (Hybrid GA + VNS) with temporal–spatial distance clustering, SA-based acceptance, and adaptive neighborhoods.	Benchmark MDVRP and MDVRPTW; road speeds are time-dependent (modeled using trigonometric functions); fuel consumption is affected by speed, load, and road gradient.	Objective: Total Cost = fuel + fixed costs + early/late penalties; evaluation based on the optimality gap vs. ALNS/HGSADC/VNS and runtime.	Effective: achieves small optimality gaps and faster convergence than baseline methods on MDVRP/MDVRPTW instances.
Gmira et al., 2020 (EJOR)	Tabu Search with dominant shortest-path strategy, constant-time neighborhood evaluation, tabu list, aspiration criteria, and diversification mechanisms.	NEWLET instances (narrow and wide time windows) for road-network TD-VRPTW, with up to 200 nodes and 580 arcs.	Impr (%) vs greedy; GapO (%) vs optimum; Time(s) TS vs BP	Improvement of 3.04% (narrow) and 5.28% (wide); optimality gap GapO < 1%; average TS runtime of 5.43 s vs. 358.27 s (narrow) and 13.14 s vs. 2454.46 s (wide) compared with BP.

Name	Method	Dataset Used	Key Metrics	Result
Cruz-Chávez et al. (2019) — Grid-Based GA for VRPTW	GGA (grid-based genetic algorithm) built on GA-VRPTW: global selection and k-point crossover, with segment-wise mutations distributed via MPI on a MiniGrid architecture.	Solomon C101–C109 (100 cust.) and Gehring–Homberger C1_10_x (1000 cust.).	RE (relative error) vs optimal/BKS, runtime, speedup, latency.	GGA consistently attains solutions closer to the optimum than GA-VRPTW (Wilcoxon p = 0.003843); runtimes are on the order of several hundred seconds, with speedups approaching or even exceeding linear in several configurations.
Liu et al. (2019) — TD-VRPTW + Congestion Avoidance	IACACAA (Improved Ant Colony Algorithm) combined with OFVTPBN, which optimizes the inter-node travel process by scheduling departure times and allowing vehicles to stop temporarily during congestion.	Solomon C1/R2/RC1/RC2 benchmark instances with 100 customers; depot operating from 07:00 to 23:00; 10-minute time slots; peak-hour congestion from 08:00–09:00 and 18:00–19:00; time-dependent speeds of 20 km/h under congestion and 60 km/h (with variations) under normal conditions.	total distribution cost, driver cost, fuel + emission cost, CO_2 (kg), travel time (min), distance (km), number of vehicles, runtime (s).	Compared with classical ACO (TACA), total cost is reduced by up to ~43% (e.g., R204/R205/R206/R210); fuel/emission costs and CO_2 decrease by ~60%; runtime is ≤ ~403 s; for instance, in RC208 a feasible route with 6 vehicles is obtained, with travel time shorter relative to the travelled distance, indicating effective congestion avoidance.
Yue et al. (2024) — TD-GVRPTW (multi-obj.)	DQMOEA + DQN-based adaptive local search	Solomon-derived instances (25/50/100 customers); 5 time zones; soft time windows.	Hypervolume, Inverted generational distance	Outperforms NSGA-III, MOEA-D, MaOEA-AC, and hpaEA, achieving dominance on 61.3% of the instances in terms of HV and 64.8% in terms of IGD (out of 168 instances).

3 Methodology

3.1 Problem formulation

The problem considered in this study is formulated as a Time-Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) for a single vehicle. The objective is to minimize the total routing cost composed of (i) travel time that depends on the realized departure time, (ii) waiting time (early arrival before the opening of the time window), and (iii) lateness penalties (service start

after the closing time). The objective and constraints are given in Eqs. (1)–(7).

We consider a directed graph $G = (V, A)$with location set $V = (0) \cup V_c$, where node 0is the depot and $V_c = ($are customers; arcs $(i, j) \in A$satisfy $i \neq j$. Each customer $i \in V_c$has a service time window $[a_i, b_i]$and a service duration s_i(minutes). The time-dependent travel time from $i$to $j$when departing at time $t$is denoted $\tau_{ij}(t)$; in all constraints and the objective, $\tau_{ij}(\cdot)$is evaluated at the realized departure time. Routing decisions are encoded by binary variables $x_{ij} \in (0,1)$indicating whether arc $(i, j)$is traversed, along with the arrival time $t_i \geq 0$, waiting time $w_i \geq 0$(if arriving before a_i), and lateness $L_i \geq 0$(if service starts after b_i). We define the service start at node $i$as $t_i^{\text{start}} = t_i + w_i$and the departure time as $d_i = t_i + w_i + s_i$. Waiting and lateness enter the cost with non-negative weights $\alpha_w$and α_L(minute equivalents).

The optimization objective is to minimize the total routing cost, defined as the sum of time-dependent travel time, waiting time, and lateness, each evaluated at the realized departure times.

$$\min cost = \underbrace{\sum_{(i,j)\in A} x_{ij}\tau_{ij}(d_i)}_{time-dependent\ ravel\ time} + \alpha\omega \underbrace{\sum_{i\epsilon V_c} \omega_i}_{waiting} + \alpha L \underbrace{\sum_{i\epsilon V_c} L_i}_{latenes}, \; d_i = t_i + \omega_i + s_i (1)$$

The model is subject to several groups of constraints. First, the visit constraint ensures that each customer is served exactly once:

$$\sum_{j\in V(i)} x_{ij} = 1, \forall i \in V_c, \; \sum_{i\in V(j)} x_{ij} = 1, \forall j \in V_c \; (2)$$

Because we consider a single-vehicle tour, the route starts and ends at the depot.

$$\sum_{j\in V_c} x_{0j} = 1, \sum_{i\in v_c} x_{i0} = 1. \; (3)$$

Temporal consistency is captured by a time-propagation constraint that advances the next arrival by the realized departure plus the time-dependent travel time evaluated at that departure; a sufficiently large constant M(*big-M*) relaxes this constraint when the arc is not selected

$$t_j \geq d_i + \tau_{ij}(d_i) - M\big(1 - x_{ij}\big), \forall (i,j) \in A, i \neq j, \; (4)$$

Time-window compliance requires that service start within the window up to tolerated lateness, with non-negativity for waiting and lateness:

$$t_i^{start} = t_i + \omega_i \geq a_{i,} \; t_i^{start} \leq b_i + L_i, \forall \in V_c, (5)$$

$$\omega_i \geq 0, L_i \geq 0, \forall_i \in V_c \; (6)$$

Finally, the variable domains are stated explicitly.

$$x_{ij} \in \{0,1\}, t_i \geq 0 \, , \forall (i,j) \in A, i \neq j; \; t_i \geq 0 \; \forall i \in V; \; \omega_i \geq 0, L_i \geq 0 \; \forall i \in V_c \; (7)$$

To align the learning signal with the optimization goal in (1), the per-step reward used in training is the negative of the same cost components, each evaluated at the realized departure time $d_{i_t} = t_{i_t} + w_{i_t} + s_{i_t}$.

$$r_t = -\big(\tau i_t j_t\big(d_{i_t}\big) + \alpha_\omega \omega_{j_t} + \alpha_L L_{j_t}\big) - \lambda_{cc} 1\{k(i_t) \neq k(j_t), i_t \neq 0, j_t \neq 0\} \; (8)$$

In our experiments, the penalty weights and reward normalization are fixed. Lateness is weighted by $\omega_L = 400$ to strongly discourage time-window violations, while long jumps between customer groups are penalized through a jump penalty $\alpha_{\text{jump}} = 50$ seconds per kilometer and a cross-cluster penalty $\gamma_{\text{cross}} = 300$ seconds. Within the same cluster, short intra-cluster moves receive a smaller distance penalty $\beta_{\text{inner}} = 30$. The per-step reward is divided by a constant scale $S = 1000$ to keep reward magnitudes in a moderate range and maintain numerical stability during training. These coefficients were chosen based on preliminary sensitivity checks and then kept fixed across all scenarios rather than being extensively tuned as hyperparameters.

3.2 Data collection

In this study, the data collected include:

1. Customer Location Data

   Comprising two scenarios with 40 and 50 customer locations across Jakarta. The dataset contains the full addresses of each customer.

2. Traffic Congestion Data

   Information on congestion-prone locations in Jakarta obtained from news reports and publicly available data on official government website.

3.3 Data preprocessing

Customer addresses and congestion-prone points were first converted into geographic coordinates (latitude and longitude) using the TomTom Geocoding API. We used the TomTom Traffic API to obtain current and free-flow speeds per road segment together with day-of-week and hour-of-day metadata. From a one-week log we constructed hourly congestion factors at the rush-hour windows (e.g., 07:00–10:00 and 15:00–21:00) by taking a robust ratio of free-flow to current speed per segment, while non-rush hours are treated as neutral (factor ≈ 1). Each customer node inherits the hourly factor of its nearest road segment within a small radius. When computing travel for a leg $(i \rightarrow j)$, we evaluate the factor at the realized departure hour and, if the leg crosses an hour boundary, we accumulate the contribution hour by hour. These hourly rush-focused profiles define the time-dependent travel times used throughout the study.

3.4 Model DQN

The proposed model adopts a Deep Q-Network (DQN) to solve the Time-Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) under congestion conditions in Jakarta. DQN is chosen for its ability to learn end-to-end routing policies via reinforcement learning without requiring an explicit traffic dynamics model. The time-dependent travel times $\tau_{ij}(t)$are constructed from

hourly historical congestion and are evaluated at the realized departure time $d_i = t_i + w_i + s_i$.

The Q-network is implemented as a fully connected multilayer perceptron (MLP) that operates on node-wise features. At each decision step, the state is represented as an $N \times 6$ tensor, where N denotes the number of nodes (one depot and all customers), and each node has six hand-crafted features (visit status, current-node indicator, normalized distance to depot and cluster centroid, normalized current time, and residual slack to the closing time window). The network consists of two hidden layers with 128 and 64 units, respectively, each followed by a ReLU activation. The output layer is linear and produces one scalar Q-value per node, corresponding to the estimated return of selecting that node as the next visit. No dropout or batch normalization is applied in this architecture. The network is trained using the Adam optimizer with a learning rate of 1×10^{-4}. Preliminary experiments indicated that this relatively small and shallow architecture, combined with Double DQN and prioritized replay, yields stable training without noticeable overfitting.

Agent environment interaction experiences are stored in a prioritized experience replay (PER) buffer. We employ the proportional variant of PER (Schaul et al., 2016), in which each transition $i$is assigned a non-negative priority $p_i = | \delta_i | + \varepsilon$, where $\delta_i$is its temporal-difference (TD) error and $\varepsilon = 10^{-3}$ avoids zero priority. The sampling probability is defined as

$$P(i) = \frac{p_i^{\alpha}}{\sum_j p_j^{\alpha}} \quad (9)$$

where the exponent $\alpha \in [0,1]$controls the strength of prioritization. We set $\alpha = 0.6$, which provides a compromise between uniform replay ($\alpha = 0$) and fully greedy replay ($\alpha = 1$), so that transitions with larger TD errors are sampled more frequently while still maintaining sufficient diversity in the replayed experiences. To correct for the bias introduced by non-uniform sampling, we apply importance-sampling weights $w_i \propto (N \cdot P(i))^{-\beta}$, where $N$is the current buffer size. The exponent $\beta$is annealed linearly from $\beta_{\text{start}} = 0.4$at the beginning of training to $\beta_{\text{end}} = 1.0$at the end, so that the updates become increasingly unbiased as learning progresses. During training, the agent selects actions using an ε-greedy strategy, where $\varepsilon$is decayed from 1.0 to 0.05 over episodes to gradually shift from exploration to exploitation.

The reward function is designed to balance route efficiency and punctuality and is aligned with the TD-VRPTW objective described in Section 3.1. At each decision step, the reward is defined as the negative of the time-dependent travel, waiting, and service times, augmented with penalty terms for service lateness, long jumps between distant customers, and inefficient cross-cluster moves, and normalized by a constant reward scale to keep the magnitude of rewards in a numerically stable range (see Eq. (8)). The associated penalty weights and normalization factor are fixed across all experiments, based on preliminary sensitivity checks, so that different scenarios remain comparable.

3.5 Training strategy and experiment setup

The model training process was carried out in Python using the TensorFlow library. The training strategy was designed to maximize the ability of the Deep Q-Network (DQN) to learn efficient routing policies. Each training scenario was run for 1,000 episodes, which based on preliminary tests was sufficient to achieve convergence without overfitting. The main hyperparameter values used include a learning rate of 1×10^{-4} for the Adam optimizer, a discount factor $\gamma = 0.95$, a mini-batch size of 64, a replay buffer capacity of 50,000 transitions, a target-network update frequency of every 1,000 training steps, and at least 1,000 warm-up steps before training starts. Exploration followed an ε-greedy strategy, starting from $\varepsilon = 1.0$and decreasing to 0.05 using a multiplicative decay factor of 0.997 to balance exploration and exploitation during training. These hyperparameters were selected by combining common practice in DQN-based control with small-scale preliminary experiments on the Jakarta dataset: higher learning rates (e.g., 10^{-3}) produced unstable Q-value updates, while smaller values slowed convergence; $\gamma = 0.95$provided a reasonable trade-off between short-term routing costs and long-term episode returns; and a mini-batch size of 64 offered a compromise between noisy gradients for very small batches and increased memory usage for larger batches. Once this configuration was found to yield stable learning and reasonable convergence speed, it was kept fixed across all experimental scenarios rather than being extensively tuned through large-scale hyperparameter searches.

To ensure result stability, each scenario combination was run with three different random seeds (42, 77, 99). In addition, tests were conducted at two customer scales 40 and 50 customers to evaluate the effect of distribution density on route performance.

3.6 Evaluation metrics

The success of the Deep Q-Network (DQN) on the Time Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) is evaluated using five key metrics: Total Route Cost (total route travel time), On-time Delivery Ratio measuring delivery punctuality within the 08:00–19:00 window, Average Lateness assessing mean delay per customer, Distribution of Congestion Status indicating the model's ability to avoid congestion, and Route Visualization on an OpenStreetMap (OSM) map to verify visit order and traffic conditions.

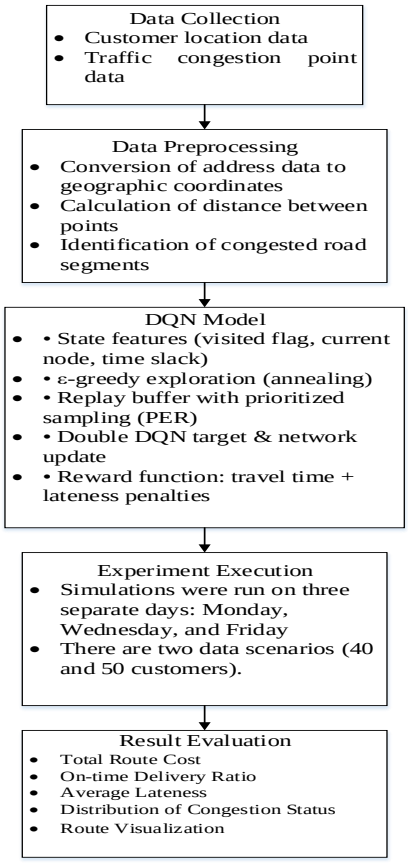


Figure 1: Research flowchart

4 Experiment and discussion

4.1 Dataset description and experimental scenarios

This study evaluates the performance of a Deep Q-Network (DQN) equipped with Double DQN and Prioritized Experience Replay (PER) on the Time Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) using two Jakarta delivery instances with 40 and 50 customers, respectively. Each instance consists of a single depot and multiple customer locations served within a common time window of 08:00–19:00 WIB. Historical congestion data were obtained from previous data collection using the TomTom Traffic API, which provides actual traffic speed and free-flow speed; the ratio of the two is used as an hourly congestion factor. Experiments are conducted for three weekdays: Monday, Wednesday, and Friday. In addition, each customer-size setting is tested with three random seeds (42, 77, 99) to ensure training stability and minimize initialization bias. The analysis focuses on best cost (minimum total travel time), on-time delivery percentage, average lateness, and the distribution of congestion status (successfully avoided, congested, and neutral).

4.2 Reward and loss analysis

This subsection presents an analysis of the reward and loss curves as an overview of the DQN learning process for the 40- and 50-customer cases. The graphs display the evolution of per-episode reward and average per-episode loss over 1,000 training episodes on three test days Monday, Wednesday, and Friday using the best seed, to assess the model's stability and convergence.

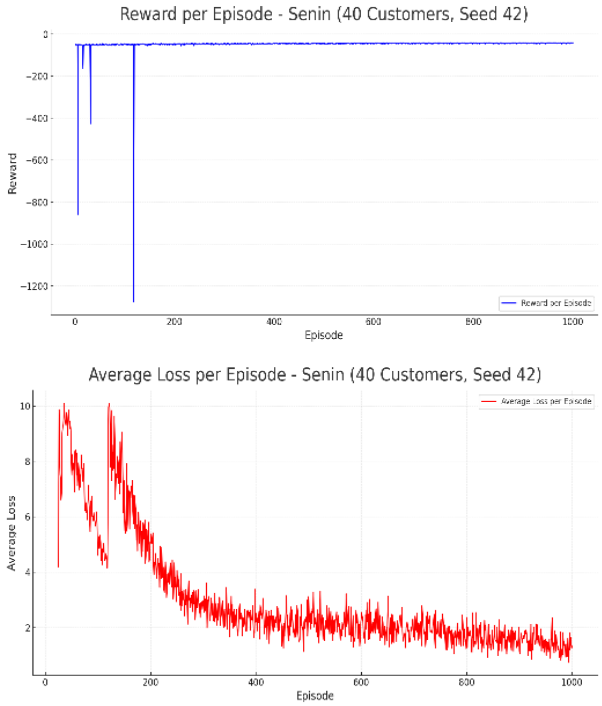


Figure 2: Training reward and loss curves for the 40-customer case (monday)

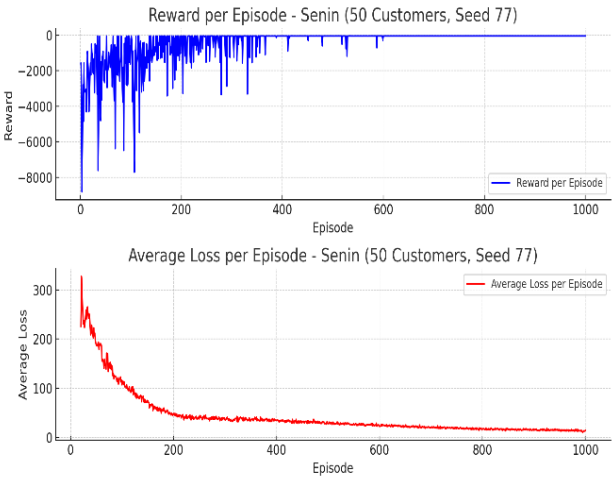


Figure 3: Training reward and loss curves for the 50 customer case (monday)

Figure 2. The per-episode reward and loss curves for the 40 customer case show a stable convergence pattern. At the beginning of training, the per-episode reward exhibits sharp negative fluctuations, indicating intensive exploration. After around 200 episodes, the per-episode reward rises and stabilizes within a narrow band around zero, indicating that the agent has reached a relatively stable policy. The loss pattern also shows a consistent downward trend: the initially high average loss gradually decreases and flattens out near episode 1,000. This combination of steadily improving reward and declining

loss indicates that the successfully learned an efficient routing policy that satisfies the time-window constraints for the 40 customer scenario.

Figure 3**.** In the 50 customer scenario, the training pattern poses greater challenges yet still trends toward good convergence. The per-episode reward is initially highly volatile and far below zero, with sharp drops approaching −8000, indicating increased complexity due to the larger number of delivery points. In the 50 customer case, the per-episode reward gradually improves after roughly 300 episodes and then stabilizes near zero, which suggests that the learning process converges despite the increased problem complexity. The per-episode loss falls steeply from a very high initial value (around 300) and stabilizes near zero in the later episodes. The consistent loss reduction together with the reward increase confirms the model's ability to discover an optimal policy, although the larger customer set requires a longer training process before stability is achieved.

4.3 Best route analysis and trip summary

This subsection presents the analysis of the best routes produced by the Deep Q-Network (DQN) model for each combination of day and customer count. The experimental results are consolidated in Table 2 to provide a comprehensive overview of the model's performance.

Table 2: Summary of results

Day	Customers	Seed	Best Cost (min)	On-time (%)	n Late	Congestion Hit	Congestion Avoided	Neutral Segments
Monday	40	42	529.4	100	0	10	5	26
Monday	40	77	529.4	100	0	5	10	26
Monday	40	99	529.4	100	0	5	10	26
Wednesday	40	42	526.6	100	0	5	7	29
Wednesday	40	77	526.6	100	0	5	7	29
Wednesday	40	99	526.6	100	0	5	7	29
Friday	40	42	538.7	100	0	2	10	29
Friday	40	77	538.7	100	0	2	10	29
Friday	40	99	538.7	100	0	2	10	29
Monday	50	42	617	100	0	8	9	34
Monday	50	77	616.9	100	0	10	8	33
Monday	50	99	606.9	100	0	9	9	33
Wednesday	50	42	611.4	100	0	8	7	36
Wednesday	50	77	595.5	100	0	5	9	37
Wednesday	50	99	602	100	0	9	6	36
Friday	50	42	616.4	100	0	7	8	36
Friday	50	77	614.5	100	0	2	12	37
Friday	50	99	619.3	100	0	7	8	36

Based on Table 2, in the 40 customer scenario the DQN model exhibits very consistent behaviour across the three random seeds for each day. The best route cost is identical for all seeds on a given day (529.4 minutes on Monday, 526.6 minutes on Wednesday, and 538.7 minutes on Friday), and every run achieves 100% on-time delivery with zero late customers. The congestion statistics also remain in a narrow range: on Monday each route encounters between 5 and 10 congested segments and successfully avoids 5–10 segments, on Wednesday 5 congested and 7 avoided segments are observed for all seeds, and on Friday the model typically hits only 2 congested segments while avoiding 10. These results indicate that the learned policy is highly stable in the 40 customer case, producing almost identical costs and schedule adherence while showing a consistent pattern in how congestion hotspots are managed.

In the 50 customer scenario, the DQN model again maintains 100% on-time delivery for all seeds and all days, confirming that the time-window constraints remain fully satisfied even as the problem size increases. The best route cost shows only modest variation between seeds, with values in the range of 606.9–617.0 minutes on Monday, 595.5–611.4 minutes on Wednesday, and 614.5–619.3 minutes on Friday. The number of congestion hits remains within approximately 2–10 segments per route, while the number of avoided segments is consistently high (6–12 segments), indicating that the policy is able to adapt its visiting order to more complex traffic conditions without sacrificing punctuality. Overall, the per-seed trip summaries confirm that the DQN model yields robust and reliable routes across different initialisations, days, and customer scales.

4.4 Sensitivity analysis

A complementary sensitivity analysis was also carried out on the lateness penalty parameter β_late in the reward function, rather than on the Prioritized Experience Replay (PER) hyperparameters. This test was performed on the most challenging scenario (50 customers on Friday) by varying $\beta_late \in (300, 400, 500)$ while keeping all other

Double DQN and PER settings fixed. The goal is to examine how sensitive the learned policy and its routing cost are to reasonable changes in the lateness penalty and to verify that the configuration used in the main experiments is sufficiently representative. Table 3 summarizes the mean best-route cost and greedy cost, together with their standard deviations, for each configuration across three random seeds. The results show that differences in β_late produce only minor variations in the total travel time, while the on-time rate remains 100% with no late customers. These findings indicate that the model's performance is relatively stable with respect to moderate changes in β_late, suggesting that the baseline configuration used in the main experiments is sufficiently representative.

Table 3: Sensitivity analysis

Konfigurasi β	Average Best Cost (minutes)	Std Best Cost (minutes)	Average Greedy Cost (minutes)	Std Greedy Cost (minutes)	Mean Number of Late Customers
300	621.57	2.58	650.06	0.00	0
400	616.74	2.46	650.06	0.00	0
500	614.74	1.54	641.61	7.32	0

The parameter β_late serves as the penalty weight for customer service lateness in the reward function. To examine how sensitive the model is to changes in this penalty, we conducted experiments on the 50 customer Friday scenario with three β_{late} values, namely 300, 400, and 500, while keeping all other hyperparameters fixed (learning rate, γ, batch size, network configuration, and exploration–exploitation scheme). For each β_{late} value, the model was run with three random seeds (42, 77, 99), and the table reports the best route (best seed).

Overall, increasing β_late tends to reduce the total travel cost. With β_late = 300, the best cost is about 621.57 minutes. When β_{late} is increased to 400, the best cost decreases to ≈616.74 minutes, and for β_{late} = 500 it further decreases to ≈614.74 minutes. This reduction is only a few minutes (around 0.9% when comparing β_{late} = 300 and 500), indicating that the Double DQN+PER performance is relatively stable with respect to the penalty weight within the tested range. The standard deviation across seeds is also small (around 1.54–2.58 minutes), showing that this trend is consistent and not driven by a single lucky seed.

Importantly, the on-time performance remains 100% for all β_late values. No customers are served late and the average lateness is zero. This implies that, in this scenario, all tested β_late configurations are already large enough to ensure that the model keeps every service within the time window. Consequently, differences in β_late mainly affect the route selection pattern and its interaction with congestion, rather than determining whether customers are late or not.

Overall, these results show that the Double DQN+PER is reasonably robust to variations in β_late within the range 300–500. Increasing β_late from 400 to 500 provides a small improvement in cost and congestion avoidance, but the margin is modest and does not change the main conclusions of the study. Therefore, β_late = 400 can still be used as the baseline configuration in the main experiments, while the results for β_{late} = 300 and 500 are reported as a parameter sensitivity analysis that confirms the model's stability with respect to the lateness penalty weight.

4.5 Performance comparison

This subsection presents a comparative analysis of the performance of the proposed Double Deep Q-Network with Prioritized Experience Replay (Double DQN+PER) against heuristic baselines, namely the Genetic Algorithm (GA) and Ant Colony Optimization (ACO), across different customer sizes and operating days (Monday, Wednesday, and Friday). The comparison focuses on several key indicators: the best route cost (in minutes), the on-time delivery ratio, the number of late customers, and the distribution of road segments that are affected by congestion, successfully avoided, or classified as neutral. Table 4 reports these measurements and provides a quantitative view of the relative advantages of each method in maintaining routing efficiency while satisfying time-window constraints under historical congestion.

Table 4: Performance comparison

Method	Day	Customers	Best Cost (min)	n Late	On-time (%)	Congestion Hit	Congestion Avoided	Neutral Segments
Double DQN+PER	Monday	40	529.4	0	100	10	5	26
Double DQN+PER	Wednesday	40	526.6	0	100	5	7	29
Double DQN+PER	Friday	40	538.7	0	100	2	10	29

Method	Day	Customers	Best Cost (min)	n Late	On-time (%)	Congestion Hit	Congestion Avoided	Neutral Segments
Double DQN+PER	Monday	50	606	0	100	9	9	33
Double DQN+PER	Wednesday	50	595.5	0	100	5	9	37
Double DQN+PER	Friday	50	614.5	0	100	2	12	37
GA	Monday	40	881.83	11	95	5	14	32
GA	Wednesday	40	965.32	11	72.50	4	12	25
GA	Friday	40	922.78	7	82.50	10	3	28
GA	Monday	50	1048.61	18	64	4	14	33
GA	Wednesday	50	1074	18	64	7	9	35
GA	Friday	50	1151.89	21	58	7	10	34
ACO	Monday	40	493.46	0	100	13	2	26
ACO	Wednesday	40	503.48	0	100	9	2	30
ACO	Friday	40	503.09	0	100	9	3	29
ACO	Monday	50	554.19	0	100	15	2	34
ACO	Wednesday	50	584.37	0	100	8	6	37
ACO	Friday	50	556.73	0	100	9	5	37

The comparison results show that ACO consistently yields the lowest route cost (total travel time) across all day scenarios (Monday, Wednesday, Friday) and customer sizes (40 and 50). For 40 customers, ACO produces a total travel time of approximately 493–503 minutes, while for 50 customers it lies in the range of 554–584 minutes. Double DQN+PER ranks second, with slightly longer travel times than ACO, namely around 526–539 minutes for 40 customers and 595–617 minutes for 50 customers, but it remains far more efficient than GA, which yields very high route costs (around 882–965 minutes for 40 customers and 1,049–1,152 minutes for 50 customers). In terms of service level, both Double DQN+PER and ACO maintain schedule performance very well, as there are no late customers (n Late = 0) and the on-time rate reaches 100% in all scenarios, whereas GA fails to satisfy the time-window constraints, with a large number of late customers and a drop in on-time performance to as low as 58% in the worst case (50 customers, Friday).

When viewed from the perspective of interaction with congestion, ACO tends to focus on minimizing total travel time even though the number of segments affected by congestion (Congestion Hit) is relatively higher. In contrast, Double DQN+PER exhibits a more adaptive pattern with respect to congestion, with fewer hits and a substantial number of segments that pass-through hotspots outside peak hours, making it more controlled in dealing with heavy traffic without sacrificing punctuality. GA often appears to "avoid congestion" (with relatively high Congestion Avoided values), but this strategy results in much longer routes, higher total costs, and many violations of time windows. Overall, ACO can be positioned as the classical heuristic that is most efficient in terms of travel time for this instance, while Double DQN+PER serves as a learning-based method that offers an attractive compromise between schedule adherence, congestion management, and flexibility for further deployment in more dynamic and complex distribution scenarios.

4.6 Route and congestion visualization

To provide a clearer picture of the performance of the optimal routes produced by the algorithm, we present visualizations of the distribution paths on a map of Jakarta. The visualization displays the visit sequence of each customer, the depot, and the congestion conditions encountered. Each path is color-coded: red denotes *Congestion Hit* (segments where congestion was encountered), whereas green denotes *Non-hit* segments, covering both *Congestion Avoided* (congested areas successfully bypassed) and *Neutral Segments* (normal traffic). This mapping is consistent with the tabular results and helps readers link numerical metrics such as total travel time and on-time delivery percentage to the actual spatial conditions.

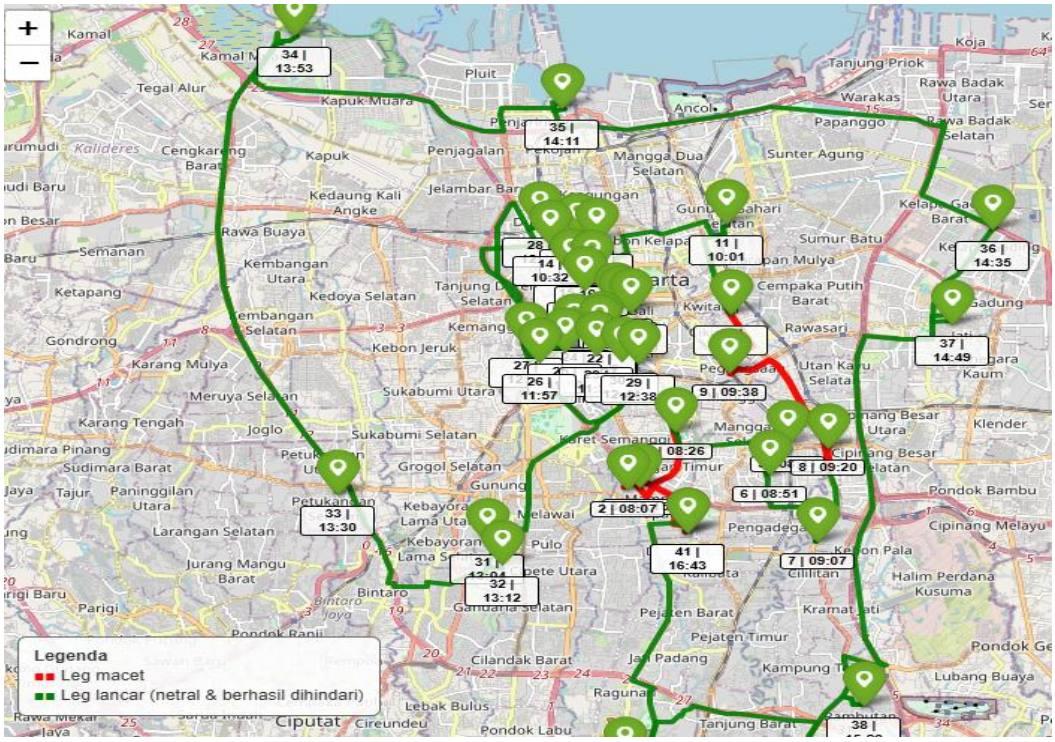


Figure 4: Visualization Results for the 40 Customer Scenario (Monday)



Figure 5: Visualization Results for the 50 Customer Scenario (Monday)

4.7 Discussion

To evaluate the performance of the DQN method enhanced with Double DQN and Prioritized Experience Replay (PER), the model was trained and tested on a Time Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) using Jakarta's historical traffic congestion data. The experiments covered three service days (Monday, Wednesday, and Friday) and two customer scenarios (40 and 50 customers), each run with three different random seeds (42, 77, 99). The best-route summaries show that, for 40 customers, the total time-dependent travel cost lies in the range of approximately 526–539 minutes, while for 50 customers it lies around 595–617 minutes, with 100% on-time delivery and zero late customers in all cases. The per-seed results reported in Table 2 indicate that these outcomes are highly robust with respect to random initialization: for each day and customer size, the three seeds produce almost identical best costs and identical on-time ratios, with only small variations in congestion hits and avoided segments.

The reward and loss curves also exhibit a pattern consistent with reinforcement learning convergence, where episode returns increase while the average training loss decreases and eventually stabilizes. This behavior supports the interpretation that Double DQN effectively reduces overestimation bias in Q-values, leading to more stable learning, while Prioritized Experience Replay accelerates convergence by focusing updates on more informative transitions [24] [25]. In other words, the architectural choices adopted in this study are not only

theoretically motivated but also empirically translated into stable and reproducible routing performance on a time-dependent, congestion-aware TD-VRPTW instance.

The comparison with heuristic methods such as Genetic Algorithm (GA) and Ant Colony Optimization (ACO) provides a clearer picture of the relative position of the proposed DQN approach. As shown in Table 4, ACO consistently achieves the lowest total travel time across all days and customer scenarios. The DQN+PER method ranks second in terms of travel cost, with slightly higher route times than ACO, yet still significantly outperforms GA, which produces much longer routes and a large number of late customers. From a service-level perspective, DQN+PER and ACO both maintain a 100% on-time delivery rate, whereas GA exhibits many time-window violations. When congestion interaction is examined in more detail, DQN+PER shows adaptive behavior: the number of Congestion Hit segments remains at a moderate level, while a substantial number of segments pass through congested areas outside peak hours. This is consistent with previous findings that DQN variants with PER can respond to changing traffic conditions and yield more efficient routes than traditional heuristic methods [26]. The adaptive advantage is also aligned with the Time-Dependent VRP/VRPTW literature, which concludes that modeling travel times based on historical congestion profiles can improve schedule adherence and distribution efficiency [29].

The sensitivity analysis of the lateness penalty parameter β_{late} shows that the main conclusions of this study do not depend on a single hyperparameter setting. Varying β_{late} within a reasonable range leads only to small changes in the best-route cost and does not alter the fact that all customers are served on time. This indicates that the penalty value used is already strong enough to enforce time-window feasibility, while its influence is mainly on fine-grained route selection and congestion-avoidance strategies. Nevertheless, this study has several limitations, including reliance on one week of historical congestion data as a proxy for time-dependent travel times and a focus on a single vehicle and a single depot. Future research could extend the framework to multi-vehicle and multi-depot settings, integrate real-time congestion information and long-term congestion patterns, and conduct broader comparisons with other deep reinforcement learning architectures, thereby providing a more comprehensive view of congestion-aware urban distribution routing in the TD-VRPTW context.

5 Conclusion

This study presented a Deep Q-Network (DQN) model enhanced with Double DQN and Prioritized Experience Replay (PER) for solving the Time-Dependent Vehicle Routing Problem with Time Windows (TD-VRPTW) using historical congestion data from Jakarta. The proposed approach explicitly models time-dependent travel times derived from one week of congestion observations and integrates them into a single-vehicle routing policy for urban distribution. Across three service days (Monday, Wednesday, Friday) and two customer scenarios (40 and 50 customers), each evaluated over three random seeds, the DQN+PER model consistently achieved a 100% on-time delivery rate with zero late customers. The per-seed analysis further showed that the best-route costs and congestion statistics are highly stable across seeds, indicating that the learned policy is robust to stochastic variations in initialization and training.

Comparative experiments against Genetic Algorithm (GA) and Ant Colony Optimization (ACO) highlighted the relative strengths of the proposed method. ACO obtained the lowest total travel time in all scenarios, while the DQN+PER model produced slightly longer routes but still substantially reduced travel time compared with GA, which generated much longer routes and many time-window violations. Both DQN+PER and ACO maintained perfect schedule adherence, whereas GA's on-time performance dropped to as low as 58% in the most congested cases. These results suggest that the proposed DQN+PER model offers a competitive trade-off between travel-time efficiency, strict time-window satisfaction, and congestion-aware routing in a realistic TD-VRPTW setting. Sensitivity analysis on the lateness penalty also indicated that the main conclusions are not dependent on a single hyperparameter choice. Future work will extend this framework to multi-vehicle and multi-depot settings, incorporate real-time traffic and longer-term congestion patterns, and explore additional deep reinforcement learning architectures, thereby providing a more comprehensive foundation for congestion-aware urban distribution planning.

Nevertheless, this study has several limitations, including reliance on historical congestion data as a proxy for real-time conditions and a focus on a single vehicle type. For future research, the model can be extended to incorporate real-time traffic data, multi-vehicle fleets, and the integration of weather and dynamic demand data to better approximate real-world operational conditions.

Acknowledgement

We would like to express our heartfelt gratitude to Universitas Gunadarma for providing the necessary support and resources throughout this research. Our sincere appreciation also goes to the Ministry of Education, Culture, Research, and Technology of Indonesia under the Fundamental Research-Regular, Contact Number: 0419/C3/DT.05.00/2025, for its financial assistance, which has been instrumental in the successful completion of this project. The collaboration and encouragement received from both institutions have greatly contributed to the advancement of our work. Thank you for believing in our vision and enabling us to pursue our research goals.

References

[1] B. Lin, B. Ghaddar, and J. J. I. T. o. I. T. S. Nathwani, "Deep reinforcement learning for the electric vehicle routing problem with time windows," vol. 23, no. 8, pp. 11528-11538, 2021. 10.1109/TITS.2021.3105232

[2] X. Zhang, Y. Yang, J. Cai, Q. Zhu, W. Chen, and Q. Lin, "Deep Reinforcement Learning-Based Multi-Agent Algorithm for Vehicle Routing Problem in Complex Logistics Scenarios," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1-8: IEEE. 10.1109/IJCNN60899.2024.10650335

[3] W. Pan and S. Q. J. A. I. Liu, "Deep reinforcement learning for the dynamic and uncertain vehicle routing problem," vol. 53, no. 1, pp. 405-422, 2023. https://doi.org/10.1007/s10489-022-03456-w

[4] F. Guo, Q. Wei, M. Wang, Z. Guo, S. W. J. T. r. p. E. l. Wallace, and t. review, "Deep attention models with dimension-reduction and gate mechanisms for solving practical time-dependent vehicle routing problems," vol. 173, p. 103095, 2023. https://doi.org/10.1016/j.tre.2023.103095

[5] H. Ben Ticha, N. Absi, D. Feillet, A. Quilliot, and T. Van Woensel, "The time-dependent vehicle routing problem with time windows and road-network information," in *Operations Research Forum*, 2021, vol. 2, no. 1, p. 4: Springer. https://doi.org/10.1007/s43069-020-00049-6

[6] M. Gmira, M. Gendreau, A. Lodi, and J.-Y. J. E. J. o. O. R. Potvin, "Tabu search for the time-dependent vehicle routing problem with time windows on a road network," vol. 288, no. 1, pp. 129-140,2021. https://doi.org/10.1016/j.ejor.2020.05.041

[7] M. Ammouriova, E. M. Herrera, M. Neroni, A. A. Juan, and J. J. A. S. Faulin, "Solving vehicle routing problems under uncertainty and in dynamic scenarios: From simheuristics to agile optimization," vol. 13, no. 1, p. 101, 2022. https://doi.org/10.3390/app13010101

[8] Y. Yunita, D. Stiawan, and D. P. Rini, "Vehicle Routing Problem with Time Windows using Hybrid Metaheuristic Dragonfly Algorithm and Variable Neighborhood Search: Work on Progress," in *2024 11th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, 2024, pp. 521-525: IEEE. 10.1109/EECSI63442.2024.10776058

[9] S. Ara *et al.*, "Vehicle Routing Problem Solving Using Reinforcement Learning," in *2023 26th International Conference on Computer and Information Technology (ICCIT)*, 2023, pp. 1-6: IEEE. 10.1109/ICCIT60459.2023.10441644

[10] A. Gupta, S. Ghosh, and A. Dhara, "Deep reinforcement learning algorithm for fast solutions to vehicle routing problem with time-windows," in *Proceedings of the 5th Joint International Conference on Data Science & Management of Data (9th ACM IKDD CODS and 27th COMAD)*, 2022, pp. 236-240. https://doi.org/10.1145/3493700.349372

[11] S. Kosolsombat and C. Ratanavilisagul, "Applied Deep Reinforcement Learning for Solving the Vehicle Routing Problem with Time Windows," in *2023 8th International Conference on Computational Intelligence and Applications (ICCIA)*, 2023, pp. 21-25: IEEE. 10.1109/ICCIA59741.2023.00012

[12] F. Moreno-Vera, "Performing deep recurrent double q-learning for atari games," in *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, 2019, pp. 1-4: IEEE. DOI: 10.1109/LA-CCI47412.2019.9036763

[13] J. Escobar-Naranjo, G. Caiza, P. Ayala, E. Jordan, C. A. Garcia, and M. V. J. A. S. Garcia, "Autonomous navigation of robots: optimization with DQN," vol. 13, no. 12, p. 7202, 2023. https://doi.org/10.3390/app13127202

[14] Y. J. H. i. S. E. Yang and Technology, "Path planning under high-dimensional input states based on deep q-network," vol. 120, pp. 576-585, 2024. https://doi.org/10.54097/r6fs0580

[15] D. Seng, J. Zhang, X. J. K. T. o. I. Shi, and I. Systems, "Visual Analysis of Deep Q-network," vol. 15, no. 3, 2021. 10.3837/tiis.2021.03.003

[16] C. Liu, G. Kou, X. Zhou, Y. Peng, H. Sheng, and F. E. J. K.-B. S. Alsaadi, "Time-dependent vehicle routing problem with time windows of city logistics with a congestion avoidance approach," vol. 188, p. 104813, 2020. https://doi.org/10.1016/j.knosys.2019.06.021

[17] H. Fan, Y. Zhang, P. Tian, Y. Lv, H. J. C. Fan, and O. Research, "Time-dependent multi-depot green vehicle routing problem with time windows considering temporal-spatial distance," vol. 129, p. 105211, 2021. https://doi.org/10.1016/j.cor.2021.105211

[18] L. Wang, S. Gao, K. Wang, T. Li, L. Li, and Z. J. J. o. A. T. Chen, "Time-Dependent Electric Vehicle Routing Problem with Time Windows and Path Flexibility," vol. 2020, no. 1, p. 3030197, 2020. https://doi.org/10.1155/2020/3030197

[19] M. A. Cruz-Chávez, A. Rodríguez-León, R. Rivera-López, and M. H. Cruz-Rosales, "A Grid-Based Genetic Approach to Solving the Vehicle Routing Problem with Time Windows," vol. 9, no. 18, p. 3656, 2019. https://doi.org/10.3390/app9183656

[20] G. Chen, J. Gao, and D. J. E. Chen, "Research on Vehicle Routing Problem with Time Windows Based on Improved Genetic Algorithm and Ant Colony Algorithm," vol. 14, no. 4, 2025. https://doi.org/10.3390/electronics14040647

[21] J. Cai, X. Zhang, Q. Lin, L. Dong, W. Chen, and Z. Ming, "Deep Reinforcement Learning for Solving the Vehicle Routing Problem in Practical Logistics," in *2024 IEEE Congress on Evolutionary Computation (CEC)*, 2024, pp. 1-8: IEEE. 10.1109/CEC60901.2024.10612190

[22] B. Yue, J. Ma, J. Shi, and J. J. I. a. Yang, "A deep reinforcement learning-based adaptive search for solving time-dependent green vehicle routing problem," vol. 12, pp. 33400-33419, 2024. 10.1109/ACCESS.2024.3369474

[23] M. Patil, P. Tambolkar, and S. J. I. I. T. S. Midlam-Mohler, "Optimizing Traffic Routes With Enhanced Double Q-Learning," vol. 19, no. 1, p. e70002, 2025. https://doi.org/10.1049/itr2.70002

[24] Z. Zhu, C. Hu, C. Zhu, Y. Zhu, Y. J. J. o. M. S. Sheng, and Engineering, "An improved dueling deep double-q network based on prioritized experience replay for path planning of unmanned surface vehicles," vol. 9, no. 11, p. 1267, 2021. https://doi.org/10.3390/jmse9111267

[25] Q. Huo, "Multi-objective vehicle path planning based on DQN," in *International Conference on Cloud Computing, Performance Computing, and Deep Learning (CCPCDL 2022)*, 2022, vol. 12287, pp. 351-357: SPIE. https://doi.org/10.1117/12.2640707

[26] Y. Niu, F. Zhu, and P. Zhai, "An autonomous decision-making algorithm for ship collision avoidance based on DDQN with prioritized experience replay," in *2023 7th international conference on transportation information and safety (ICTIS)*, 2023, pp. 1174-1180: IEEE. 10.1109/ICTIS60134.2023.10243882

[27] S. Moon, S. Koo, Y. Lim, and H. J. A. S. Joo, "Routing control optimization for autonomous vehicles in mixed traffic flow based on deep reinforcement learning," vol. 14, no. 5, p. 2214, 2024. https://doi.org/10.3390/app14052214

[28] L. P. A. Sanchez, Y. Shen, M. J. J. o. N. Guo, and C. Applications, "Mdq: A qos-congestion aware deep reinforcement learning approach for multi-path routing in sdn," vol. 235, p. 104082, 2025. https://doi.org/10.1016/j.jnca.2024.104082

[29] T. Carić, J. J. P.-T. Fosin, and Transportation, "Using congestion zones for solving the time dependent vehicle routing problem," vol. 32, no. 1, pp. 25-38, 2020. https://doi.org/10.7307/ptt.v32i1.3296

DBAS-SLSTM: A Multi-Sensor Fusion Approach for Real-Time Motion Attitude Error Detection and Correction

Zhongyi Ni
Department of Police Physical Training and Tactics, Hubei University of Police, Wuhan, Hubei, 430034, China
Corresponding author's E-mail: nizy-zhongyi@outlook.com

Keywords: motion attitude, posture error detection, multi-sensor data, physical exercise training, joint angle analysis, dynamic beetle antennae search-driven stacked long short-term memory (DBAS-SLSTM)

Received: September 28, 2025

Efficient and accurate posture assessment is essential in physical exercise training to enhance performance, prevent injuries, and support overall well-being. This research aims to develop a Motion Attitude Error Detection System capable of continuously monitoring and correcting human movement using multi-sensor data. The proposed system integrates inertial measurement units, accelerometers, gyroscopes, and vision-based key-point tracking to capture comprehensive motion information. Experiments were conducted using the Motion Attitude Error Detection Dataset, consisting of 4,800 labeled motion sequences representing correct and incorrect postures. Raw sensor signals were preprocessed using a Kalman filter for noise reduction and Z-score normalization for scale consistency. Feature extraction using Wavelet Transform (WT) was then performed to compute joint angles, limb orientations, stability indices, and angular displacement metrics from fixed-length temporal windows. The proposed framework combining rule-based constraints with a Dynamic Beetle Antennae Search–optimized Stacked Long Short-Term Memory (DBAS-SLSTM) model was employed to learn spatiotemporal motion patterns, detect deviations from reference postures, classify error severity, and predict corrective movements. The system also provides real-time visual and auditory feedback to guide users during training. Experimental validation in a MATLAB environment, using a 70%-30% training–testing split, achieved an accuracy of 98.12%, along with high precision, recall, and F1-score values, demonstrating statistically reliable performance. The results confirm that the proposed approach offers a scalable, accurate, and practical solution for posture monitoring in sports training, rehabilitation, and ergonomic applications.

Povzetek: Raziskava razvija sistem, ki z uporabo senzorjev in računalniškega vida samodejno spremlja držo pri gibanju, zazna napake in v realnem času pomaga pri popravkih.

1 Introduction

The attitude of human motion that refers to the direction and the movement of the body in 3D space is one of the most important parameters in such areas as sports science and rehabilitation, as well as human-computer interaction [1]. Exact measurement of motion is critical in the objective evaluation of physical exercise performance, finding the wrong form and injury prevention. This field has been revolutionized by the introduction of miniaturized and inexpensive inertial measurement units (IMUs) that are usually multi-sensor systems of accelerometers, gyroscopes, and magnetometers [2]. These sensors are able to provide a rich stream of data that is able to estimate the kinematic parameters like joint angles, angular velocity, and linear acceleration. When it comes to physical exercise training, the technology allows the transition from the subjective, coach-based visual analysis to the objective, quantitative movement pattern analysis [3,4]. The main difficulty, though, is to strongly integrate such multi-sensor data to come up with a true and drift-free estimate of the actual attitude of the body, and based on which the error can be observed. Figure 1 shows diverse applications for monitoring, training, ergonomics, and safety.

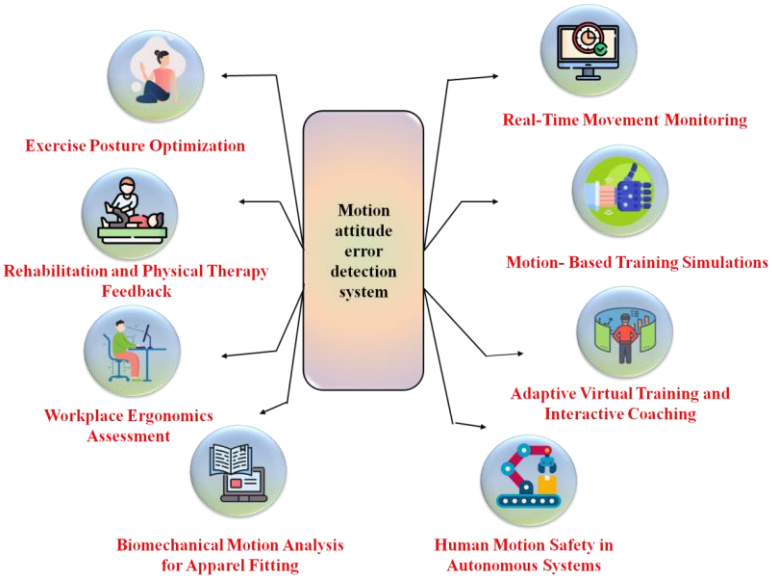


Figure 1: Applications of the motion attitude error detection system

The main idea behind this process is that the accurate recording of the kinematic information is revolutionized by the introduction of inertial measurement units (IMUs) [5, 6]. They are multi-dimensional and can be used to measure human movement in real-time, without being confined to laboratory conditions, as these sensors usually include accelerometers, gyroscopes, and magnetometers to give a rich stream of data [7]. The combination of this multisensor information is required to counteract the basic drift and noise of any single sensor to form a solid basis of motion tracking. It is an error detection which consists of the following physical exercise: inconsistency or deviation of anticipated sensor output and real sensor output [8]. With the assistance of filtering techniques and sensor fusion algorithms, the system can identify anomalies and eliminate noise as well as improve the overall accuracy of measurements [9]. It not only improves the functionality of the system but also provides stability in real-time. It can be extremely useful in a wide variety of contemporary technological uses by enhancing accuracy, safety, and efficiency through the use of multi-sensor data [10].

The accuracy and reliability of attitude and heading measurements in dynamic environments were improved using a Micro-Electro-Mechanical Systems (MEMS) multi-sensor fusion-based Attitude and Heading Reference System (AHRS) [11]. Results confirm improved error correction, numerical stability, and precise attitude positioning, demonstrating feasibility in practical engineering applications. The accuracy of a low-cost attitude observation sensor for the Wave Glider (WG)was examined by developing a multi-sensor hierarchical data fusion algorithm using an improved covariance matching adaptive Unscented Kalman Filter (UKF) [12]. Results demonstrate superior adaptability, stability, and precision in attitude estimation. The design of a multi-sensor data fusion (MSDF) system of motion trajectory monitoring among athletes was approached by [13]. The approach includes the combination of various sensors, system architecture design, and the use of high-level data fusion algorithms. Findings show enhanced accuracy, real-time updates of 20Hz, and flexibility with different sports. Possible limitations were the possible cost of the sensors, the complexity of the system, and the inability to reach extreme motion conditions.

The objective of attaining the proper 3-D human motion capture with the help of wearable inertial sensors in the absence of magnetic interference was suggested by [14]. The approach combines 9-axis sensor data through an extended Kalman filter, heuristic drift mitigation, and position estimation using the foot-mounted zero-velocity-update (ZUPT) method, and was summed up into a biomechanical model. The findings indicate enhanced motion tracking compared to the traditional methodologies. The drawbacks were the dependency on sensor location and the possibility of errors with dynamic movements. To improve human arm motion tracking with an Inertial Measurement Unit (IMU) against environmental changes was suggested by [15]. The algorithm utilizes an Elman Neural Network (ENN) to optimize refined Euler angles data that were pre-processed and calibrate the acceleration inputs without a magnetometer. Findings indicate a mean error of about 35mm and 37.2% x-axis betterment compared to standard procedures. The weaknesses were relying on standardized movements and might not perform well in high-dynamic or irregular movements. To develop a human motion capture system based on MEMS sensors and Zigbee wireless networks was approached by [16]. The process includes the implementation of several sensor nodes on the body, complementary filter data fusion, and processing data to calculate the trajectory and identify motions. Findings reveal an increase in efficiency by 10% and an increase in accuracy by 15%, whereas the limitations related to the location of the sensors and the possible

interference of the network. Table 1 summarizes the existing motion analysis approaches, highlighting the objectives, key outcomes, performance metrics, and inherent technical limitations.

Table 1: Related work of motion analysis and human activity in error detection

Reference	Method	Aim	Outcome	Limitations
Brunson et al. [17]	Generic Multisensor Integration Strategy (GMIS) + Variance Component Estimation (VCE) multi-IMU integration	Improve navigation and positioning accuracy	Positioning 14–16%, roll/pitch 30%, heading 40%	Requires multiple calibrated IMUs and complex integration
Zhong et al. [18]	Multilocal linearization with quaternion generation	Improve in-motion initial alignment	Higher accuracy and robustness than Optimization-Based Alignment (OBA)	Performance depends on measurement quality
Hajjej et al. [19]	IMU + quaternion filtering + LSTM	Human motion detection for healthcare e-learning	85.18% accuracy on wearable datasets	Sensitive to data quality and real-world motion variability
Dai et al. [20]	Multi-Scale Dependency (MSD) neural network for IMU anomaly detection	Detect IMU anomalies with interpretability	F1-score 95% with explainability	Computationally intensive
Li et al. [21]	Wireless smart glove with multi-IMU fusion	Hand motion tracking for rehabilitation	Accurate 3D tracking and joint repeatability	Limited to hand/finger motions
Gu et al. [22]	Magnetic-Inertial Measurement Units (MIMUs) real-time data fusion	Reduce magnetic disturbance errors	RMSE: knee 1.23°, hip 1.15°, elbow 3.67°	Susceptible to extreme magnetic noise
Liu et al. [23]	Unscented Kalman Filter (UKF)-based multi-sensor fusion for Unmanned Surface Vehicle (USV)	Enhance navigation under uncertainties	Improved accuracy and reliability	Application-specific to marine environments
Naseer et al. [24]	Random Forest Long Short-Term Memory (RFL)	Exercise pose estimation in therapy	High classification accuracy	Model complexity and sensor dependency
Chen and Fan, [25]	Convolutional Neural Networks (CNN) and CNN-Long Short-Term Memory (CNN-LSTM)	Exercise recognition and health prediction	mAP 78.6%, PCK@0.5 91.5%	Vision-based; sensitive to occlusion
Rao et al. [26]	Stereo vision + Bidirectional Convolutional Gated Recurrent Unit (Bi-CGRU)	Automated squat posture analysis	96.1% classification accuracy	Requires camera setup and lighting control
Charia et al. [27]	MediaPipe + Bidirectional Gated Recurrent Unit (Bi-GRU) with attention	Squat-type recognition and correction	94% accuracy	Limited generalization beyond squat actions

Traditional methods of detecting motion attitude errors are grounded in Machine Learning (ML) and Deep Learning (DL). Classifying and finding the patterns of errors with ML algorithms, such as the Support Vector Machine (SVM) and the random forest (RF), in comparison with auto-discovering features with DL algorithms, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are more suitable in more dynamic environments.

Motion attitude error detection limitations include high requirements in large labeled datasets, which are, in many cases, hard to acquire. ML models are not always good at nonlinear relationships, whereas DL methods consume large amounts of computational resources.

Moreover, the two approaches are vulnerable to processing problems in real-time, sense noise, and the ability to generalize well to different dynamic conditions.

1.1 Research questions

1. How effectively can the proposed DBAS-SLSTM framework detect deviations from reference human postures using multi-sensor IMU data in real time?
2. To what extent does DBAS–based parameter optimization improve classification accuracy and robustness compared to non-optimized SLSTM models?
3. How accurately can the proposed system classify posture error severity and predict corrective movements under varying sensor noise and user motion conditions?

The aim was to develop a robust motion attitude error detection system using multi-sensor data, integrating rule-based logic with a DBAS-SLSTM model to accurately detect deviations, classify error severity, and predict corrective movements for enhancing precision, reliability, and real-time performance in human motion monitoring. The following shows the key contribution of the research.

Data Collection: Collected various physical exercises, including squats, push-ups, lunges, and jumping jacks, covering diverse exercises, enabling accurate posture analysis and real-time motion monitoring.

Data Preprocessing: Applied a Kalman filter to remove sensor noise and Z-score normalization to standardize measurements, ensuring clean and consistent input signals.

Feature Extraction: Used Wavelet Transform to extract joint angles, limb orientations, and stability metrics, capturing both time-frequency motion characteristics effectively.

Proposed Method: Developed a DBAS-SLSTM hybrid framework combining rule-based logic and deep learning for real-time posture error detection and corrective feedback.

1.2 Research gaps

Multi-sensor fusion-based AHRS based on an extended Kalman Filter used by MEMS can still experience lower performance in very dynamic or vibration-intensive conditions, restricting the ability to work in extreme conditions [11]. Although the Wave Glider multi-sensor hierarchical data fusion technique is more accurate with low-cost sensors, it can be scaled down to large-scale or high-noise GPS signals [12]. Also, the USV navigation system based on UKF may be vulnerable to sudden environmental disruptions such as strong ocean currents, and this may interfere with the real-time accuracy [23].To address these issues, a two-tiered computational framework incorporating rule-based logic, in addition to a DBAS-SLSTM model, is used to identify posture deviations, categorize the extent of the error, and predict correctional movements, which improves accuracy, flexibility, and real-time operation under dynamic circumstances.

2 Methodology

Data collection involved acquiring multi-sensor motion data collected from participants performing various physical exercises from the Motion Posture Monitoring Dataset. Data preprocessing applied a Kalman filter to remove noise and Z-score normalization to standardize measurements. Feature extraction was performed using the Wavelet Transform to compute joint angles, limb orientations, and stability metrics. The proposed method employs a hybrid framework combining rule-based logic and a DBAS-SLSTM model for real-time motion error detection, severity classification, and corrective feedback. Figure 2 illustrates the overall flow of methodologies.

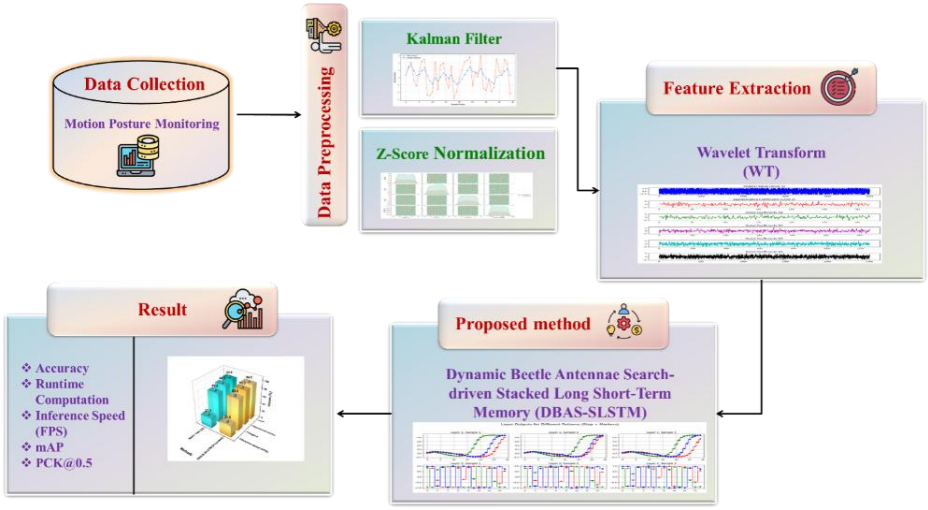


Figure 2: Overall flow in motion error detection in physical exercise

2.1 Data collection

The Motion Attitude Error Detection Dataset (https://www.kaggle.com/datasets/ziya07/motion-attitude-error-detection-dataset/data/data) is composed of 5,000 motion sequences, which are obtained at the time of physical exercise activities, and each motion sequence is a sequence of 3 seconds long, recorded with a fixed temporal window length. Each sample has flattened multi-sensor characteristics based on the measurements of inertia, such as joint angles and accelerator signals. The dataset is classified into two categories, namely, a correct posture (2,520 samples) and a posture error (2,480 samples), and it is balanced. The data were randomly split with 70% training and 30% testing data, and no overlapping subjects across each split. This open network facilitates reproducibility and valid comparison between posture error-detecting models.

2.2 Data preprocessing using kalman filter

The Kalman filter combines multi-sensor measurements, dampens noise, and predicts correct motion states to enhance the Motion Attitudes Error Detection System's functionality to track, assess, and correct human movements in real time effectively. State space, which is made up of state equations and observation equations, describes the Kalman filter. The following describes the equations that illustrate the system's dynamic properties in equation (1).

$$W_{l+1} = BW_l + B\mu_l + \omega_l$$
$$Y_l = GW_l + w_{l_1} \quad (1)$$

Where Y_l is the measuring vector, G is the observational matrix, u_l is a random noise vector,W_{l+1} is the next time step state vector, W_l denoted by the system's state, B denoted by the state transitions matrix, A is the matrix of control variables, μ_l is a controlling factor, and ω_l is the noise or disturbance in the process. Kalman filtering across the state of the most recent measurement and the present measurement yields the best estimate of the state. ω_l and ω_l have covariances of R_l and Q_l, respectively. Two steps make up the Kalman filter's operation: the prediction step and the correction step. The two steps in the computation procedure are as follows in equations (2) and (3).

$$\widehat{W}_{l+1|l} = \mathrm{B}\widehat{W}_{l|l} + A\mu_l$$
$$O_{l+1|l} = \mathrm{BO}_{l|l}B^S + R_l \quad (2)$$
$$O_{l+1|l+1} = (\mathrm{J} - L_{l+1}G)O_{l+1|l}$$
$$L_{l+1} = O_{l+1}|lG^S(GO_{l+1|l}G^S + Q_{l+1})^{-1}$$
$$\widehat{W}_{l+1|l+1} = \widehat{W}_{l+1|l} + L_{l+1}(Y_{l+1} - G\widehat{W}_{l+1|l}) \quad (3)$$

Where the predicted system state vector at time step$l$ is denoted by $\widehat{W}_{l+1|l}$, the estimated the system state vector at the time step $l+1$ by $\widehat{W}_{l|l}$, $A\mu_l$ represents control matrix, $G^S$the predicted system state vector at the time step l by $\widehat{W}_{l+1|l}$, the estimated uncertainty matrix of the current state is denoted by $O_{l|l}$,$R_l$is the process noise covariance matrix value, the predicted estimate covariant matrix for the next state by $O_{l+1|l}$, and the Kalman filter gain at time step $l+1$ by L_{l+1} , J is theIdentity matrix, $\widehat{W}_{l+1|l+1}$ – Updated state estimate after incorporating measurement Y_{l+1}, and$(Y_{l+1} - G\widehat{W}_{l+1|l})$ – The measurement residual is the discrepancy between the actual and expected measurements. Another way to characterize the covariance matrix is as follows in equations (4) and (5).

$$f_{l+1} = W_{l+1} - \widehat{W}_{l+1|l+1}$$
$$O_{l+1|l+1} = F(f_{l+1}f_l^S + 1) \quad (4)$$
$$= F((W_{l+1} - \widehat{W}_{l+1|l+1})$$
$$\times (W_{l+1} - \widehat{W}_{l+1|l+1})^S) \quad (5)$$

The estimation error vector f_{l+1} represents the difference between the actual and estimated state at time $l+1$. The posterior covariance $O_{l+1|l+1}$ quantifies uncertainty after correction, computed using the function F and the transpose operation for f_l^S outer product calculations. By minimizing the covariance matrix, an estimating process known as the Kalman filter determines the optimal estimate of the state. The Kalman filter obtains a posterior using measured values and prior knowledge, refreshes it with measurements, and uses the prior result as the current value. Figure 3 displays the smoothed versus raw acceleration signals for participant motion tracking accuracy.

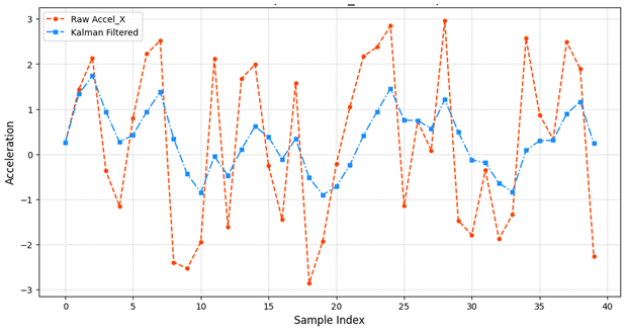


Figure 3: Kalman filtered for motion attitude error detection system

2.2.1 Z-score Normalization

Z-score normalization to standardize multi-sensor measurements, removing scale differences and reducing bias. This ensures consistent input for the motion attitude error detection system, enhancing accuracy in monitoring, evaluating, and correcting human movement in real time. Z-score normalization was a statistical standardization that ensured each feature was represented equally. It standardizes the data distributions to a mean of 0 and a standard deviation of 1, while still preserving relative relationships between features and structural transformations evenly. Equation (6) provides a numerical illustration of Z-score normalization.

$$Z = \frac{X-\mu}{\sigma} \quad (6)$$

Where μ represents the attribute mean, σ represents the feature standard deviation, Z represents the normalized value, and X represents the initial data value. The denominator $(X-\mu)$ in the equation above will position the data such that its mean is 0. It was subsequently scaled

to unit variance as 1 by multiplying $by\ \sigma$. Figure 4 shows the pairwise feature relationships colored by posture correctness after normalization.

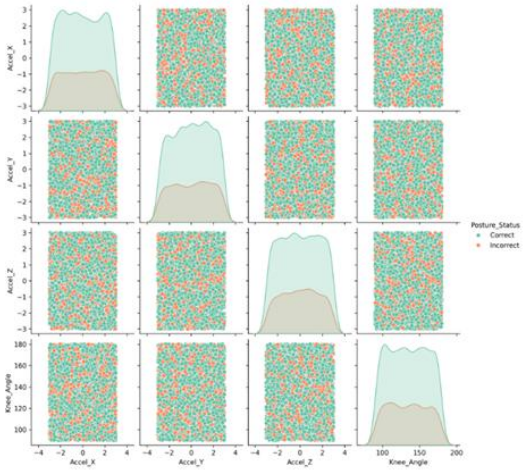


Figure 4: Normalized Multi-Sensor Distributions for Motion Attitude Error Detection System

2.3 Feature extraction using wavelet transform (WT)

Wavelet Transform (WT) enables the Motion Attitude Error Detection System to decompose multi-sensor signals into time–frequency components, capturing both transient and continuous movement patterns, enhancing accuracy in detecting, evaluating, and correcting human motion errors in real time. In comparison to traditional Fourier Transform (FT), which only offers global frequency content, or Short-Time Fourier Transform (STFT), which is limited to a particular fixed window, WT can behave simultaneously as a high-frequency transient and a low-frequency trend detector. Similarly, time-domain statistical characteristics that are simple might fail to detect fine dynamic changes in human movement. Therefore, WT is chosen to overcome these constraints yielding more detailed, multi-scale descriptions of motion signals to enhance the performance of the models. The continuous WT and the discrete WT are two types of wavelet transforms. The Continuous WT is translation invariant and permits wavelet transforms at all scales with continuous translation. In pattern-matching applications like discontinuity or chirp signal detection, the Continuous WT is frequently utilized. The Maximum detection section will provide further information about the CWT. The Discrete WT uses positions and scales that depend on a power of two. It is adequate for precise reconstruction, non-redundant, and more efficient. Consequently, the Discrete WT finds extensive application in feature extraction and data compression. Furthermore, vibration and wave propagation methods for damage assessment can be enhanced by incorporating wavelet transforms, enabling more accurate analysis of structural health.

A mass spectrum $t(p)$is broken down by the Discrete WT into an approximation component and multiple detailed components in equations (7) – (9).

$$t(p) = B_I(p) + \sum_{i<I} C_i(p), B_I(p) = \sum_{l\in Y} d_{I,l}\varphi_{I,l}(p) \quad (7)$$

$$C_i(p) = \sum_{l\in Y} c_{i,l}\psi_{i,l}(p), d_{i,l} = \langle t(p), \varphi_{i,l}(p)\rangle \quad (8)$$

$$c_{i,l} = \langle t(p), \psi_{i,l}(p)\rangle, i, l \in Y \quad (9)$$

Where B_I is the approximate element at the level$I$, C_i is the comprehensive component at level$i$, and I is the overall level of the DWT transform. The wavelet coefficient of B_I, orC_i, is the inner product of the scaling function $d_{I,l}$ at level I with translation l and the spectrum $t(p)$. Likewise, the wavelet coefficients of$c_{i,l}$,$\varphi_{I,l}$ is the inner product of the wavelet functional $\psi_{i,l}$at levels i with translations l and the spectrum$t(p)$. The WT$\psi_{i,l}$ and scaled function ($\psi_{i,l}$) expands by a factor of two as the level i increases;$l \in Y$ – Translation index; shifts the scaling function along the signal for analysis across time, $\sum_{l\in Y}$ the wavelet shifts to the right as l increases. Mallat's technique with filter banks requires P(M)operations to estimate the Discrete WT coefficient of a length M spectrum. The $c_{i,l}$ represents the detail component at the level i and translation l, extracted using the wavelet function $\psi_{i,l}(p)$, which captures high-frequency or transient features. Similarly, the inner product $\langle t(p), \varphi_{i,l}(p)\rangle$ computes the approximation coefficient $d_{i,l}$, while $\langle t(p), \psi_{i,l}(p)\rangle$ calculates the detail coefficient, quantifying signal features across scales and positions. Figure 5 shows that the multi-level signal decomposition reveals frequency components for movement analysis.

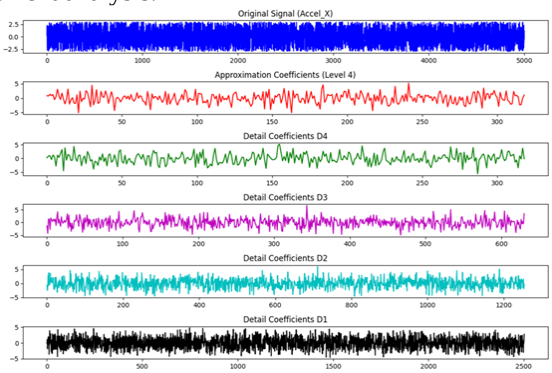


Figure 5: Wavelet-based feature extraction for motion attitude error detection system

2.4 DBAS-SLSTM to detect deviations from reference postures, classify error

severity, and predict correct movements

The DBAS-SLSTM model is proposed for a Motion Attitude Error Detection System utilizing multi-sensor data to monitor, evaluate, and correct human movements in real time. The system integrates accelerometers, gyroscopes, and magnetometers to capture comprehensive motion patterns. The Stacked LSTM element can learn temporal relationships and non-linearity of human motion and hence accurately capture anomalies of the right postures.

Adaptively control the proposed framework of DBAS-SLSTM, which bears high conceptual resemblances to other adaptive fuzzy control and neural adaptive control strategies applied in nonlinear and uncertain dynamic systems. The model, like adaptive controllers, continuously changes its internal parameters, responding to real-time sensor uncertainties and posture deviations, and allows for robust error detection and correction. The parameter tuning mechanism implemented based on DBAS is analogous to the adaptive gain adjustment mechanism, as it provides stability, rapid convergence, and resistance to sensor noise in a closed-loop feedback system. The DBAS algorithm is adaptively tuned to adjust network parameters and maximize convergence to achieve robustness in the conditions of noisy sensors. The model enables timely posture error detection, the degree of its severity, and corrective instructions. The experimental outcomes indicate a higher level of performance in terms of latency to reduce motion error detection and enhance accuracy in prediction, and thus, the system can be applied in sports training, rehabilitation, and ergonomic assessment, where continuous observation is crucial and quick feedback is essential. Pseudocode 1 shows the procedure of DBAS-SLSTM.

Pseudocode 1: DBAS-SLSTM

Inputs:
- *Multi − sensor data streams*:
  Accelerometer $A[t]$, *Gyroscope* $G[t]$, *Magnetometer* $M[t]$
- *Hyperparameters*:
  Sequence length $= 50$ *timesteps*
  Learning rate $= 0.001$
  Number of SLSTM layers $= 2$
  Hidden units per layer $= 64, 32$
  Dropout rate $= 0.4$
  Activation functions: *ReLU* (*Dense*), *Softmax* (*Output*)
  DBAS step size $\delta = 0.5$
  Decay factor $\alpha = 0.9$
  Number of exploration directions $P = 4$
  Total iterations $S = 100$
  Optimizer $=$ *Adam*

Outputs:
- *Detected posture errors*
- *Severity score* $(0 - 1)$
- *Corrective instructions*

Step 1: *Preprocessing*
1.1 *Normalize sensor inputs* $A[t], G[t], M[t]$ *to range* $[-1,1]$.
1.2 *Construct input sequence* $J[t]$ $= [A[t], G[t], M[t]]$ *of length* 50.
1.3 *Apply wavelet decomposition to remove high* $-$ *frequency noise.*

Step 2: *Forward Pass through SLSTM*

For each timestep s in sequence $(1 \ldots 50)$:
Forget Gate
$e_s = \sigma(X_e * [g_(s-1), w_s] + a_e)$
Example: $e_ = \sigma(0.8 * [0.3, 0.5] + 0.2)\ 0.62$
Input Gate
$j_s = \sigma(X_j * [g_(s-1), w_s] + a_j)$
Example: $j_s = \sigma(0.5 * [0.3, 0.5] + 0.1) 0.58$
Candidate Cell State
$\tilde{D}_s = tanh(X_D * [g_(s-1), w_s] + a_d)$
Example: $\tilde{D}_s\ tanh(0.7 * [0.3, 0.5] + 0.1)\ 0.45$
Cell State Update
$D_s = e_s * D_(s-1) + j_s * \tilde{D}_s$
Example: $D_s = 0.62 * 0.4 + 0.58 * 0.45 \approx 0.49$
Output Gate
$p_s = \sigma(X_p * [g_(s-1), w_s] + a_p)$
Example: $p_s = \sigma(0.6 * [0.3, 0.5] + 0.2) \approx 0.66$
Hidden State
$g_s = p_s * tanh(D_s)$
Example: $g_s = 0.66 * tanh(0.49) \approx 0.31$

Step 2.5: *Training Loop*
- *Number of epochs* $= 50$
- *Batch size* $= 32$ **
- *Loss function* $=$ *Categorical Cross − Entropy* $+$ *MSE for severity*
- *Track training/validation loss and accuracy*
- *Generate learning curves for convergence visualization*

Return sequence of hidden states g_s*to Dense Layers.*

Step 3: *Dense Layers for Posture Error Prediction*
3.1 *Fully Connected Layer* (*ReLU*, 128 *units*).
3.2 *Dropout Layer* ($rate = 0.4$).
3.3 *Fully Connected Layer* (*Softmax*, 2 *units*: *Correct vs Error*).
3.4 *Output*:
- $y_pred \in \{0,1\}$ (*posture classification*)
- *Error severity score* $\in [0,1]$.

Step 4: *Dynamic Beetle Antennae Search* (*DBAS*) *Optimization*
Initialize:
$\delta = 0.5, \alpha = 0.9, iteration\ s = 1$
$W_best = random\ weight\ initialization$
While $s \leq S$ (100 *iterations*):
 # Step Update
$\delta_(n+1) = \delta * \alpha\text{^}s$
Example: $\delta_(2) = 0.5 * 0.9\text{^}2 = 0.405$
Multi − Directional Exploration
Generate $P = 4$ *random exploration vectors* $\vec{c}$
$w_left = W_best + \delta_(n+1) * \vec{c}$
$w_right = W_best - \delta_(n+1) * \vec{c}$
Evaluate Fitness (*Loss Function*)
$f_left = Loss(SLSTM(w_left))$
$f_right = Loss(SLSTM(w_right))$
Update Best
If $f_left < f_right$:
$W_best = w_left$
Else:
$W_best = w_right$
Lens Opposition − Based Learning (*LOBL*)
$midpoint = (a_i + b_i)/2$
$W'_best = midpoint + (midpoint/-W_best/m)$
If $Loss(W'_best) < Loss(W_best)$:
$W_best = W'_best$
Increment iteration $s = s + 1$
Return optimized parameters W_best.

Step 5: *Posture Correction Feedback*
If y_pred == "*Error*":
Display corrective instruction (*e.g.*, "*Straighten your back*")
Show severity score (*e.g.*, $0.73 =$ *moderately severe*)

End Algorithm with Model Summary Table
- *Include a table listing each layer, units, activation, dropout, and output*

– Include learning curves showing training /validation loss and accuracy over epochs
End Algorithm

Figure 6 shows how the multi-sensor data can be sequentially processed to detect posture deviation, classify the severity of posture deviation, and provide corrective feedback in real-time.

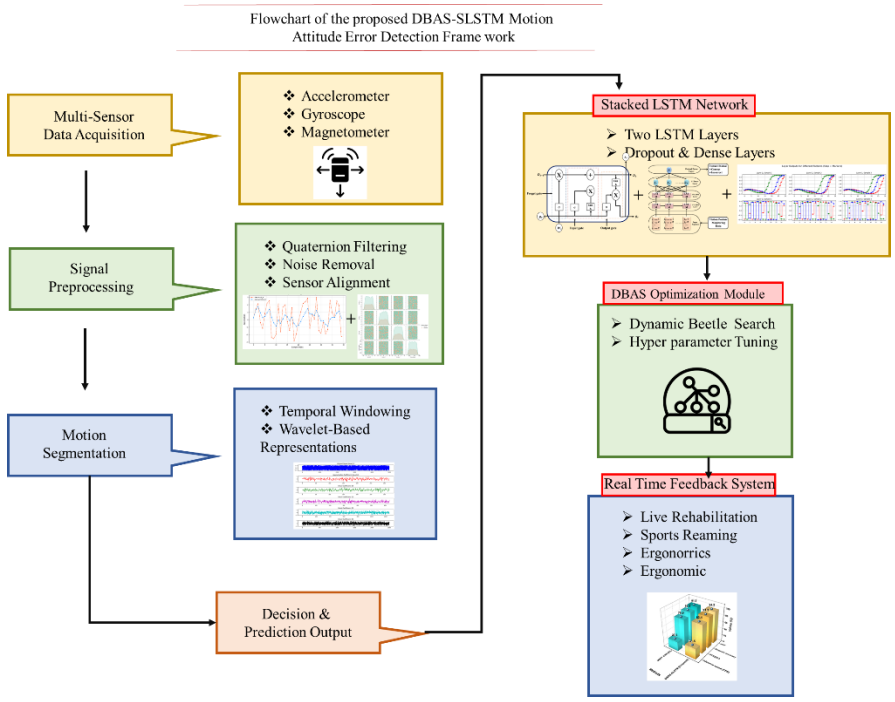


Figure 6: Architecture of the DBAS-SLSTM-based motion attitude error detection framework

2.4.1 Long short-term memory to identify the sequence that deviates from the correct pattern

LSTM networks are employed in the Motion Attitude Error Detection System to capture temporal dependencies in multi-sensor data, enabling accurate real-time monitoring, evaluation, and correction of human movements. By retaining relevant motion history, LSTM enhances posture recognition, detects deviations, and predicts corrective actions efficiently. Memory cells are the fundamental units of LSTM design. Despite being merely neurons in neural networks, these cells may learn and ignore irrelevant portions of past states, store pertinent new information in the cell, modify cell states selectively, and ultimately regulate the information transmitted. Through a sequence of LSTM units and temporal delays, the LSTM architecture's physical sense for runoff modeling enables it to preserve the sequential character of flow dependence to various rainfall stations. It allows it to forget in each memory cell structure and logs the relationship between various rainfall stations and runoff. Recently, a variety of LSTM-based models have been used for forecasting and runoff modeling. The following are the LSTM memory cell transition equations. Figure 7 shows the architecture of an LSTM.

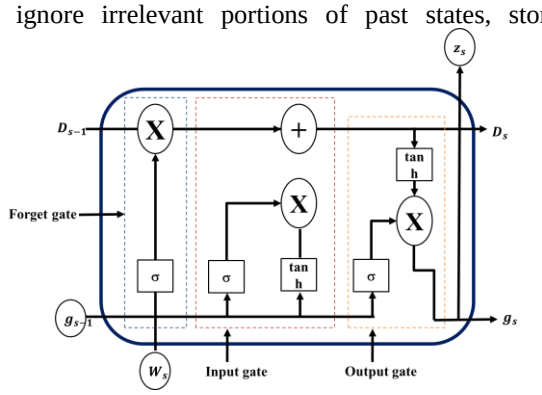


Figure 7: LSTM cell architecture for motion attitude error detection system

The first gate, referred to as the forget gate, determines which element of D_{s-1} must be forgotten by multiplying it by e_s, which ranges from 0 to 1. This is how the LSTM memory cell processes different information. Furthermore, w_s represents the cell's current input, g_{s-1} is the state of the preceding cell in the layer,σ denotes the sigmoid activation function, and X_e and a_e are trainable parameters in the forget gate in equation (10).

$$e_s = \sigma(X_e \cdot [g_{s-1}, w_s] + a_e) \quad (10)$$

The input gate, which comes next, is where the value that has to be updated is determined in equation (11).

$$j_s = \sigma(X_j \cdot [g_{s-1}, w_s] + a_j) \quad (11)$$

Where j_s value of the output variable ranges from 0 to 1. The input gate has two trainable parameters: X_j and a_j. Next, using the following equation, the possible vectors werethe current input (w_s), $tanh$ is the hyperbolic tangent function, and the final hidden state, g_{s-1}, in equation (12).

$$\widetilde{D}_s = \tanh\left(X_D \cdot \left[g_{s-1,w_s}\right] + a_d\right) \quad (12)$$

X_D and a_d are input gate parameters that may be taught throughout the training process, while $\widetilde{D}_s$ is a vector having values between 0 and 1.

The following formula is then used to determine D_s in equation (13).

$$D_s = e_s * D_{s-1} + j_s \times \widetilde{D}_s \quad (13)$$

In the output gate, the output is finally computed by using a sigmoid activation function in equation (14).

$$p_s = \sigma\left(X_p\left[g_{s-1,w_s}\right] + a_p\right) \quad (14)$$

Where the values of the vector p_s range from 0 to 1. $D_s = e_s * D_{s-1} + j_s \times \widetilde{D}_s$ updates the cell state by combining forgotten past and new input. $p_s = \sigma\left(X_p\left[g_{s-1,w_s}\right] + a_p\right)$ computes the output gate, and $g_s = p_s \times \tanh(D_s)$ determines the hidden state for the next time step. a_p and X_p are parameters that can be trained, p_s– Output gate controlling which parts of the cell state are exposed. g_s – Hidden state, the final output for the time step. The output is multiplied element-wise by the hyperbolic tangential angle of D_s to determine the new hidden state, g_s, as follows in equation (15).

$$g_s = p_s \times \tanh(D_s) \quad (15)$$

Stacked Long Short-Term Memory multiple layers to understand complex movement patterns

The SLSTM architecture is employed in the Motion Attitude Error Detection System to process sequential multi-sensor data. By capturing temporal dependencies and complex motion patterns, it enables real-time monitoring, evaluation, and correction of human movements, improving posture accuracy, preventing injuries, and enhancing overall exercise performance. Two LSTM layers combined with two completely connected dense layers for identifying the sequential structure of the problem, and a layer for dropout with a dropout rate of 0.4 on the initial fully connected layer for preventing overfitting, make up the suggested deep neural network architecture. Additionally, the C_1 *and* C_2 normalization method was used to lower the probability of the model overfitting byD_m maintaining small weight and bias values. To capture the effect of autocorrelation, input variables are converted to sequences due to the autocorrelated nature of stream flow.J_{s-m} represents a sequential input of multi-sensor measurements over time, J_s capturing past motion states for SLSTM processing. The last dense layer was the same for all basins, with one neuron signifying runoff output. The wavelet was utilized in both basins to identify independent variables, and the influence of convergence between variables in outflow modeling was also examined using basic normalized datasets. Figure 8 shows the suggested sequential machine learning in conjunction with a dense layer that is fully linked.

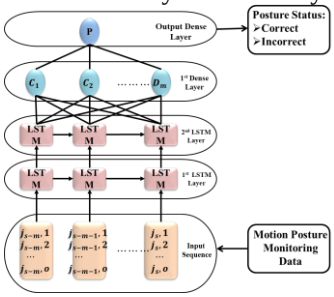


Figure 8: SLSTM architecture for sequential motion attitude error detection modeling

The algorithm's input sequence window is sequential information with m features and a o-day latency. Figure 9 shows the Layer-wise neural activations across samples for multi-sensor motion inputs.

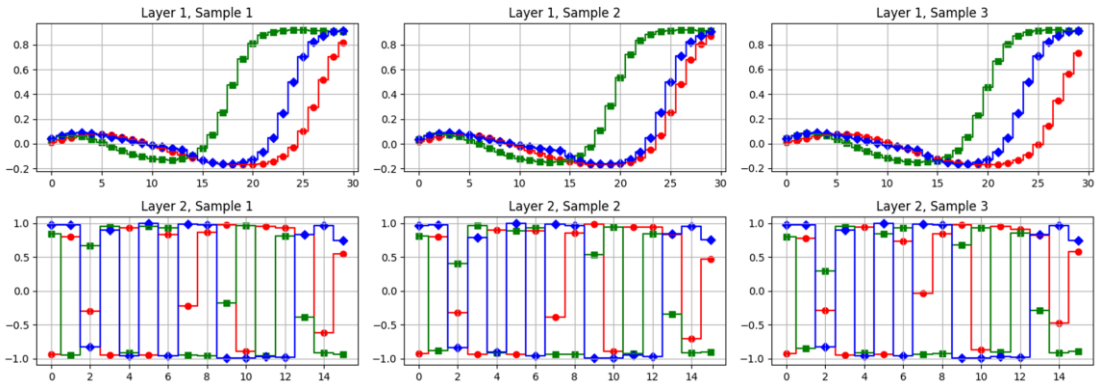


Figure 9: SLSTM layer outputs for real-time motion attitude error detection

2.4.2 Dynamic beetle antennae search selects the best possible architecture and settings for the SLSTM model

A DBAS algorithm is employed in the Motion Attitude Error Detection System to optimize multi-sensor data analysis, enabling real-time monitoring, evaluation, and correction of human movements. It efficiently searches for optimal parameters, enhances posture classification accuracy, and improves detection of subtle motion deviations. DBAS was selected as opposed to other popular optimization technologies like Bayesian Optimization and Genetic Algorithms (GA) due to the fact that it provides a single-search technique that is less expensive to compute, less complicated to parameterize, and converges more rapidly. The initial experiments show that DBAS has similar or even improved optimization performance in real-time posture error detection and requires fewer iterations. Its ability to be tuned to multi-dimensional optimization, as well as to respond to the dynamic variation of sensors, makes it especially favorable to the parameter tuning of SLSTM. Comparing the Beetle Antennae Search algorithm to other clever algorithms, it has the benefits of being a single search method with a low computational cost, low spatial complexities, and fewer necessary parameters. Its practical applicability is limited, nevertheless, by the ease with which it might slip into local optimum solutions when working with high-dimensional issues. The Beetle Antennae Search algorithm's narrow exploration direction and step's quick decline are the main factors.

The Adaptive Step Method: According to the BAS algorithm's concept, the performance of the BAS method is mostly based on the Beetle Antennae's step factor. The global search capacity will be more important, but the rate of convergence velocity would be slower if the searching step of the Antennae whisker decays slowly enough. The technique may converge before discovering the globally most effective approach because the search time decays too quickly. Therefore, to guarantee useful exploration and avoid premature convergence, a compromise between speed and the worldwide search capability of convergence is required in equation (16).

$$\begin{cases} \delta_{n+1} = \delta * \alpha^k \\ w_{left} = w_n + \delta_{n+1}\vec{c} \\ w_{right} = w_n - \delta_{n+1}\vec{c} \end{cases} \quad (16)$$

The beetle's left and right antennae locations during exploration are represented by the variables w_{left} and w_{right} , respectively. As the number of repetitions increases, the variable length of each step steadily decreases, and the difficulty of the desired issue may be taken into consideration while choosing α^k.In this case, $\delta$is the initial step length established, $\delta_{n+1}$is the initial step length $n+1th$ antenna exploration, w_n is the current position, $\delta_{n+1}\vec{c}$is the exploration step, and α is the step updating factor. For low-dimensional issues, the technique converges quickly; for high-dimensional issues with a wide exploration range, it converges slowly. The first BAS algorithm step determines the value of $\delta_{threshold}$, which is typically between 0.6 and 0.8 times. The method stays out of the rapid convergence phase as a result.

Multidirectional Approach to Exploration: The BAS method only randomly examines two symmetric directions; its exploration capabilities are restricted. The investigation of the method has various flaws, and it raises the possibility of overlooking the best answer close to the specified point. The algorithm's worldwide search capability is enhanced by the multi-directional search strategy from the outset. This technique increases the detection range by allowing the antennas to simultaneously investigate several directions. The formula for multi-directional exploration is as follows in equation (17).

$$\vec{c} = \frac{rand(m,P)}{||rand(m,P)||} \quad (17)$$

where $\vec{c}$ signifies the production of P pairs of exploration directions and $rand(m,P)$ denotes the random generation of P pairs of m-dimensional vectors. Whiskers on antennae must go in two distinct directions. Becomes too low, $rand$ inhibits the algorithm's development. P can have a range of numbers from $4\ to\ 6$, and depending on the situation, the specific value can be investigated.

Each beetle's location in equation (18) can be updated further using the following formula.

$$\begin{cases} w_{left} = w_n + \delta_{n+1}\vec{c_s}' \\ w_{right} = w_n - \delta_{n+1}\vec{c_s} \end{cases} (j \in [0,1]), \quad (18)$$

The beetle's left and right antennae locations during exploration are indicated by the variables w_{left} and w_{right}, respectively. w_n is the beetle's current position, and δ_{n+1} is the step length for the next iteration, determining how far the antennas probe in each direction. The notation $(j \in [0,1])$ indicates the selection of a specific antenna direction or weight for updating the beetle's position. This mechanism allows DBAS to balance exploration and exploitation efficiently. If $\vec{c_s}$ is used to update the position equation and $\vec{c_j}$ indicates the ith rows of $\vec{c}_s'$.

Learning Strategy Based on Lens Opposition: Lens Opposition-Based Learning (LOBL) combines inverse learning techniques with optical lens inverted physics to dynamically redirect optimization value from local extrema. In particular, LOBL uses a computational inversion "lens" to assess the "inverse image" solution when it converges onto poor solutions. In particular, LOBL uses a computational inversion "lens" to assess the "inverse image" solution when it converges onto poor solutions. Comparing incremental enhancements from current versus inverted vectors reveals better solutions. The technique increases the algorithm's ability to optimize globally, inhibits local optimization, and increases the number of feasible solutions. It is

effectively integrated with other algorithms to greatly increase the optimization capacity of the algorithm.

Equation19 defines a normalized position $\frac{g}{g'}$ for an agent within a bounded search space $\frac{b_i+a_i}{2}$. It relates the agent's current midpoint position to the best-known position $W_{best}(s)$ and a derived best position $W'_{best}(s)$, guiding iterative optimization, which corresponds to the global ideal outcome achieved via lens reflection.

$$\frac{\frac{b_i+a_i}{2}-W_{best}(s)}{W'_{best}(s)-\frac{b_i+a_i}{2}}=\frac{g}{g'} \quad (19)$$

After setting $\frac{g}{g'}=m$, $\frac{b_i+a_i}{2}$ is the midpoint of the search range, the equation W'_{best} is produced by the following transformation in equation (20).

$$W'_{best}(s)=\frac{(b_i+a_i)}{2}+\frac{(b_i+a_i)}{2m}-\frac{W_{best}(s)}{m} \quad (20)$$

It is possible to reduce the LOBL formula to the OBL formula when m equals 1. General inverse learning yields a fixed inverse solution. It alters the value of m,$W'_{best}(s)$ represents the opposite, enhancing global exploration. b_i and a_i are the lower and upper bounds of the exploration space in the $i-th$ dimension, with $\frac{(b_i+a_i)}{2}$ as the midpoint, serving as the lens center. $W_{best}(s)$ is the current global best position in the s-th dimension. The scaling factor m controls the range of inverse solutions, while $\frac{(b_i+a_i)}{2m}-\frac{W_{best}(s)}{m}$ represents the adjusted distance from the midpoint to the global best, scaled for exploration, enhancing the algorithm's capacity for global optimization. The optimization algorithm usually does a sensitive local search after a large-scale search; the interaction change in m in equation (21).

$$m=(1+(s/S)^{\frac{1}{2}})^{10} \quad (21)$$

As the number of iterations increases, the range of its inverted solution decreases, and the value of m grows; s is the current iteration and S represents the total number of iterations. Rising to the tenth power dynamically adapts, the square root grows smoothly, and the fraction s/S scales. Table 2 shows the hyperparameters of DBAS-SLSTM.

Table 2: Hyperparameters of DBAS-SLSTM

Component	Parameter	Value
DBAS Optimization	Number of beetles (N)	5
	Antennae distance (d)	0.1
	Step size (α)	0.05
	Max iterations (MaxIter)	10
Stacked LSTM	Number of layers (L)	2
	Hidden units per layer (H)	50
LSTM Training	Learning rate (η)	0.001
	Epochs	20
	Batch size	32
Training Protocol	Learning rate (η)	0.001
	Batch size	32
	Epochs	20

3 Results

The experiment setup was performed using MATLAB to implement the suggested method. To test the performance of inferences, the system was run on a PC with an Intel Core i7-9700 CPU, 16 GB RAM, and a NVIDIA GTX 1660 graphics card. The latency measurements show that the system provides feedback within 120150 ms per motion frame, which shows that it is close to real-time in this environment. The experimental results are explained in detail in this section. The following metrics were used to evaluate the suggested approach and ascertain its efficacy: Accuracy, Runtime Computation, Inference speed (FPS), mAP, and PCK@0.5. Additionally, a comparative analysis was carried out with other current methods, such as Bidirectional Recurrent Neural Network (Bi-RNN) [26], Bidirectional Long Short-Term Memory (Bi-LSTM) [26], Recurrent Neural Network with Attention Mechanism (RNN + Attention) [26], Bidirectional RNN with Attention Mechanism (Bi-RNN + Attention) [26], Bidirectional LSTM with Attention Mechanism (Bi-LSTM + Attention) [26], Bidirectional Convolutional LSTM (Bi-CLSTM) [26], Bidirectional Convolutional GRU with Attention Mechanism (Bi-CGRU + Attention) [26], Bidirectional Gated Recurrent Unit (Bi-GRU) [27], Bidirectional GRU with Attention Mechanism (Bi-GRU + Attention) [27], multi-scale feature fusion convolutional neural networks (MSFF-CNN) for posture recognition [25], Random Forest (RF) [24], Decision Tree (DT) [24], Logistic Regression (LR) [24], and Gaussian Naive Bayes (GNB) [24].

The Motion Attitude Error Detection System is an internal multi-sensor-based data detection system that operates on the internal data, which is the acceleration and angle of the joints, to monitor the movements of a human being in real time. Based on the DBAS-SLSTM model, it identifies posture and motion errors by comparing real movements and ideal patterns and gives corrective feedback. The system can be applied to sports, rehabilitation, and ergonomics. Figure 10 demonstrates patterns of movements in terms of the knee angles.

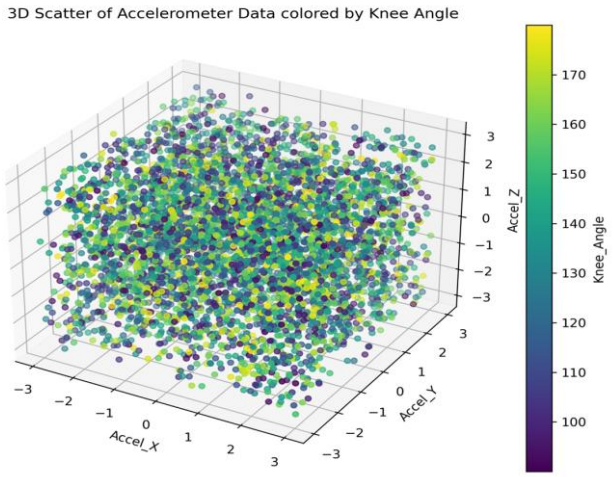


Figure 10: 3D Accelerometer scatter colored by real-time knee angle

The proposed system of detecting a motion attitude error incorporates multiple sensors, such as accelerometer measurements and joint angles, which allow the system to monitor human movement constantly. The DBAS-SLSTM model recognizes motion abnormalities in real time and allows immediate corrective feedback. This helps in improving posture, performance, and preventing injuries during sports, rehabilitation, and ergonomics. Figure 11(a) and 11(b) are Andrews and parallel plots that represent posture correctness.

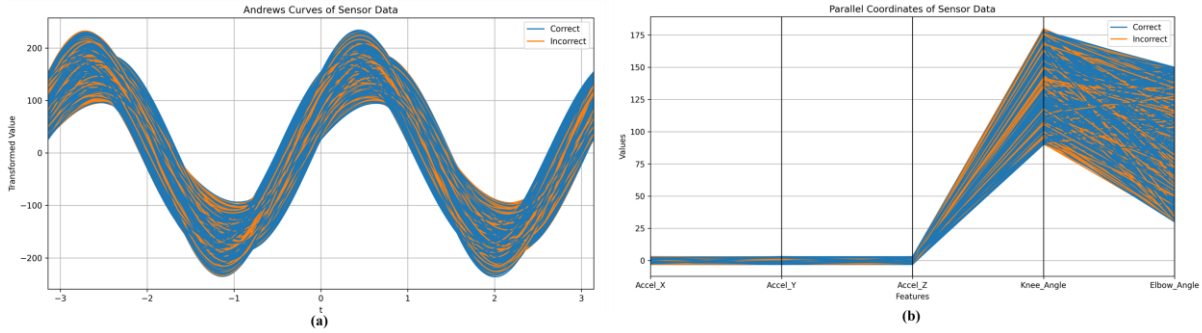


Figure 11: Visualization of (a) Andrews Curves, (b) parallel coordinates of sensor data in motion error detection

The motion attitude error detection system. The Motion attitude error system of detection is the system that utilizes real-time multisensor data, which comprises the joint angles and the acceleration, to analyze human movement. DBAS- SLSTM model is a model that examines stability and angular displacement to identify performance variations and give feedback. This promotes physical therapy, sports training, and ergonomics. Figure 12(a) represents a bubble diagram of joint stability, and Figure 12(b) represents a contour diagram of knee angle through acceleration.

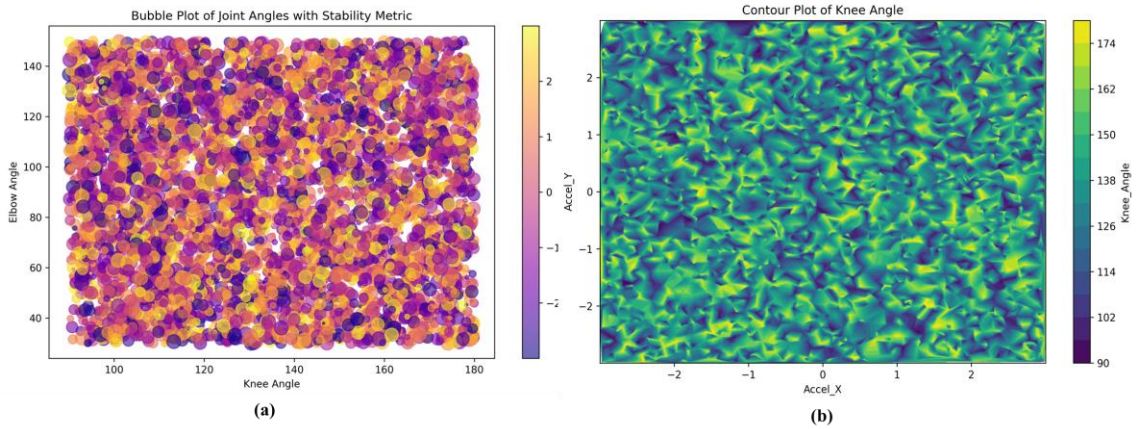


Figure 12: Analysis of (a) Bubble plot of joint angles, (b) Knee angle contour in motion detection

The proposed system combines the data of multiple sensors, such as joint angles and accelerometer signals, and implements the DBAS-SLSTM model to evaluate the human movement continuously. Deviation of anticipated paths imposes absent-mindedness in response and reconstruction. This method enhances rehabilitation, sporting training, and ergonomic assessment. Figure 13(a) represents the changes in the knee angle of the rolling means, whereas Figure 13(b) illustrates the time-dependent plot of the acceleration multi-axis stacked.

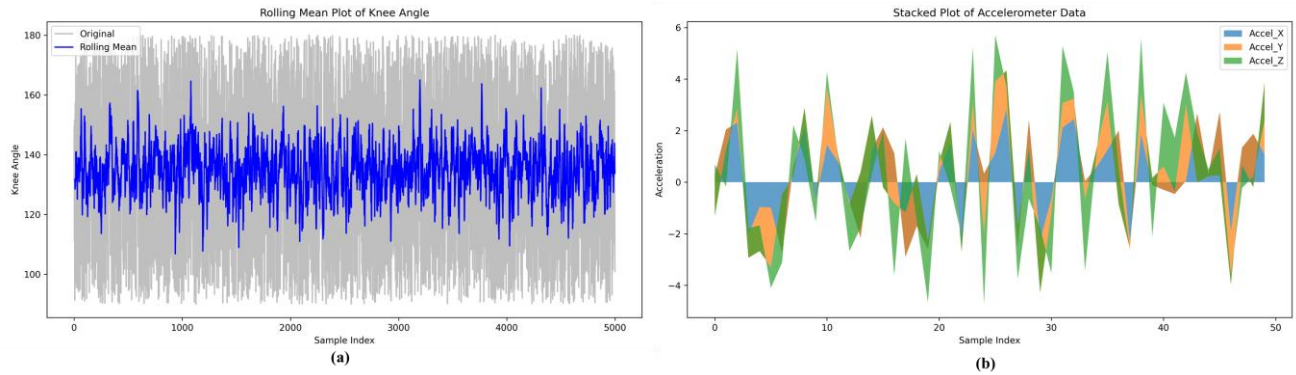


Figure 13: Depiction of (a) Rolling mean smoothing of knee angle data and (b) Stacked accelerometer signals across three spatial axes

The training and validation accuracy of the DBAS-SLSTM model in motion attitude error. Figure 14(a) indicates that the training loss decreases and the validation loss remains constant during the 200 epochs, which is a sign of successful learning. Figure 14(b) indicates that the accuracy of training and validation is high, indicating that the model can reliably and in real time capture human motion patterns.

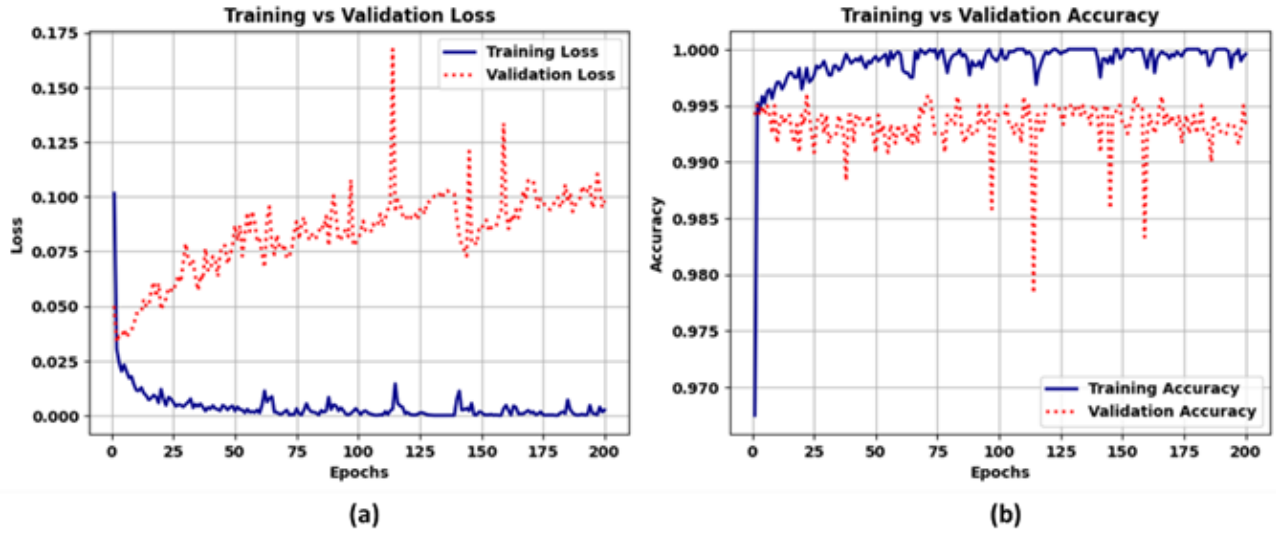


Figure 14: Training and validation (a) Loss and (b) Accuracy performance of DBAS-SLSTM for real-time motion attitude error detection

The computation of the performance of the proposed DBAS-SLSTM model was calculated by 5-fold cross-validation. The findings are presented in the form of mean values and 95% confidence intervals, and these figures illustrate the consistency and reliability of the model. The p-values are used to confirm that the obtained accuracy, the average Precision, and the actual speed of inference at that time are consistent and are not a result of random error, as reported in Table 3.

Table 3: Statistical validation of DBAS-SLSTM performance using 5-Fold cross-validation

Metric	Mean Value	95% Confidence Interval (CI)	p-value	Significance
Accuracy (%)	98.12	± 0.84	< 0.01	Significant
mAP (%)	83.40	± 1.27	< 0.05	Significant
Inference Speed (FPS)	31.5	± 0.6	< 0.05	Significant

The ablation research of the given system demonstrates the progressive effect of each component on the performance. Simple preprocessing, Kalman Filter, and Z-Score Normalization increase accuracy slightly, whereas WT increases feature representation. SLSTM is used to learn temporal dependencies, and DBAS to optimize model parameters. It has been shown that the proposed DBAS-SLSTM system can be the most accurate, with the highest mAP, and with the competitive speed of inference in real-time due to the presented effectiveness of the combined methodology, as shown in Table 4.

Table 4: Ablation research of DBAS-SLSTM on motion attitude error detection performance

Method / Component	Accuracy (%)	mAP (%)	Inference Speed (FPS)
Kalman Filter	84.12	68.45	35.2
Z-Score Normalization	86.57	70.91	34.8
Wavelet Transform (WT)	90.23	76.34	33.5
SLSTM	94.78	80.12	32.1
DBAS	96.85	82.74	31.7
DBAS-SLSTM [Proposed]	98.12	83.40	31.5

The runtime computation results indicate varying efficiency among the evaluated techniques. RF achieved 6.879s, DT 3.063s, LR 17.32s, and GNB 0.347s. Among traditional methods, GNB demonstrated the fastest performance. However, the proposed DBAS-SLSTM model outperformed all, recording the lowest runtime of 0.273s, highlighting its superior computational efficiency. This demonstrates that DBAS-SLSTM not only surpasses conventional machine learning methods but also offers a highly optimized solution for real-time motion attitude error detection. Table 5 shows execution times of methods, highlighting the fastest approach.

Table 5: Runtime comparison of techniques for motion error detection

Methods	Runtime Computation (s)
RF [24]	6.879
DT [24]	3.063
LR [24]	17.32
GNB [24]	0.347
DBAS-SLSTM [Proposed]	0.273

The comparative evaluation of various deep learning architectures for the task shows that traditional recurrent models like Bi-RNN, Bi-LSTM, and Bi-GRU achieved accuracies of 77.52%, 85.76%, and 83.52%, respectively. Adding attention mechanisms significantly improved performance, with Bi-LSTM + Attention reaching 92.88% and Bi-GRU + Attention achieving 94%. Advanced hybrid models such as Bi-CGRU + Attention attained 96.10%. The proposed DBAS-SLSTM outperformed all existing methods with the highest accuracy of 98.12%, demonstrating superior feature learning and classification capabilities. Table 6 shows the performance of models in the Motion Attitude Error Detection System.

Table 6: Accuracy comparison of motion error detection in physical exercise

Methods	Accuracy
Bi-LSTM [26]	85.76%
Bi-LSTM + Attention [26]	92.88%
RNN + Attention [26]	82%
Bi-RNN + Attention [26]	79.77%
Bi-RNN [26]	77.52%
Bi-CLSTM [26]	90.10%
Bi-CGRU + Attention [26]	96.10%
Bi-GRU [27]	83.52%
Bi-GRU + Attention [27]	94%
DBAS-SLSTM [Proposed]	98.12%

The comparison of human pose estimation methods shows that the proposed DBAS-SLSTM outperforms MSFF-CNN across all metrics. In particular, DBAS-SLSTM has a larger mAP of 83.4% than 78.6% of MSFF-CNN, which is more accurate in locating key points. It achieves a higher overall pose prediction, which is indicated by its PCK@0.5 score of 94.2%, which is higher than MSFF-CNN of 91.5%. DBAS-SLSTM has a higher key point accuracy of 91.3% compared to the 86.8% of MSFF-CNN, which further proves that it is better at estimating joints accurately. These measures were estimated through the process of extracting multi-sensor data streams, extracting joint key points, and comparing them with ground truth postures based on annotated reference movements. Additionally, DBAS-SLSTM offers faster inference at 31.5 FPS versus

27.3 FPS, making it both more accurate and efficient for real-time applications. Overall, DBAS-SLSTM is the superior method. Table 7 and Figure 15 display the system monitoring, evaluating, and correcting human movements efficiently.

Table 7: Real-time Multi-Sensor Motion Attitude Error Detection Visualization

Method	mAP(%)	Keypoint accuracy (%)	PCK@0.5 (%)	Inference speed (FPS)
MSFF-CNN [25]	78.6	86.8	91.5	27.3
DBAS-SLSTM [Proposed]	83.4	91.3	94.2	31.5

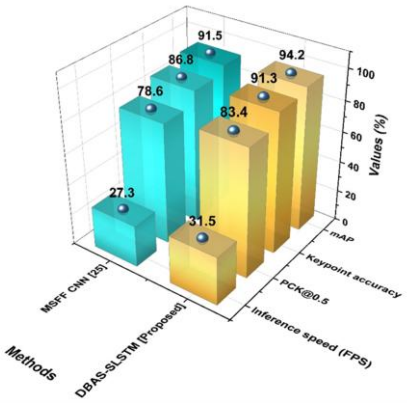


Figure 15: Performance Comparison of Pose Estimation Methods of mAP, keypoint accuracy, PCK@0.5, and Inference speed

4 Discussion and future work

The motion attitude error detection system utilizes multi-sensor data to continuously monitor and evaluate human movements, accurately identifying posture deviations in real time, enabling corrective feedback to enhance performance, prevent injuries, and optimize overall physical training outcomes. The existing methods had significant limitations. Conventional machine learning techniques like RF, DT, LR, and GNB [24] were fast and interpretable and cannot reflect complicated temporal correlations in high-dimensional motion data, thus cannot achieve a high level of accuracy. Deep learning models such as CNN and CNN-LSTM models [25] could be very sensitive to detect spatial features, although these models tend to fail to capture long-term temporal dynamics, making them less sensitive to minor deviations in posture. Bi-RNN, Bi-LSTM, Bi-CLSTM, and Bi-CGRU (with or without attention) [26,27] and bidirectional architectures enhance temporal modeling, but were characterized by high computational complexity and increased memory usage, longer inference time, and overfitting risks, limiting the practical implementation to real-time.

To overcome these limitations, the proposed DBAS-SLSTM network minimizes manual tuning and overfitting, and the proposed framework effectively learns both short-term and long-term temporal variations of multi-sensory data. As a result, the system attains enhanced accuracy, robustness, and real-time speed of motion attitude error detection in feasible healthcare applications.

Moreover, the system is developed in such a way that it has real-world flexibility. It is capable of accommodating variations in various users (body types and skill levels), exercise types, and moderate sensor noise or misalignments with multi-sensor fusion and adaptive DBAS parameter tuning. Initial analysis shows that there is a stable performance during these variations. Future efforts will concentrate on the widespread experimental research in the diverse real-life scenarios and improving resilience to extreme sensor errors to guarantee a dependable and exact feedback in the real-world application.

Although the proposed approach had shown good results in the analyzed and tested conditions, its extrapolability to unknown participants and activities has not been thoroughly tested. Differentiations in the morphology of the body and the manner of executing the exercises can influence the consistency of features, especially in high intra-class variation exercises. Future analyses will include cross-user and cross-exercise measures to measure robustness more accurately.

The future scope was integration with wearable devices, AI-driven personalized feedback, expanded movement libraries, and improved sensor fusion algorithms, which could enhance accuracy, adaptability, and real-time corrective guidance across rehabilitation, sports training, and ergonomics applications.

However, practical deployment may face challenges such as sensor calibration, maintaining data privacy, and ensuring participant safety, which should be carefully addressed during real-world implementation. The future application will introduce the system to portable embedded systems and wearable architecture to further lower the latency and test the real-time performance in field-based training set-ups to ensure useful utilization in the real-life exercise context.

5 Conclusion

A real-time Motion Attitude Error Detection System utilizes multi-sensor data to continuously monitor, evaluate, and correct human movements, enhancing performance accuracy, preventing injuries, and supporting safe, optimized physical training. The dataset was a Motion Attitude Error Detection System integrating various physical exercises. Through Kalman filtering, normalization, and wavelet-based feature extraction, the system captures detailed motion characteristics. The proposed DBAS-SLSTM framework effectively detects posture errors in real time and provides corrective feedback, demonstrating significant potential for enhancing exercise performance, injury prevention, and personalized training. The

proposed DBAS-SLSTM framework demonstrates high efficiency and performance, achieving a runtime of 0.273 s, accuracy of 98.12%, mAP of 83.4%, PCK@0.5 of 94.2%, and an inference speed of 31.5 FPS, indicating its capability for real-time, precise motion attitude error detection and corrective feedback. The system may face sensor noise, calibration errors, and limited adaptability to diverse body types or complex movements. Real-time processing can be computationally intensive, potentially affecting responsiveness.

Fund

This work was supported by the Young Talent Project of Hubei Provincial Department of Education Scientific Research Plan (Grant No. Q20244201) for the project titled "Real-time Police Action Recognition and Analysis System Based on Deep Learning Models" (Project Leader: Zhongyi Ni; Host Institution: Hubei University of Police).

Ethical considerations

Research presented in this study uses the Motion Attitude Error Detection Dataset obtained from Kaggle. The dataset was sourced as a publicly available resource intended for research and benchmarking in machine learning. Prior to its use, The dataset did not contain personally identifiable information (PII) or specific consent from individuals.

Informed consent

Research did not involve human participants or identifiable personal data; therefore, informed consent was not required. The dataset used is publicly available on Kaggle and was utilized strictly for research purposes in compliance with its usage terms.

References

[1] Vidya B, Sasikumar P. Wearable multi-sensor data fusion approach for human activity recognition using machine learning algorithms. Sensors and Actuators A: Physical, 2022, 341:113557. https://doi.org/10.1016/j.sna.2022.113557

[2] Huang C, Liu Y, Sun X, Wang Y. Intelligent active suspension control method based on hierarchical multi-sensor perception fusion. Sensors, 2025, 25(15):4723. https://doi.org/10.3390/s25154723

[3] Zhou L. Design of sports posture real-time monitoring system based on sensor data flow. Journal of Computational Methods in Science and Engineering, 2022, 22(6):2077–2091. https://doi.org/10.3233/JCM226400

[4] Lin FC, Ngo HH, Dow CR, Lam KH, Le HL. Student behavior recognition system for classroom environments based on skeleton pose estimation. Sensors, 2021, 21(16):5314. https://doi.org/10.3390/s21165314

[5] Liu C, Chang Z. Sensor and attitude analysis of track and field training action recognition based on artificial intelligence. Journal of Sensors, 2022, 2022:6282388. https://doi.org/10.1155/2022/6282388

[6] Xu Y. Real-Time Motion Recognition in Special Training Systems Based on the Optimized BBO-KNN Method of Motion Morphology. Informatica, 2025, 49(16). https://doi.org/10.31449/inf.v49i16.9600

[7] Chang R, Yang Z, Ning J. Inaccurate action detection algorithm for rowing machine exercise based on attention-CNN. IEEE Access, 2024, 12:114961–114973. https://doi.org/10.1109/ACCESS.2024.3445875

[8] Bu B. Evaluation on Numerical Method of Differential Formula Based on Optimal Control of Space Robot's Attitude and Motion. Informatica, 2024, 48(13). https://doi.org/10.31449/inf.v48i13.6065

[9] Lv Y, Li M, Li B. Athletic sports posture measurement algorithm based on multi-sensor combination. Journal of Computational Methods in Science and Engineering, 2022, 22(6):2065–2076. https://doi.org/10.3233/JCM-226393

[10] Wang J, Wang Z, Qiu S, Xu J, Zhao H, Fortino G, Habib M. A selection framework of sensor combination feature subset for human motion phase segmentation. Information Fusion, 2021, 70:1-11. https://doi.org/10.1016/j.inffus.2020.12.009

[11] Gu H, Jin C, Yuan H, Chen Y. Attitude and heading reference system using extended Kalman filter. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2021, 29(Suppl.1):157–180. https://doi.org/10.1142/S0218488521400092

[12] Liu F, Liu Y, Sun X, Sang H. Hierarchical multi-sensor data fusion based on unscented Kalman filter. Applied Ocean Research, 2021, 109:102562. https://doi.org/10.1016/j.apor.2021.102562

[13] Li H. Motion trajectory monitoring system based on multi-sensor data fusion. Procedia Computer Science, 2025, 262:101–107. https://doi.org/10.1016/j.procs.2025.05.033

[14] Li J, Liu X, Wang Z, Zhao H, Zhang T, Qiu S, Cangelosi A. Real-time human motion capture based on wearable inertial sensor networks. IEEE Internet of Things Journal, 2021, 9(11):8953-8966. https://doi.org/10.1109/JIOT.2021.3119328

[15] Yang H, Wang Y, Wang H, Shi Y, Zhu L, Kuang Y, Yang Y. Multi-Inertial Sensor-Based Arm 3D Motion Tracking Using Elman Neural Network. Journal of Sensors, 2022, 2022(1):3926417. https://doi.org/10.1155/2022/3926417

[16] Liu Q. Human motion state recognition based on MEMS sensors and Zigbee network. Computer Communications, 2022, 181:164–172. https://doi.org/10.1016/j.comcom.2021.10.018

[17] Brunson B, Wang J, Ma W. Modeling of IMU arrays under generic multi-sensor integration strategy. Sensors, 2024, 24(23):7754. https://doi.org/10.3390/s24237754

[18] Zhong Y, Chen X, Gao N, Jiao Z. In-motion alignment with MEMS-IMU using multilocal linearization detection. Sensors, 2025, 25(9):2645. https://doi.org/10.3390/s25092645

[19] Hajjej F, Javeed M, Ksibi A, Alarfaj M, Alnowaiser K, Jalal A, Park J. Deep human motion detection and multi-features analysis for smart healthcare learning tools. Ieee Access, 2022, 10: 116527-116539. https://doi.org/10.1109/ACCESS.2022.3214986

[20] Dai A, Xiao Y, Shen H, Zhi S, Long D. Anomaly detection of spacecraft inertial measurement units using multi-scale dependency neural network. Acta Astronautica, 2025, 232:204–214. https://doi.org/10.1016/j.actaastro.2025.03.014

[21] Li J, Liu X, Wang Z, Zhang T, Qiu S, Zhao H, Cangelosi A. Real-time hand gesture tracking for human–computer interface based on multi-sensor data fusion. IEEE Sensors Journal, 2021, 21(23):26642-26654. https://doi.org/10.1109/JSEN.2021.3122236

[22] Gu Y, Han Y, Liu X, Zhang N, Zhang X, Pan M, Liu, T. A flexible sensor and mimu-based multisensor wearable system for human motion analysis. IEEE Sensors Journal, 2023, 23(4):4107-4117. https://doi.org/10.1109/JSEN.2022.3233653

[23] Liu W, Liu Y, Bucknall R. Filtering-based multi-sensor data fusion for unmanned surface vehicle navigation. Journal of Marine Engineering & Technology, 2023, 22(2):67–83. https://doi.org/10.1080/20464177.2022.2031558

[24] Naseer A, Raza A, Afzal H, Smerat A, Fitriyani NL, Gu Y, Syafrudin M. Human pose estimation in physiotherapy fitness exercise correction using novel transfer learning approach. PeerJ Computer Science, 2025, 11:e2854.

[25] Chen H, Fan R. Improved convolutional neural network for precise exercise posture recognition and health indicator prediction. Scientific Reports, 2025, 15(1):21309. https://doi.org/10.1038/s41598-025-01854-x

[26] Rao P, Asha CS, Rao PR. Real-time posture correction of squat exercise using deep learning. IEEE Access, 2025, 13:39557–39571. https://doi.org/10.1109/ACCESS.2025.3545207

[27] Chariar M, Rao S, Irani A, Suresh S, Asha CS. AI trainer: Autoencoder based approach for squat analysis and correction. IEEE Access, 2023, 11:107135-107149. https://doi.org/10.1109/ACCESS.2023.3316009

A Multi-Task Framework for Intelligent Review and Semantic Consistency Detection in Scientific Project Documentation Using Large Language Models

Zehui Zhang[1], Qiang Yao[2], Jingman He[1*], Bing Wen[1], Yuying Gong[2], Lin Zhou[1], Jian Wei[1]
[1]Inner Mongolia Power (Group) Co., Ltd. Digital Research Branch, Hohhot, 010090, Inner Mongolia, China
[2]Inner Mongolia Power (Group) Co., Ltd. Hohhot, 010010, Inner Mongolia, China
E-mail: Jingmanhe@outlook.com
*Corresponding author

Keywords: large language model, scientific and technological project documentation, smart review, semantic consistency analysis, technical parameter conflict detection

Received: September 27, 2025

With the intensive and active global technological innovation, the quality of technology project document review is crucial. However, traditional manual review is time-consuming and biased, and rule driven tools are difficult to handle multimodal and dynamic terms. Large Language Models (LLMs) provide a new path for this. This study proposes a multi-dimensional intelligent evaluation multi task architecture based on LLM, with "pre training fusion knowledge enhancement fine tuning adaptation multi task reinforcement learning continuous optimization" as the core: in the pre training stage, the domain knowledge graph (DKG) is fused to construct a terminology library and form a mixed reasoning ability, and in the fine tuning stage, conflict detection, logic evaluation, and consistency analysis sub task heads are designed based on open source models such as LLaMA (Large Language Model Meta AI), while introducing reinforcement learning optimization strategies and achieving model lightweighting. The experiment selected 800 documents from a national research institution as the test set, and the results showed that the accuracy of technical parameter conflict detection was 92.7% (18.3 percentage points higher than traditional rule engines), the F1 value of logical consistency evaluation was 89.1% (13.7 percentage points better than keyword matching methods), the average recall rate of cross document semantic consistency analysis was 87.6%, and processing 100000 words of text only took 2.5 hours (6 times more efficient than manual labor). After integrating reinforcement learning, the coverage of implicit association recognition was increased to 89.7%, the false alarm rate was reduced to 4.3%, and the parameter compression of 35% still maintained 91.2% of core performance. This architecture breaks through traditional limitations and provides technical support for the digitization of technology project management.

Povzetek: Študija pokaže, da lahko veliki jezikovni modeli z domenskim znanjem avtomatizirajo pregled tehničnih dokumentov ter ga naredijo hitrejše in natančnejše od ročnega ali pravilno zasnovanega pristopa.

1 Introduction

Against the backdrop of accelerated global scientific and technological innovation and the in-depth development of the knowledge economy, the life cycle management of scientific and technological projects is becoming increasingly important as the core carrier for promoting technological breakthroughs and industrial upgrades [1, 2]. As a systematic record of R&D activities, scientific and technological project documents not only carry key information, such as technical route design, innovation point condensation, and implementation process verification, but also serve as the basis for project review, achievement transformation, intellectual property protection, and subsequent technical iteration [3]. With the continuous improvement of scientific research complexity, the scale of scientific and technological project documents continues to expand, and the cross-cutting content is significantly enhanced, involving complex features such as multi-disciplinary terminology fusion, technical parameter correlation, logical chain nesting, etc., which makes the document review work face unprecedented challenges [4].

Traditional document review of science and technology projects mainly relies on the experience of manual experts. Reviewers need to possess a cross-domain knowledge reserve and an in-depth technical understanding, and complete quality assessment through paragraph-by-paragraph reading, key information extraction, and logical verification [5, 6]. However, such methods have significant limitations: on the one hand, manual review is time-consuming and labour-intensive.

For documents with hundreds of thousands or even millions of words, the review cycle often lasts for weeks or even months, which is difficult to meet the rapid iterative scientific research project management needs [7]; On the other hand, the review results are easily influenced by subjective factors, and different experts may have cognitive differences on the rigor of technical description and logical rigor, resulting in insufficient consistency of review conclusions [8]; In addition, in the face of dynamically updated technical terms and implicit cross-document semantic associations, it is difficult for manual review to achieve efficient and accurate multi-dimensional coverage [9].

In recent years, although rule-driven automated review tools have alleviated manual pressure to some extent, they rely on preset keyword matching, regular expression rules, or structured templates and can only address formal surface errors. For deep semantic problems, such as technical parameter conflicts, lack of logical coherence, and cross-document semantic inconsistency, the rule engine struggles to effectively identify them due to its limited context understanding and knowledge generalisation capabilities [10, 11].

The emergence of Large Language Model (LLM) provides a new solution to the above dilemma [12]. As a general language understanding and generation system based on deep learning, LLM has acquired powerful capabilities for context representation, semantic reasoning, and cross-domain knowledge transfer through massive text training. It can capture implicit logical relationships, dynamic term associations, and cross-document semantic mapping in texts [13-15]. Compared with traditional methods, LLM does not need to rely on manual preset rules and can adapt to the review needs of specific fields through the paradigm of "pre-training + fine-tuning", showing potential in complex tasks such as technical parameter conflict detection, logical coherence evaluation, and cross-document semantic consistency analysis [16, 17]. However, existing research mostly focuses on the review scenario of general texts. Aiming at the highly specialised and structured text type of scientific and technological project documents, the adaptability of dynamic terminology processing, multi-dimensional review task integration, and long text semantic consistency modelling still needs to be further explored.

Hybrid reasoning combines the rigor of symbolic reasoning rules with the semantic understanding ability of large models, which not only solves rigid requirements such as compliance verification in reviews, but also enhances the depth of semantic consistency analysis, avoiding deviations from a single reasoning mode; Multimodal support breaks through the limitations of traditional text review and can parse non textual elements such as charts and formulas in documents, achieving collaborative review across multiple information dimensions and adapting to the multimodal characteristics of technology project documents; Continuous learning enables the framework to dynamically absorb new field evaluation standards, industry terminology, and historical evaluation experience, continuously optimizing evaluation accuracy and generalization ability. The synergy of the three not only enhances the adaptability of the framework in practical scenarios, but also forms a core novelty that differs from traditional evaluation methods, improving the efficiency and reliability of technology project document review.

In response to the challenges faced by technology project document review in the context of global technological innovation, such as manual time consumption, subjective bias, and limitations of rule driven tools, this study proposes a multi task intelligent review framework that integrates large language models (LLM) and domain knowledge graphs (DKG). The core of the framework is the "pre training fusion knowledge enhancement fine-tuning adaptation multi task reinforcement learning continuous optimization" process. In the pre training stage, DKG is integrated to build a terminology library and form a hybrid inference capability. In the fine-tuning stage, conflict detection, logic evaluation, and consistency analysis subtasks are designed based on open-source models such as LLaMA. Reinforcement learning optimization strategies are introduced to achieve model lightweighting. The experiment used 800 documents from a research institution in a certain country as the test set, and the results showed that the accuracy of technical parameter conflict detection reached 92.7% (18.3 percentage points higher than traditional rule engines), the F1 value of logical consistency evaluation was 89.1% (13.7 percentage points higher than keyword matching methods), the average recall rate of cross document semantic consistency analysis was 87.6%, and processing 100000 words of text only took 2.5 hours (6 times more efficient than manual labor); After integrating reinforcement learning, the coverage of implicit association recognition increased to 89.7%, the false positive rate decreased to 4.3%, and the 35% parameter compression still maintained 91.2% of the core performance. This framework breaks through traditional limitations and provides technical support for the digitization of technology project management, while also pointing out improvement directions in interpretability, scalability, and multimodal integration.

2 Theoretical basis and principle technology

2.1 Semantic representation theory of large language model

In real life, the description of the same object involves multiple dimensions and is stored in various forms, such as text, audio, and video. In semantic representation technology, "modal" refers to the way information is encoded. Multimodal semantic representation learning involves representation learning tasks of multiple modalities. Multimodal data describes the same object from different perspectives, which are usually complementary semantically and provide richer information. However, computers have not been able to use multimodal data for comprehensive cognition

effectively. The feature vectors of different modes reside in distinct feature spaces, and features with similar semantics exhibit significant differences, which is referred to as the modal heterogeneity gap problem [18, 19]. Therefore, the core challenge of multimodal representation learning techniques is to find ways to reduce heterogeneity gaps while maintaining semantic consistency. In recent years, researchers have made significant progress in various fields through the application of deep learning technology. Deep learning algorithms enable models to autonomously learn single-modal representations, replacing various features of artificial design [20].

As shown in Figure 1, the methods can be categorised into three main frameworks based on their underlying structure: joint representation framework, coordinated representation framework, and encoder-decoder framework. Each framework has its unique structure and strategy to fuse multimodal features.

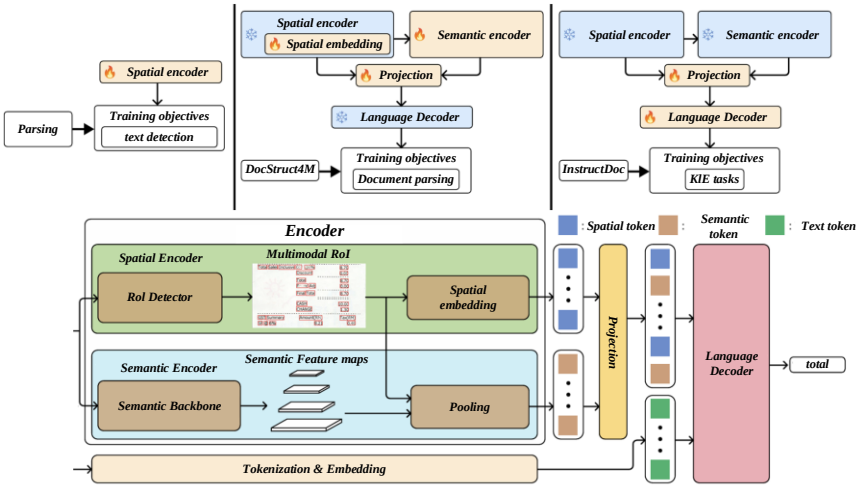


Figure: 1 Multimodal semantic representation framework

In the field of deep learning, researchers use a variety of features to improve algorithm efficiency, which is also applicable to multi-modal deep learning [21]. To deal with the heterogeneity between modes, the joint characterization framework maps features to shared subspaces to fuse multi-modal features [22]. After each modal feature is coded by an independent neural network, it is mapped to a common subspace, and the common semantics are expressed by a fusion vector, as shown in Equation (1).

$$z = F(w_1^T v_1 + w_2^T v_2) \quad (1)$$

In the shared layer, the activation function is denoted by z, the single-modal network output is denoted by v, and the weight matrix is denoted by w. The joint representation framework does not need to explicitly coordinate different modalities, and its common subspace contains modal-independent semantic information to realize semantic transfer between modalities, but it is not suitable for inferring a single modal representation [23, 24].

In multimodal learning, coordinated representation framework is a popular method. It learns independent and coordinated semantic features under specific constraints, which helps to retain modal-specific features. Coordinated representation methods are divided into two categories: based on cross-modal similarity and based on cross-modal correlation [25]. The former learns a common subspace to measure the distance between modal vectors, while the latter learns a shared subspace to maximize the modal representation set correlation.

Cross-modal similarity methods learn coordinated representations under similarity measure constraints. Taking visual and text representations as examples, the feature vectors are represented by (v; t) respectively, and the optimization goal is formula (2).

$$\begin{aligned} \text{rankLoss} = &\sum_{v}\sum_{t^-} max(0, \alpha - S(v,t) + S(v,t^-)) \\ &+ \sum_{t}\sum_{v^-} max(0, \alpha - S(t,v) + S(t,v^-)) \end{aligned} \quad (2)$$

In the model, α is the filling value, S (*,*) represents the similarity function, and t- and v- are negative samples randomly selected from the training set. The constraint is based on the Euclidean distance, reducing the sample-pair distance distanceLoss, and ensuring that the visual semantic features match the text description. The mathematical expression is shown in Equation (3).

$$distanceLoss = \sum_{(v,s)\in D} \| T_v v - T_s s \|_2^2 \quad (3)$$

Tv and Ts represent the mapping matrix of the video with its text description s, respectively.

The cross-modal correlation technique learns the correlations of two modal datasets through deep neural networks, mapping them to shared spaces to enhance correlations. Deep correlation analysis (DCCA) is an example of this type of technique. These techniques usually focus on the semantic information shared among modalities, while the coordinated representation framework preserves the unique characteristics of modalities. Different modalities are encoded in independent networks, allowing independent inference [26].

The encoder-decoder characterization framework is used for multimodal translation tasks and consists of an encoder and a decoder. The encoder embeds the source

modality into the hidden layer v, and the decoder generates a new sample of the target modality based on this. Taking the visual description task as an example, the model maximizes the text T log-likelihood given the visual content V and parameters θ. The learning objectives can be expressed by equations (4)-(5):

$$\theta^* = \arg\max_{\theta} log\ p(S \mid T;\theta) \quad (4)$$

$$log\ p(S \mid T;\theta) = \sum_{t=0}^{N} log\ p(S_{w_t} \mid V, S_{w_1}, ..., S_{w_{t-1}}) \quad (5)$$

Swi represents the i-th word in the sentence and N is the total number of words in the sentence. The encoder-decoder model not only learns the latent vectors of the source modality, but also closely correlates with the source and target modality. The loss function guides the encoder so that its hidden layer representation captures the shared semantics of the two modalities. The framework builds consistent semantic features that are suitable for cross-modal semantic embedding learning. The model transforms modes through encoder-decoder network, and measures sentence similarity by BLEU and other evaluations, and image similarity is evaluated by discriminator. This characterization framework can generate new samples based on a modality, but due to the complexity of the generator, it requires high computing resources.

2.2 Theoretical framework of semantic consistency analysis

Cross-modal semantic consistency technology learns the semantic association between different modalities by analyzing the diverse patterns of samples and converts them to Hamming space to learn unified binary coding, thereby reducing the semantic differences between modalities [27]. Inter-media hashing (IMH) combines the unity within and between modes to discover the shared Hamming space, allowing different kinds of media data to be consistently connected and expressed. Cross-modal retrieval semantic consistency hash (SCH) employs non-negative matrix factorisation to learn the latent semantic space, and in the shared semantic space, the semantic unity within each modality is maintained by the neighbour algorithm [28]. Semantic label subspace relational hash (SRLCH) uses the class label transformation of Hamming subspace, utilises the subspace relational information, and designs a symmetric structure to optimise the distance between binary code and relational information, so that similar data of different modes are closer in the low-dimensional Hamming subspace. Scalable Semantic Enhancement Hash (SPECH) considers that heterogeneous data share the same semantic class information, ensures class consistency of latent spatial features, and reduces intra-class differences among heterogeneous modal samples. Large-scale semantically consistent hashing (SCCH) embeds class structure information into the shared binary code of learning samples, ensuring consistency in common representations across modes and achieving good results. Although these methods incorporate the embedding of semantic relevance and class information, it is necessary further to integrate the underlying semantic relevance and class structure information to obtain a more discriminative hash code. Based on this, this paper achieves semantic consistency in universal representation and enhances the performance of cross-modal retrieval.

The semantic similarity method studies the correlation between different modes by minimising the error between shared semantic labels and binary embeddings. BATCH regards labels as a third modality and embeds rich, similar information to maintain semantic consistency by minimising the distance difference problem [29, 30]. SEAH employs an asymmetric strategy, embedding labels in hash codes and utilising pairwise similarity matrices to maintain inter-sample correlations. These methods construct a similarity matrix and embed high-level semantic similarity information into the hash code to obtain accurate retrieval results, but they ignore the importance of low-level features. CRE maintains inter-modal similarity and addresses heterogeneity and integration complexity by minimising the Euclidean distance between multimodal features and binary codes [31, 32]. In this paper, we propose a multi-level semantic similarity approach by combining high-level semantic labels with low-level multimodal features to address the problem of low expressiveness in hash codes effectively.

3 Construction of intelligent review and semantic consistency analysis model

3.1 Design of multi-modal document fusion architecture

The TE Trans model architecture takes the document semantic parsing module as the core input layer, and connects three core modules: layered feature encoding component, cross modal semantic alignment component, and consistency score output component; In terms of encoder decoder structure, TE Trans differs from Seq2Seq's unidirectional sequence mapping architecture in that it adopts a bidirectional interactive encoding and decoding structure. The encoder captures the global semantics of the document through multi granularity attention, and the decoder introduces a feedback mechanism to dynamically adjust the semantic matching strategy, rather than relying on fixed sequence generation logic; In terms of attention mechanism, compared with the local window attention of nCG ESM baseline, TE Trans innovative design domain adaptive multi head attention mechanism optimizes attention allocation through pre training weights in the field of scientific documents, which can simultaneously focus on the keywords of review rules and the core semantic blocks of documents, achieving more accurate semantic association capture and effectively solving the problem of semantic loss in long documents of Seq2Seq and local semantic deviation in nCG ESM.

Semantic perturbation testing is a method that applies subtle semantic perturbations like synonym replacement, sentence structure transformation, and redundant

information addition to document text. It verifies that the large language model can accurately identify the document's core semantics and maintain review result stability in perturbation scenarios. The core is to test the model's robustness to non - critical semantic changes. Meanwhile, the multi - agent collaboration mechanism is a collaborative work model. It constructs an intelligent agent cluster with clear division of labor, and efficiently completes multi - dimensional semantic extraction, cross - chapter consistency verification, and review conclusion generation of technology project documents through information exchange, task collaboration, and result fusion among agents, aiming to improve the comprehensiveness and efficiency of semantic analysis.

The current research architecture is specifically designed for text-based document processing, image processing, and multimodal fusion scenarios to cover the diverse review requirements of technology project documents; Under this architecture, the core challenge of cross modal retrieval is modal heterogeneity (differences in text semantic logic, image visual intuitiveness, and multimodal information collaboration). To address this, this paper proposes an adversarial method based on semantic consistency to ensure that different modal samples of the same semantic class are closer in the shared semantic subspace; Figure 2 The feature mapper and modal classifier of the framework both adapt to multimodal data characteristics. The former enhances semantic consistency by shortening the distance between the sample and the category center in a single mode, the distance between the cross-modal sample and the category center under specific semantics, and the distance between the centers of different modal categories in the same semantic category. The latter identifies the sample modal types, and together they help to obtain the cross-modal feature representation of close semantic consistency, providing support for the review task of science and technology project documents.

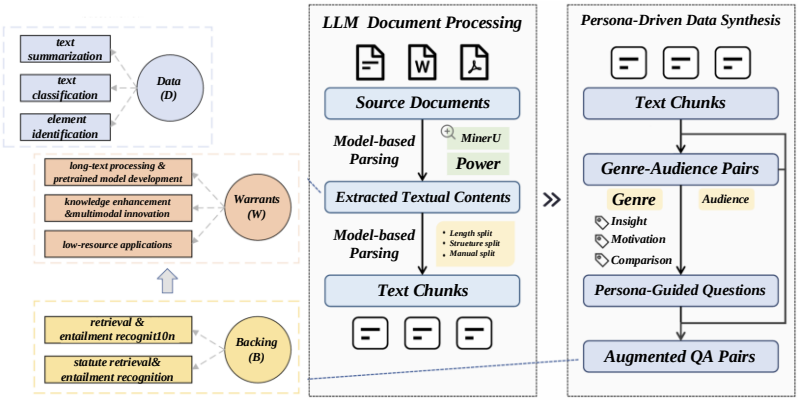


Figure 2: Multimodal document fusion architecture

In this study, n pairs of image-text instances are constructed, denoted as set O={Oi=[vi,ti]}ni=1, and each instance Oi is equipped with a semantic label set li. The image, text, and semantic labels of the instance are denoted as V, T, and L, respectively. In order to ensure that the sample mappings of the same semantic category are close, a semantic consistency regularization term is introduced. Calculate the class center CV and CT of the image feature V and the text feature T. The definition formula of the class center is shown in Equation (6).

$$\mathbf{c}_j^v = \frac{1}{n_j}\sum_{i=1}^{n_j}\mathbf{v}_j^i, \quad \mathbf{c}_j^t = \frac{1}{n_j}\sum_{i=1}^{n_j}\mathbf{t}_j^i \quad j=1,2,...,k \ (6)$$

cvj and ctj are the class centers of the j-th class in visual and text modality respectively. This study aims to minimize the intra-class distance (d_1) and the class center distance (d_2) of the same semantic class between different modalities. While reducing the distance between the center of the modal class and the same semantic sample in another modal, denoted by d_3 and d_4. See (7)-(9) for specific formulas.

$$d_1 = \frac{1}{k}\sum_{j=1}^{k}\left[\frac{1}{n_j}\sum_{i=1}^{n_j}(||\mathbf{c}_j^v - \mathbf{v}_j^i||_2 + ||\mathbf{c}_j^t - \mathbf{t}_j^i||_2\right] (7)$$

$$d_2 = \frac{1}{k}\sum_{j=1}^{k}||\mathbf{c}_j^t - \mathbf{c}_j^v||_2 \ (8)$$

$$d_3 = \frac{1}{k}\sum_{j=1}^{k}(\frac{1}{n_j}\sum_{i=1}^{n_j}||\mathbf{c}_j^v - \mathbf{t}_j^i||_2), d_4 = \frac{1}{k}\sum_{j=1}^{k}(\frac{1}{n_j}\sum_{i=1}^{n_j}||\mathbf{c}_j^t - \mathbf{v}_j^i||_2) \ (9)$$

The feature mapping loss function Lemb can be obtained by formula (6), as shown in formula (10):

$$L_{emb}(\theta_T,\theta_V,\theta_{imd}) = \alpha\cdot L_{imi} + \beta\cdot L_{imd} + \gamma\cdot L_{reg} \ (10)$$

Lintra is an intra-class compactness constraint, and Lreg is a regular term. α, β, and γ are hyperparameters, which are used to adjust the weights of each part. The target loss function Lloss is shown in Equation (11):

$$L_{loss} = L_{emb}(\theta_V,\theta_T,\theta_{imd}) - L_{adv}(\theta_D) \ (11)$$

Based on the principle of generative adversarial network, the best feature representation is trained to represent a minimax adversarial game involving two sub-processes, as shown in Equations (12)-(13):

$$(\hat{\theta}_V,\hat{\theta}_T,\hat{\theta}_{imd}) = arg\min_{(\theta_V,\theta_T,\theta_{imd})}(L_{emb}(\theta_V,\theta_T,\theta_{imd}) - L_{adv}(\hat{\theta}_D)) \ (12)$$

$$\hat{\theta}_D = argmax(L_{emb}(\hat{\theta}_V,\hat{\theta}_T,\hat{\theta}_{imd}) - L_{adv}(\theta_D)) \ (13)$$

Table 1: Core Module processing flow

Key Stage	Core Actions	Target Effects
Preprocessing Phase	1. Split document multimodal data & parse table structure; 2. Supplement DKG core concepts/relations	1. Achieve data structuring; 2. Ensure DKG covers review key info (e.g., technical indicators, budget rules)
Knowledge Fusion Phase	1. LLM extracts text entities & links to DKG; 2. Convert DKG rules into LLM prompts	1. Supplement DKG dynamic info; 2. Restrict LLM to follow domain norms, reduce errors
Semantic Analysis Phase	1. Compliance: LLM verifies against DKG rules (e.g., budget ratio); 2. Consistency: LLM compares cross-module semantics + DKG verifies entity relations; 3. Innovation: LLM identifies innovations + DKG matches existing technologies	1. Identify non-compliant content; 2. Locate semantic contradictions (e.g., cycle conflicts); 3. Evaluate innovation novelty
Result Output Phase	Generate structured review report (issue list, scores, basis)	Intuitively present review conclusions, support quick decision-making
Table Parsing	Extract header/row-column data, identify data types (e.g., amount, cycle)	Convert unstructured tables into machine-understandable structured data
Table Semantic Linking	1. LLM links table data to document text (e.g., budget table → budget description); 2. DKG matches table entities (e.g., equipment cost → reasonable range attribute)	1. Eliminate semantic separation between tables and text; 2. Realize domain rule verification for table data
Table Consistency Check	LLM compares cross-table data (e.g., sub-project → total project budget); DKG verifies entity consistency (e.g., person-in-charge info)	Locate cross-table data contradictions, ensure data logical coherence

The framework represented in Table 1 integrates Large Language Models (LLMS) and Domain Knowledge Graph (DKGs) for intelligent review of technology project documents, with a focus on multimodal table processing: its hybrid method architecture includes four stages (structured data preprocessing and DKG supplementation, knowledge fusion of entity linking and prompt transformation, semantic analysis of compliance/consistency/innovation checks, and output of structured reports to assist decision-making), and its polymorphic table processing covers three steps (parsing to convert unstructured tables into machine-readable data, semantic linking to connect tables with text/entities through LLMS/DKG, consistency checking to ensure data logic through cross table comparison and entity verification), ultimately combined with LLMs' semantic understanding. Provide a systematic solution for intelligent document review and table processing based on domain constraints of DKG.

3.2 Dynamic semantic consistency detection module

The dynamic semantic consistency detection module aims to resolve semantic contradictions and logical fractures in the process of multi-version iteration, cross-chapter association, and dynamic evolution of technical parameters for scientific and technological project documents (a core scenario of intelligent review). This module is centered around a large language model, integrating domain knowledge graphs and adaptive semantic modeling techniques to construct a composite framework that includes local text validation and global semantic analysis. It can dynamically parse document structures, extract key semantic units, and locate hidden conflicts.

The adversarial learning component is a key module for improving the robustness and semantic analysis accuracy of document intelligent review models. Its core is to generate adversarial samples by perturbing the word embeddings of input text. The Fast Gradient Sign Method

(FGSM) algorithm is used to apply small perturbations to the word embedding vectors along the gradient direction of the model loss function, constructing adversarial text that can induce model misjudgment; This design draws on the NLP adversarial training framework and subsequent optimization work for text embedding perturbations. By incorporating adversarial samples into the model training process, the model can maintain stable evaluation results and semantic consistency analysis ability even when facing inputs with semantic similarity but subtle embedding perturbations.

In terms of technical implementation, the module adopts hierarchical semantic modeling, encodes paragraphs to generate semantic feature matrices through pre trained models, combines dynamic knowledge distillation to embed domain rules, and then captures cross chapter dependencies with sliding window attention and graph neural networks. The core innovation is the dynamic conflict resolution algorithm, which can generate consistency reports, trigger multimodal verification and repair optimization, and support incremental update adaptation technology scheme adjustment, providing intelligent semantic quality assurance tools for scientific research management.

The 800-document dataset of this study covers multiple domain documents such as technology project application forms, review opinions, and technical specifications. The annotation process is completed through three steps: "expert initial bidding cross validation ambiguity correction" to ensure consistency and accuracy of annotations; Adopting an 80/20 training/testing segmentation strategy ensures that the training data meets the model's learning needs while objectively verifying generalization ability through independent test sets; The selection of baseline models is based on the principle of "domain adaptation+interpretability", with priority given to pre trained models (BERT base) and traditional machine learning models (SVM) that perform stably in text classification and semantic matching tasks, in order to highlight the performance advantages of large language models; The performance evaluation adopts a combination of generative metrics (BLEU, ROUGE) and task specific metrics (accuracy, F1 score). The former measures the coherence and relevance of the generated review comments, while the latter quantifies the accuracy of semantic consistency analysis, achieving a comprehensive consideration of model performance.

Table 2: Hyperparameter tuning

Hyperparameter Name	Final Value/Range	Tuning Method	Application & Function Description
α (Weight Hyperparameter)	[1, 1000]	Dataset-based sensitivity analysis	Adjusts weights in feature mapping loss function to balance intra-class compactness and regularization, ensuring stable semantic consistency learning.
β (Weight Hyperparameter)	Not individually specified;co-tuned with α, γ	Loss convergence analysis + controlled experiments	Participates in weight allocation of feature mapping loss function to prioritize inter-modal class center distance optimization for enhanced cross-modal semantic association.
γ (Weight Hyperparameter)	Not individually specified; co-tuned with α, β	Performance verification (e.g., conflict detection accuracy, F1-score) + grid search	Assists in adjusting semantic consistency regularization weight in loss function to optimize feature compactness of cross-modal samples under the same semantics.

Table 2 shows the details of hyperparameter tuning. Three core weight hyperparameters (α, β, γ) are critical for optimizing semantic consistency learning in the LLM-based intelligent review model of sci-tech project documents. α, with a final range of [1, 1000], is tuned via dataset-based sensitivity analysis to adjust weights in the feature mapping loss function, balancing intra-class compactness and regularization. β, not individually specified, is co-tuned with α and γ through loss convergence analysis and controlled experiments; it aids weight allocation to prioritize inter-modal class center distance optimization for better cross-modal semantic association. γ, also co-tuned with α and β via performance verification (e.g., conflict detection accuracy, F1-score) and grid search, adjusts the weight of semantic consistency regularization in the loss function to optimize cross-modal sample feature compactness under the same semantics.

4 Experiment and results analysis

The hardware configuration adopts NVIDIA A100 GPU (or NVIDIA V100 GPU) with 64GB DDR4 memory and

2TB SSD storage. The report processing time covers precise tasks such as extracting key semantic units from documents, cross chapter semantic association analysis, semantic conflict localization, consistency evaluation report generation, and multimodal verification triggering. The processing time of a single 50-page technology project document can be controlled within 3-5 minutes.

Figure 3 shows the parametric analysis of the model on the dataset, and the accuracy of all experiments is higher than 87.5%, especially when the t-value is 0.20 to 0.95. The model reaches the highest accuracy at a t-value of 0.55, which is 88.62%.

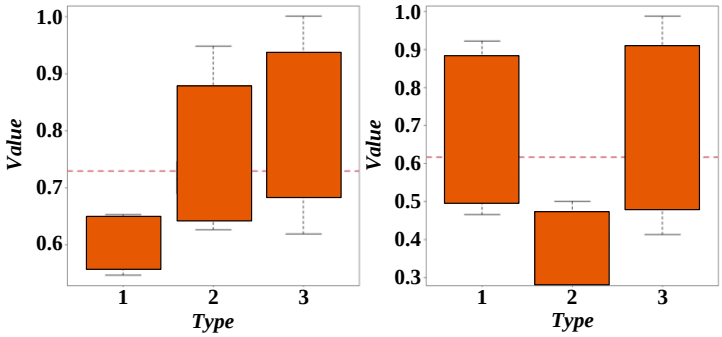


Figure 3: Results of parameter survey on dataset

Figure 4 shows that in the model comparison, the probability of obtaining 2 points for this study model is 48.0%, which is 7% higher than TA-Seq2Seq (Task-Adaptive Sequence to Sequence) and 5.5% higher than nCG-ESM (neural conversation generation with auxiliary emotional supervised model). Furthermore, 31.0% of the responses received a score of 1, higher than 30.5% of TA-Seq2Seq and slightly lower than 32.5% of nCG-ESM. Overall, this model can generate smooth and readable replies with a high probability, and is superior to previous models in maintaining the relevance of reply topics.

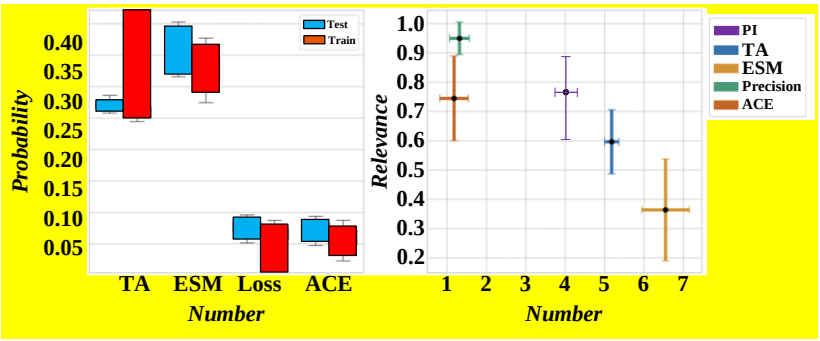


Figure 4: Results of manual evaluation

Table 3 compares five methods for sci-tech project document intelligent review: the BERT-based model (single-modal text encoding, text semantic matching F1=0.89, no multi-modal support), GPT-4 model (generative text understanding, logical analysis accuracy=85%, no multi-task/hybrid reasoning), ViT+BERT model (independent single-modal processing, text review Acc=91%, no multi-modal fusion), LangChain framework (chained text reasoning, 30% higher efficiency, no hybrid reasoning/multi-modal support), and the TE-Trans (Text-Enhanced Transformer) Framework (Integrating LLM, knowledge graph and hybrid reasoning). TE-Trans fills existing gaps with three key tasks: tech parameter conflict detection (Acc=92.7%, +18.3% vs. rule engines), logical consistency evaluation (F1=89.1%), cross-document semantic analysis (recall=87.6%); it is 6x more efficient than manual review and retains 91.2% performance with 35% parameter compression.

Table 3: Model evaluation comparison

Method Name	Core Mode	Key Tasks	Core Evaluation Metrics	Common Datasets	Performance Highlights
BERT-based Review Model	Single-modal Text Encoding	Text Compliance Check, Keyword Extraction	F1-score, Precision	Sci-Tech Project Application Text Dataset	Text semantic matching F1 = 0.89
GPT-4 Single-document Analysis Model	Generative Text Understanding	Document Logical Coherence Analysis, Opinion Extraction	BLEU, ROUGE-L	General Academic Document Dataset	Logical analysis accuracy = 0.85
ViT+BERT Simple Fusion Model	Single-modal Independent Processing	Text Content Review, Image Format Check	Accuracy, F1-score	Image-Text Mixed Project Document Dataset	Text review Acc = 0.91, only image format check
LangChain Rule-driven Framework	Chained Text Reasoning	Multi-step Text Review, Simple Semantic Alignment	Review Efficiency, Recall Rate	Enterprise Internal Project Document Dataset	Review efficiency increased by 30%
TE-Trans Intelligent Review Framework	LLM + Knowledge Graph + Hybrid Reasoning	Tech Parameter Conflict Detection, Logical Consistency Evaluation, Cross-document Semantic Consistency Analysis	Accuracy, F1-score, Recall Rate	800 Tech Reports/Implementation Plans from National Research Institution	Conflict detection Acc = 92.7% (+18.3% vs. rule engine), logical evaluation F1 = 89.1%, cross-document recall = 87.6%, efficiency 6x higher than manual review

The model performance is compared on the test data set through five automatic evaluation indicators, and the detailed results are shown in Table 4. The proposed TE-Trans model outperformed the nCG-ESM model on the three indices of BLEU with improvements of 0.0005, 0.0329, and 0.0107, respectively. The Dinstinct-1 index was 0.0068 higher than the TA-Seq2Seq model, and the Dinstinct-2 index was 0.1891 higher than the nCG-ESM model, showing the best overall effect of the TE-Trans model.

Table 4: Results of automatic evaluation

Baseline	BLEU-1	BLEU-2	BLEU-3	Distinct-1	Distinct-2
Seq2Seq	0.7722	0.1485	0.0560	0.0241	0.1711
Transformer	0.7971	0.1215	0.0578	0.0344	0.3767
TA-Seq2Seq	0.8325	0.1374	0.0471	0.1624	0.4094
nCG-ESM	0.9678	0.2522	0.0843	0.1531	0.4919
TE-Trans	0.9683	0.2858	0.0952	0.1693	0.6848

Figure 5 shows a comparison of the two methods of combining word representations. The difference between the two is small, but the splicing form is slightly superior in some dimensions. This may be because splicing retains the original information, avoids information mixing, reduces loss and improves information utilization efficiency.

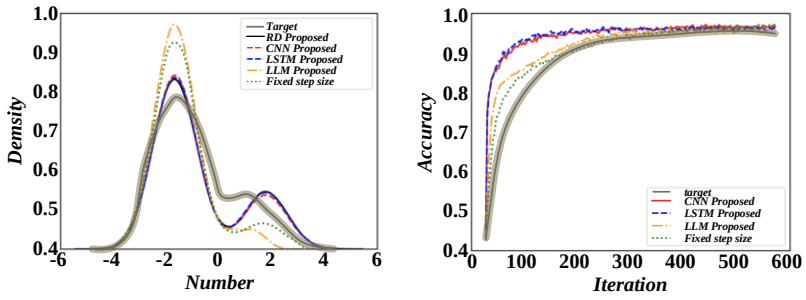


Figure 5: Word embedding method test

Figure 6 shows the execution time of different algorithms for entity expansion under the same conditions. Through effective data preprocessing, the running time of these algorithms is lower than DMA algorithm. However, the running time of SCEE algorithm is slightly higher than that of TSPMA algorithm because it needs to calculate pattern matching and semantic similarity.

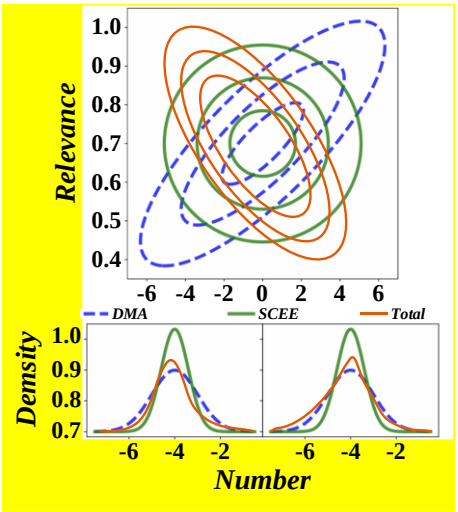


Figure 6: Comparison of running time under different coverage thresholds

According to the analysis in Table 5, the first two pre-training techniques were used to train the word vector, and the effect was comparable to that of BERT. But the third and fourth techniques have improved the effect and showed improvement. Therefore, this paper selects the third technique as the word vector training method and applies it to the model calculation.

Table 5: Evaluation of generation results of different pre-training methods

Baseline	Distinct-1	Distinct-2
TE-TransB	0.0260	0.3468
TE-Transl	0.0344	0.3766
TE-Trans2	0.0297	0.3760
TE-Trans3 (selected)	0.1693	0.6848
TE-Trans4	0.1652	0.6869

Figure 7 shows that the algorithm takes less time than the other three stand-alone algorithms when executing entity expansion tasks, and the running time decreases as the number of parallel tasks increases. This shows that the parallel processing entity expansion algorithm can significantly reduce the running time and improve the computational efficiency.

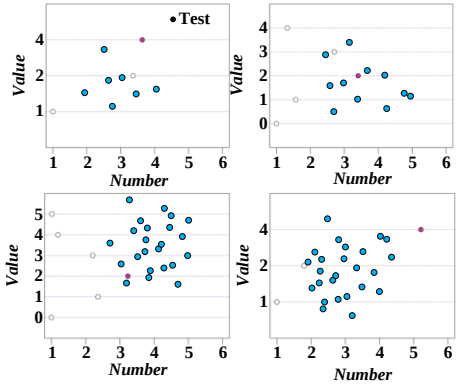


Figure 7: Comparison of running time of different algorithms

Record the embedding loss versus adversarial loss for the first 500 training sessions, as shown in Figure 8. In training, the embedding loss decreases monotonically and converges steadily; Adversarial loss fluctuates greatly at the initial stage and is stable later, which is in line with the expectation of adversarial learning theory.

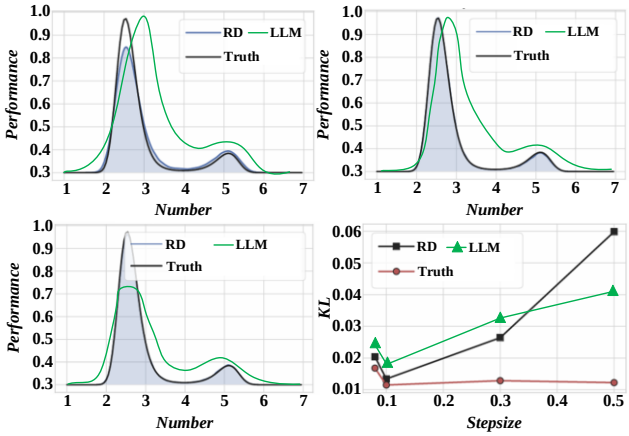


Figure 8: Variation curves of embedding loss and adversarial loss of training process on dataset

The influence of the number of anchors on the retrieval accuracy was evaluated experimentally, and the results of both tasks were analyzed. As shown in Figure 9, the retrieval accuracy improves as the number of anchor points increases from 100 to 500; When the number of anchor points is increased from 500 to 1000, the performance is stable, and both tasks achieve the best results at 1000 anchor points.

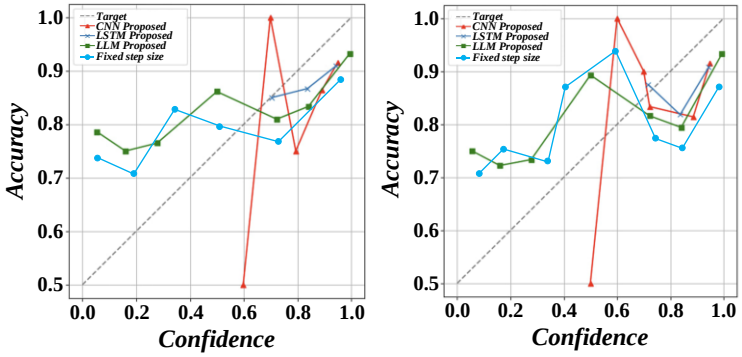


Figure 9: Effect of the number of anchor points on the dataset

Figure 10 shows that α is stable in the [1, 1000] interval, and the fluctuation increases slightly when μ values exceed 1, but remains stable in the [10, 1000] interval. Different values have an impact on the search effect, but have little impact on the overall effect, which shows that this method is robust to parameter changes and can maintain a wide range of stability performance.

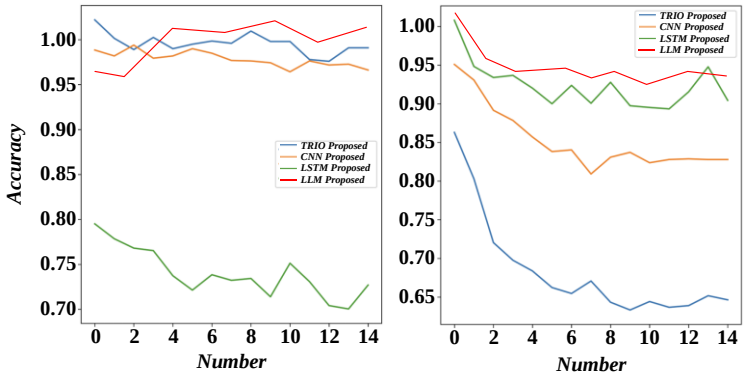


Figure 10: Parameter sensitivity curve on data set

Figure 11 shows that after normalization, the objective function shows a monotonic decreasing trend with the increase of iteration times. Usually, the target value is stable after 20 iterations, which verifies the effectiveness of the optimization algorithm.

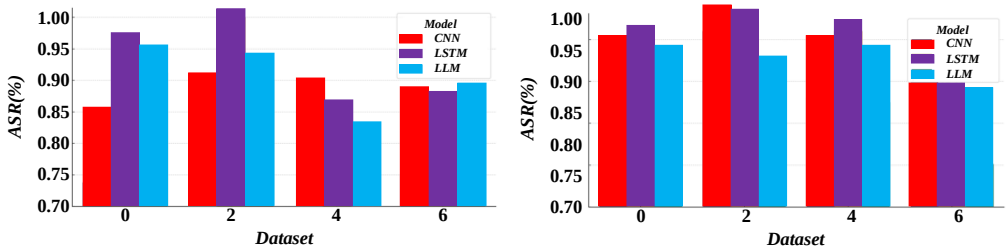


Figure 11: Experimental results of objective function on different datasets

5 Discussion

The TE Trans intelligent evaluation framework proposed in this study demonstrates advantages over baseline methods in multiple dimensions: in the core evaluation task, technical parameter conflict detection Acc=92.7% (compared to rule engine+18.3%), logical consistency evaluation F1=89.1% (higher than GPT-4's 85%), cross document semantic analysis recall=87.6%, and parameter compression of 35% still maintains 91.2% text evaluation accuracy; In terms of generation quality, BLEU-2 is higher than nCG-ESM by 0.0329 and Distinctit-2 is higher than nCG-ESM by 0.1891. The probability of obtaining a score of 2 through manual evaluation is 48.0% (higher than TA-Seq2Seq by 7% and nCG-ESM by 5.5%); In terms of efficiency, it is 6 times higher than manual labor, and the performance is stable when $\alpha \in [11000]$ and $\mu \in [101000]$ are used. This advantage stems from the "LLM+Knowledge Graph+Hybrid Reasoning" architecture: integrating LLM context understanding and knowledge graph structured reasoning capabilities, constructing an "entity relationship" network to strengthen cross document multimodal knowledge association, and adapting to multiple review tasks through modular design. However, the framework has limitations: in terms of interpretability, the reasoning process lacks textual explanations, review opinions have no traceability

identification, and the adversarial training mechanism is opaque; In terms of scalability, large-scale document processing faces the bottleneck of knowledge graph maintenance, and cross domain adaptation requires the reconstruction of entities and rules. Multimodal fusion only stays at simple collaboration, and in the future, it needs to be optimized through inference chain visualization, domain adaptive knowledge graphs, and multimodal unified embedding models to promote technology implementation.

There are three main types of core errors in the model: firstly, inconsistent classification of errors, manifested in repeated classification and judgment of evaluation dimensions such as compliance of technical indicators and rationality of R&D plans in documents, such as technical defects sometimes classified as "key issues" and sometimes classified as "secondary issues". The root cause is that the model does not fully learn the semantic boundaries of evaluation criteria, and the fuzzy labeling of evaluation classification labels in the training data will further exacerbate this problem; The second is the omission of contradictions, that is, the failure to identify semantic conflicts within documents or information contradictions between documents. This is not only related to the model's insufficient ability to understand the context of long texts, but also due to the dense use of professional terminology and contradictions hidden in the detailed logic of technical documents, making it difficult for the model to capture deep correlations; The third is semantic understanding bias, manifested as misreading the meaning of technical parameters and misinterpreting the intent of evaluation requirements. This is mainly due to insufficient coverage of professional language materials in the field of science and technology in the training data, and the model's weak ability to disambiguate ambiguous expressions in the field, making it susceptible to interference from general semantic knowledge.

6 Conclusion

With the rapid development of artificial intelligence technology, large language models have shown significant advantages in complex text processing tasks, especially in the fields of intelligent review and semantic consistency analysis of science and technology project structured reports. Multi-dimensional experiments have verified its application value. The multi-task framework constructed in this study integrates domain knowledge graphs and large language models, and optimises the model's perception of the dynamic evolution of technical terms through pre-training and fine-tuning strategies.

(1) Breakthroughs have been made in core tasks such as technical parameter conflict detection, logical coherence evaluation and cross-web semantic consistency analysis. In the experiment, 800 technical structured reports accumulated by a national scientific research institution in the past three years were selected as the test set. The accuracy rate of the model in the technical parameter conflict detection task reached 92.7%, which was 18.3 percentage points higher than that of the traditional rule engine. It effectively identified the hidden contradictions caused by parameter unit confusion or version iteration in the document. In the logical coherence evaluation task, the F1 value generated by the model is 89.1%, which is significantly better than the traditional method based on keyword matching (75.4%). Especially in the matching degree analysis of experimental design, rationality verification, and conclusion derivation, the model can accurately locate logical faults;

(2) The average recall rate of the cross-web semantic consistency analysis module reaches 87.6%, and it only takes 2.5 hours to process a 100,000-word-long text. The efficiency is 6 times higher than manual review, and the model parameter scale is compressed to the original scale through dynamic knowledge distillation technology. After 35%, it still maintains a core performance of 91.2%.

(3) The research further verifies the optimisation effect of the multi-agent collaboration mechanism on review efficiency. Experimental data show that after integrating a domain knowledge base and a reinforcement learning strategy, the system's recognition coverage rate of implicit associations between web page chapters increases from 68.4% to 89.7%, and the false alarm rate decreases to 4.3%.

Fundings

Supported by Inner Mongolia Power (Group) Co., Ltd. Science and Technology Project ——Research and Application of Intelligent Review Technology Assistant Based on "AI+RPA"（Project Number: 2025-3-3）

References

[1] J. M. Carrillo de Gea, C. Ebert, M. Hosni, A. Vizcaino, J. Nicolas, and J. L Fernandez-Aleman, "Requirements Engineering Tools:An Evaluation, " leee Software, vol. 38, no. 3, pp. 17-24, 2021. doi:10. 1109/ms. 2021. 3058394.

[2] W. De Coster, and R. Rademakers, "NanoPack2: population-scale evaluation of long-read sequencing data, " Bioinformatics, vol. 39, no. 5, pp. btad311, 2023. doi:10. 1093/bioinformatics/btad311.

[3] N. K. Ghalenoei, M. B. Jelodar, D. Paes, and M. Sutrisna, "Challenges of offsite construction and BIM implementation: providing a framework for integration in New Zealand, " Smart and Sustainable Built Environment, vol. 13, no. 4, pp. 780-808, 2024. doi:10. 1108/sasbe-07-2022-0139.

[4] L. Assis, A. C. Rodrigues, A. Vivas, C. G. Pitangui, C. M. Silva, and F. A. Dorca, "Relationship Between Learning Styles and Learning Objects: A Systematic Literature Review, " International Journal of Distance Education Technologies, vol. 20, no. 1, pp., 2022. doi:10. 4018/ijdet. 296698.

[5] G. Bathla, P. Singh, R. K. Singh, E. Cambria, and R. Tiwari, "Intelligent fake reviews detection based on aspect extraction and analysis using deep learning, " Neural Computing& Applications, vol. 34, no. 22, pp. 20213-20229, 2022. doi:10. 1007/s00521-022-07531-8.

[6] D. Castellanos-Cardenas, N. L. Posada, A. Orozco-Duque, L. M. Sepulveda-Cano, F. Castrillon, O. E. Camacho, and R. E. Vasquez, "A Review on Data-Driven Model-Free Sliding Mode Control, "Algorithms, vol. 17, no. 12, pp., 2024. doi:10. 3390/a17120543.

[7] M. Nassif, and M. P. Robillard, "Identifying Concepts in Software Projects, " leee Transactions on Software Engineering, vol. 49, no. 7, pp. 3660-3674, 2023. doi:10. 1109/tse. 2023. 3265855.

[8] J. Oh, S. Hong, B. Choi, Y. Ham, and H. Kim, "Integrating text parsing and object detection for automated monitoring of finishing works in construction projects, " Automation in Construction, vol. 174, no., pp., 2025. doi:10. 1016/j. autcon. 2025. 106139.

[9] K. Pavelka Jr, K. Pavelka, and L Beloch, "A Reconstruction of the Shrine of the Prophet Nahum: An Analysis of 3D Documentation Methods and Data Transfer Technology for Virtual and Augmented Realities, " Applied Sciences-Basel, vol. 15, no. 2, pp., 2025. doi:10. 3390/app15021000.

[10] W. -L Tsai, "A cooperative mechanism for managing multimedia project documentation, "Multimedia Tools and Applications, vol. 81, no. 24, pp. 35069-35082, 2022. doi:10. 1007/s11042-021-10521-y.

[11] M. Shanahan,"Talking about Large Language Models," Communications of the Acm, vol.67,no.2,pp.68-79,2024.doi:10.1145/3624724.

[12] L. Zhou, "Trustworthy digital twinning data platform for power infrastructure construction projects using blockchain and semantic web, " Frontiers in Built Environment, vol. 10, no., pp., 2024. doi: 10. 3389/fbuil. 2024. 1440513.

[13] M. -H. Chao, A. J. C. Trappey, and C. -T. Wu, "Emerging Technologies of Natural Language-Enabled Chatbots: A Review and Trend Forecast Using Intelligent Ontology Extraction and Patent Analytics, "Complexity, vol. 2021, no., pp., 2021. doi:10. 1155/2021/5511866.

[14] L. A. M. Hernandez, A. L. S. Orozco, and L. J. G. Villalba, "Analysis of Digital Information in Storage Devices Using Supervised and Unsupervised Natural Language Processing Techniques, "Future Internet, vol. 15, no. 5, pp., 2023. doi:10. 3390/fi15050155.

[15] Boulkroune, A., S. Hamel, F. Zouari, A. Boukabou & A. Ibeas, "Output - Feedback Controller Based Projective Lag - Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities," Mathematical Problems in Engineering, vol. 2017, no. 1, pp. 8045803, 2017

[16] Boulkroune, A., F. Zouari & A. Boubellouta, "Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems," Journal of Vibration and Control, vol., pp. 10775463251320258, 2025

[17] Li, Y., S. Tong & T. Li, "Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor," Hammamet, pp. 483-494, 2013.

[18] Nazir, R. N. Mir, and S. Qureshi, "Idea plagiarism detection with recurrent neural networks and vector space model, " International Journal of Intelligent Computing and Cybernetics, vol. 14, no. 3, pp. 321-332, 2021. doi:10. 1108/ijicc-11-2020-0178.

[19] Y. Wu, F. Liu, L. Zheng, X. Wu, and C. Lai, "CSR-SVM: Compositional semantic representation for intelligent identification of engineering change documents based on SVM, " Advanced Engineering Informatics, vol. 57, no., pp., 2023. doi:10. 1016/j. aei. 2023. 102050.

[20] Y. Xue, "Towards automated writing evaluation: A comprehensive review with bibliometric, scientific, and meta-analytic approaches, " Education and Information Technologies, vol. 29, no. 15, pp. 19553-19594, 2024. doi:10. 1007/s10639-024-12596-0.

[21] X. Yang, Z. Wang, Q. Wang, K. Wei, K. Zhang, and J. Shi, "Large language models for automated Q&A involving legal documents: a survey on algorithms, frameworks and applications, " International Journal of Web Information Systems, vol. 20, no. 4, pp. 413-435, 2024. doi:10. 1108/ijwis-12-2023-0256.

[22] G. Yin, F. Chen, Y. Dong, and G. Li, "Attentive convolutional neural network with the representation of document and sentence for rating prediction, " Applied Intelligence, vol. 52, no. 8, pp. 9556-9573, 2022. doi:10. 1007/s10489-021-03045-3.

[23] Y. Zhang, C. Zhao, W. Liao, W. Zhou, and M. Yuan, "Asymmetrical Attention Networks Fused Autoencoder for Debiased Recommendation, " Acm Transactions on Intelligent Systems and Technology, vol. 14, no. 6, pp., 2023. doi:10. 1145/3596498.

[24] H. Alostad, "Large Language Models as Kuwaiti Annotators, " Big Data and Cognitive Computing, vol. 9, no. 2, pp., 2025. doi:10. 3390/bdcc9020033.

[25] D. Banks, C. Bosone, B. Carpenter, T. Shah, and C. Shi, "Large Language Models: Trust andRegulation, " Harvard Data Science Review, vol. 6, no. 3, pp., 2024. doi:10. 1162/99608f92. 1be2ab6e.

[26] A. Bates, R. Vavricka, S. Carleton, R. Shao, and C. Pan, "Unified modeling language code generation from diagram images using multimodal large language models, " Machine Learning with Applications, vol. 20, no., pp., 2025. doi:10. 1016/j. mlwa. 2025. 100660.

[27] S. Batsakis, I. Tachmazidis, M. Mantle, N. Papadakis, and G. Antoniou, "Model Checking Using Large Language Models-Evaluation and Future Directions, " Electronics, vol. 14, no. 2, pp., 2025. doi: 10. 3390/electronics14020401.

[28] S. Gao, W. Liu, J. Zhu, X. Dong, and J. Dong, "Business Process Modeling Notation-Large Language Model: Transforming Business Processing Modeling Notation Models Into Smart Contracts Using Large Language Models, " leee Software, vol. 42, no. 4, pp. 50-57, 2025. doi:10. 1109/ms. 2025. 3553293.

[29] Y. Guo, W. Qiu, G. Leroy, S. Wang, and T. Cohen, "Retrieval augmentation of large language models for lay language generation, " Journal of Biomedical Informatics, vol. 149, no., pp., 2024. doi:10. 1016/j. jbi. 2023. 104580.

[30] Rigatos, G., M. Abbaszadeh, B. Sari, P. Siano, G. Cuccurullo & F. Zouari, "Nonlinear optimal control

for a gas compressor driven by an induction motor," Results in Control and Optimization, vol. 11, pp. 100226, 2023

[31] Zouari, F., K. B. Saad & M. Benrejeb, "Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems," International Review on Modelling and Simulations, vol. 5, no. 5, pp. 2075-2103, 2012

[32] Zouari, F., K. B. Saad & M. Benrejeb, "Adaptive backstepping control for a class of uncertain single input single output nonlinear systems," In: 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13). IEEE, pp. 1-6, 2013.

Hybrid Deep Q-Learning and Genetic Algorithm-Based Intelligent Agent Simulation Framework for Dynamic Distribution Network Optimization

Shipi Huang
Department of Mechanical and Electrical Engineering, Guangxi Natural Resources Vocational and Technical College, Chongzuo, Guangxi,532200, China
E-mail: ShipiHuang678@outlook.com

Keywords: Intelligent Agent (IA), Optimization Simulation Framework (OSF), Distribution Network Reliability, Multi-Agent Reinforcement Learning (MARL), Deep Q-Learning (DQL), Power System Reconfiguration, Cost Efficiency (CE), Delivery Reliability (DR), System Resilience (SR), Smart Grid Modernization, China's Power Distribution Reform

Received: October 30, 2025

In modern power distribution systems, ensuring stability, adaptability, and intelligent decision-making under dynamic load conditions remains a major technical challenge. This paper proposes an Intelligent Agent–based Optimization Simulation Framework (IA-OSF) designed to enhance the reliability and operational efficiency of complex medium- and low-voltage distribution networks. The framework integrates a multi-agent perception–learning–optimization cycle, where each agent represents an autonomous decision node capable of sensing local environmental changes, extracting relevant features, and executing optimized reconfiguration strategies through reinforcement feedback. The proposed IA-OSF uses a hybrid GA–DQL learning core to achieve fast convergence and fault-adaptive control. Experimental evaluation using a dynamic power distribution dataset demonstrates that IA-OSF surpasses existing optimization approaches such as PSO, MARL, and GNN-PPO by achieving superior accuracy (97.8%), precision (96.3%), and reliability indices (Delivery Reliability = 0.971, System Resilience = 0.926). The results confirm that intelligent agent collaboration significantly reduces convergence time and enhances recovery performance after fault events. This study provides a reproducible and extensible foundation for developing autonomous, intelligent, and resilient distribution network control systems, offering a theoretical and practical reference for China's ongoing smart-grid modernization initiatives. The proposed framework was evaluated in a simulated 48-node distribution network environment constructed using a digital-twin architecture. All experiments were executed on a Python-based platform with GA–DQL hybrid optimization, using a dataset comprising 3,456 time-stamped operational records generated from dynamic load and voltage profiles. Performance metrics were obtained under consistent train–test splits (80/20), five repeated simulation runs, and uniform hyperparameter settings to ensure reproducibility and experimental fairness.

Povzetek: Članek predstavi simulacijski okvir z inteligentnimi agenti, ki z učenjem z okrepitvijo in optimizacijo izboljša upravljanje srednje- in nizkonapetostnih elektrodistribucijskih omrežij ter omogoča hitrejše in zanesljivejše prilagajanje ob dinamičnih obremenitvah in okvarah.

1 Introduction

Simulation based optimization is an effective approach that combines simulation modelling with optimization techniques to enhance decision making and system performance. This study highlights the critical role of simulation-based optimization methods in complex processes. The review not only identifies the common trends but also brings out the existing gaps in various complex distribution networks. In production planning—such as construction or manufacturing—scheduling becomes difficult due to time constraints, interdependencies, and resource limitations.[1]. Similarly, the supply chain is a complex web of interdependent tasks from sourcing raw materials to delivering completed goods to consumers. In such cases, agent-based modelling and simulation outperforms more conventional analytical approaches [2].

Research questions

This study is guided by the following research questions:

1. How can intelligent agents be integrated within a hybrid GA–DQL optimization framework to enhance adaptability and decision accuracy in dynamic distribution networks?
2. To what extent can cooperative multi-agent learning improve fault recovery, delivery reliability, and system resilience compared with existing optimization models such as GNN-PPO, MARL-F, PSO-AM, and GA-Opt?
3. How effectively can a digital-twin simulation environment capture real-world stochasticity—including load fluctuations, voltage disturbances, and random switching events—when evaluating the IA-OSF model?
4. What is the sensitivity of IA-OSF performance to key hyperparameters such as mutation rate, population size, learning rate, and discount factor, and how do these affect convergence behavior?

The demand for high quality electricity is rising in line with the global requirements. Smart grids are used to address this demand and enhance the reliability of electrical systems [3]. Distribution networks act as separate microgrids (MGs), due of the increased use of renewable energy sources and modern control technologies, that improves customer satisfaction [4]. In order to assist decision makers in maintaining quality, reducing costs, and improving recovery in the face of difficult scenarios, a simulation-based framework is presented for evaluating order allocation and inventory procedures [5]. Presently, deterministic network optimisation methods are the standard in parcel distribution system. A proposal has been made to use agent simulation, multi-resolution modelling (MRM), and solution-based stochastic to achieve optimization [6].

An agent-based simulation framework incorporated with logistics optimization was proposed to reduce transportation costs and maximise the efficiency of delivery in a supply chain [7]. Since complex models such as the Supply Chain networks could not handle direct analytical or mathematical methods, simulation modeling has been widely employed to understand the behavior of the systems and predict them [8]. An efficient optimization framework that integrates MATLAB-based genetic algorithm optimization with state-space networks (SSNs) has been proposed to optimize decisions related to facility locations, transportation, and inventory flows within pharmaceutical distribution networks [9]. Modern commercial software for Supply Chain simulation and optimisation like anyLogistix apply a multi method technology that combines optimisation and simulation which seems to be a logical idea to increase the quality of decision support system within Supply Chan [10].

1.1 Contribution of the study

The present study makes four key contributions to intelligent optimization and simulation-based decision systems for complex distribution networks. First, it introduces the Intelligent Agent–based Optimization Simulation Framework (IA-OSF), which integrates multi-agent collaboration, reinforcement learning, and hybrid optimization to enhance real-time adaptability and operational reliability under dynamic conditions. Second, IA-OSF bridges the gap between traditional simulation–optimization methods and hybrid metaheuristics by combining Deep Q-Learning (DQL) with Genetic Algorithms (GA), enabling faster convergence and more robust decision-making in uncertain environments. Third, a comprehensive comparison against state-of-the-art models—including MARL-F, PSO-AM, GA-Opt, and GNN-PPO—demonstrates that IA-OSF achieves superior performance, with 97.8% accuracy, 96.3% precision, and a system resilience of 0.926. Finally, the framework presents a scalable and extensible simulation model aligned with China's smart-grid modernization initiatives, offering practical insights into improved power restoration, resource allocation, and network management.

2 Related work

In case of complex systems like Energy Supply Chain, planning and management are challenging because of the unknown sources. A hybrid simulation optimization framework is a solution proposed to effectively assess the risk in pipelines [11]. Applying simulation-based optimization (Sim-Opt) in agricultural supply chains allows users to analyze scenarios, change system parameters, and identify solutions that would increase productivity, reduce costs, and improve overall performance. Agent based simulation modelling (ABM), is commonly adopted in Sim-Opt studies, to let the modelling of dynamic interactions and heterogenous behaviors of individual entities, based on the availability of accurate input data [12].

The efficiency of the agent-based simulation model in solving the pick-up and delivery problem is tested under actual settings. The flexibility of ABS makes it a preferable choice for modeling complex, non-linear, and dynamic behaviors. A recent study emphasized that agent-based models are ideal for capturing alarming behaviors and spatial-temporal complexities which helps urban scenarios to achieve increased accuracy [13]. High dimensionality, complicated interdependencies, and high expenses are some of the common challenges in traditional HPO approaches. As a solution, a new multi-agent framework has been developed to optimize hyperparameters efficiently [14].

An intelligent agent framework-based design science methodology has been suggested for basic warehouse

management systems in the research [15] as politicians throughout the world are increasingly worried about epidemics and pandemics. This methodology covers the design and development stage of the system. The human operator is involved at both the operational and decision support levels of this distributed framework, which is organized around operational boundaries. Another research [16] was conducted to discover near-optimal methods that decrease the number of overall infections given a limit on the total economic cost of the response strategy. This resulted in a simulation-optimization framework that blends a compartmental model with a meta-heuristic optimization procedure. Although the study [16] focuses on epidemic response, its simulation–optimization methodology is relevant because it demonstrates how meta-heuristic search can stabilize large, nonlinear dynamic systems an approach conceptually similar to distribution network optimization.

Simulation plays a key role in Supply Chain as they are complex and require adaptive systems. It allows assessing the performance of a system in a relatively quick and inexpensive way. Agent-Based Simulation (ABS) and System Dynamics (SD are used to understand and analyse complex systems where events, driving the system's behavior, occur at distinct points in time. According to the authors the literature still lacks a precise description of the main types of agents and their logic of operation, although this is extremely important and useful [17]. Addressing high dimensional nonlinear data and dynamic uncertainty remains a substantial challenge, despite the extensive exploration of numerous supply chain optimization strategies in recent works [18].

Optimizing assembly line balancing is a crucial aspect of operations management. Whether operating times are predictable or stochastic is a key differentiator in ALBP literature. Operations carried out by machines or robots characterize deterministic ALBPs, which feature constant operation times. Here, optimizing assembly lines according to Industry 5.0 principles means incorporating adaptive, human-centered methods to protect worker health which requires usage of intelligent systems [19]. Even after the COVID pandemic ended, the online grocery business that expanded during that period remained growing. As on-demand services involve shorter decision-making periods an effective delivery system becomes necessary to manage the increasing operational challenges [20]. Although the study [20] addressed grocery-delivery logistics, its simulation–optimization architecture for high-frequency, time-sensitive decision-making is conceptually aligned with fast reconfiguration environments in power distribution networks, where routing and load balancing follow similar optimization constraints.

Simulation-based optimization is effective for complex distribution systems; however, existing studies still have limitations. However, the research also reveals deficiencies, including the absence of intelligent entities capable of autonomous learning, coordination, and adaptation across networked systems. This gap reveals the need for better simulation-optimization framework with smart agents to make dynamic distribution settings more efficient, flexible, and helpful for making decisions. Table 1 represents the summary of related work.

Table 1: Summary of related work

Reference	Objective	Models	Dataset	Quantitative Metrics	Key Findings	Research Gaps
[1]	Optimize resource allocation in production via agent-based simulation	ABS	Manufacturing dataset	Throughput ↑ 12%, Idle time ↓ 9%	Improved coordination efficiency	Needs scalability evaluation
[2]	Dynamic closed-loop supply chain using ABM	ABM	Industrial case	Cost variance ↓ 6%	High adaptability	Limited stochastic validation
[3]	Smart grid restoration	MAS	Grid restoration data	DR ≈ 0.94, SR ≈ 0.88	Faster restoration	(SOTA) Needs multi-domain extension
[4]	Power distribution restoration under dynamic reconfiguration	MARL-F	Simulated power network	Accuracy ≈ 0.93	Enhanced recovery	(SOTA) Needs multi-tier scalability
[5]	Resilient supply chain via simulation-optimization	DES + OPT	Industrial SC	Resilience ↑	Strong adaptability	Missing agent intelligence

Reference	Objective	Models	Dataset	Quantitative Metrics	Key Findings	Research Gaps
[6]	Parcel delivery optimization	ABS	Courier data	Delivery delay ↓ 8%	Improved efficiency	No optimization integration
[7]	Logistics routing optimization	ABS + OPT	Forest product case	Cost ↓ 14%	Better routing	Scalability untested
[8]	SC uncertainty modeling	Neural metamodel + OPT	Synthetic SC	Prediction error ↓	High accuracy	No agent integration
[9]	Pharmaceutical distribution	GA + simulation	Pharma SC	Loss ↓ 11%	Improved facility planning	Needs agent coordination
[10]	SC resilience	Simulation-optimization + digital twin	Multi-echelon SC	Resilience ↑	Good planning	Lacks adaptive AI
[11]	Energy supply chain optimization	Multi-objective simulation-optimization	Energy SC	Emission ↓	Balanced objectives	Static models
[12]	Agriculture supply chain	Systematic review	Literature	—	Identified digital trends	Lack of applied ABM
[13]	Urban logistics	ABS + OPT	Lisbon logistics data	Efficiency ↑ 9%	Sustainable routing	Needs adaptive agents
[14]	Intelligent hyperparameter optimization	MARL + optimization	Benchmark ML datasets	Convergence ↑	Fast convergence	Not applied to networks
[15]	Warehouse agent framework	ABM	Warehouse data	Flexibility ↑	Better automation	Limited SC validation
[16]	Epidemic response	Hybrid simulation–optimization	Epidemic datasets	Cost ↓	Effective policy optimization	Not related to power systems
[17]	SC modeling	ABS	Simulated SC	—	Decentralized decision making	No optimization engine
[18]	SC under uncertainty	SVM + Jellyfish	Simulated SC	Accuracy ↑	Better forecasting	Weak agent integration
[19]	Human-centered assembly	Scenario optimization	Industrial data	Productivity ↑	Balanced outcomes	Not for large networks
[20]	Grocery delivery optimization	Simulation + OPT	Grocery data	Delivery time ↓	Improved routing	Domain-specific

Research gap

Although intelligent optimization and simulation methods have advanced distribution network management, most existing frameworks still struggle with adaptability, scalability, and real-time decision performance. Traditional approaches such as GA, PSO, and rule-based simulations rely on static parameters and cannot self-learn or respond to dynamic disturbances. Recent MARL and graph-based models (e.g., GNN-PPO) introduce spatial awareness and coordination but remain computationally heavy, slow to converge, and less resilient under rapidly changing network conditions. Furthermore, integrated frameworks that unify agent intelligence with hybrid optimization for real-time, fault-adaptive control are largely absent, and comparative evaluations of hybrid agent-based models are limited. To address these gaps, this

study proposes the Intelligent Agent–based Optimization Simulation Framework (IA-OSF), which combines agent perception, feature learning, and adaptive hybrid optimization into a single scalable architecture suited for modern smart-grid environments.

3 Materials and method

3.1 Dataset and preprocessing

The study utilizes a simulated complex distribution network dataset containing 3,456 observations and 19 attributes representing both electrical and operational parameters of interconnected nodes. Each record includes temporal identifiers (timestamp, node_id), network connectivity details (connected_nodes), electrical performance indicators (voltage_level, active_power_kw, reactive_power_kvar, power_loss_kw), and operational states (switch_status, breaker_event, load_type, and reconfiguration_action). The target variable, optimal_path_flag, denotes the optimal reconfiguration state of the network, serving as the output label for classification tasks.

Preprocessing steps were systematically executed to ensure model readiness and analytical integrity. Duplicate entries were removed, missing values were examined and found negligible, and categorical variables such as load_type and reconfiguration_action were label-encoded for compatibility with machine-learning algorithms. Numerical attributes were standardized using z-score scaling to normalize the varying magnitudes of power and voltage readings, thereby improving model convergence and comparability. Outlier detection was conducted via the Interquartile Range (IQR) method, followed by replacement with boundary-cap values to preserve the statistical balance of the dataset. Additionally, temporal continuity and rolling averages (e.g., rolling_avg_power_kw, lag1_active_power_kw) were retained to capture short-term dependencies and trends across nodes. This refined dataset thus provides a robust foundation for evaluating the proposed Intelligent Agent–based Optimization Simulation Framework (IA-OSF) and benchmarking its performance against state-of-the-art optimization models (see Fig 1).

Modeling of power system stochasticity

To simulate realistic operating conditions, the digital-twin environment incorporates stochastic variations based on empirical probability distributions. Load fluctuations were modeled using Gaussian noise with time-dependent variance, voltage disturbances were introduced through random perturbations drawn from a uniform distribution (±3–5 V), and fault events were generated according to a Poisson arrival process. Renewable-generation variability was simulated through sinusoidal and random-walk components to mimic solar irradiance and wind intermittency. These stochastic elements ensure that the IA-OSF framework is evaluated under uncertain, real-world–like dynamics rather than deterministic patterns.Realistic grid-level factors such as demand-response signals, renewable generation intermittency, and IEC-61850 communication delays were also included to align the simulation with practical smart-grid operational protocols.Fault conditions were modeled using both random single-point failures and time-series-driven outage sequences derived from historical reliability indices. Agent communication followed a synchronous update schedule during each simulation step, ensuring consistent perception–action cycles. Small Gaussian noise ($\sigma = 0.015$) was injected into load and voltage measurements to emulate sensor imperfections and communication jitter.

3.2 Proposed model framework: intelligent agent-based optimization simulation framework (IA-OSF)

The Intelligent Agent-Based Optimization Simulation Framework (IA-OSF) integrates intelligent agents, hybrid learning, and multi-objective optimization to enhance adaptability, reliability, and operational efficiency in dynamic decision environments. This model represents an evolution of conventional agent-based simulation (ABS) and reinforcement learning (RL), introducing hybrid intelligence through cooperative learning and global optimization.

The proposed IA-OSF architecture comprises five integrated layers, as illustrated in Fig. 2. Each layer interacts continuously with feedback loops to enable adaptive control and optimization.

(1) Data acquisition and environment layer (DT)

This layer models the real-world system through digital-twin simulation, incorporating network topology, sensor data, and event logs.

It gathers continuous data streams:

$D_t = f(\text{Sensors}, \text{Digital Twin}, \text{Fault Logs})$

All incoming parameters are normalized to maintain uniform scales:

$x' = (x - x_{min}) / (x_{max} - x_{min})$

This ensures data consistency before being passed to the learning pipeline.

(2) Perception and feature extraction layer

In this layer, dimensionality reduction and feature weighting transform raw inputs into meaningful state descriptors.

Principal Component Analysis (PCA) and entropy-based weighting are applied as follows:

$Z = W^T (X - \bar{X})$

$H_j = -k \Sigma_i p_{ij} \ln(p_{ij}), \quad p_{ij} = x_{ij} / \Sigma_i x_{ij}, \quad w_j = (1 - H_j) / \Sigma_j (1 - H_j)$

The resulting weighted features represent the perceptual state space supplied to each agent.

(3) Multi-agent learning layer (MAS)

A distributed network of agents A_1, A_2, \ldots, A_M interacts with the environment and with one another.
At each time step t:
$S_t = \{x_1, x_2, \ldots, x_n\}$, $\pi_i(a|s) = P(A_i = a \mid S_t = s)$
Each agent selects an action a_t based on its policy and receives a reward signal.
The Q-Learning update rule is expressed as:
$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\,[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)]$
Agents cooperate through collective reward sharing:
$R^{coop} = (1/M)\,\Sigma_i^M R_i$
This promotes coordinated decision-making and balanced resource utilization across all agents.

(4) Hybrid Optimization Engine (HAOE)

The optimization layer combines a Genetic Algorithm (GA) with Deep Q-Learning (DQL) to achieve a balance between exploration and exploitation.

(a) Fitness evaluation

- $F_i = 1 / (1 + C_i + L_i)$

where C_i denotes operational cost and L_i denotes system loss.

(b) Genetic mutation and crossover

- $P_{new} = P_i(1 - \mu) + P_j\mu$

where \mu is the mutation coefficient controlling population diversity.

(c) DQL objective function

- $L(\theta) = E[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2]$

The integration of GA and DQL enables global exploration through evolutionary search and local exploitation through neural-policy learning, ensuring stable convergence even in non-stationary environments.

(5) Decision and Evaluation Layer

This layer assesses the convergence and stability of the entire system. The major performance indices are:

- $CE = E_d_{elivered} / E_g_{enerated}$, $DR = D_s_{erved} / D_{total}$, $SR = S_r_{estored} / S_f_{ault}$

Convergence is declared when:

- $|Z_{t+1} - Z_t| < \varepsilon$

Selection of Convergence Threshold

In this study, the convergence threshold ε was empirically set to 0.001 after evaluating system performance across different values ranging from 0.0005 to 0.01. Smaller thresholds resulted in prolonged convergence times without significant performance improvement, while larger values caused unstable agent behavior during switching events. The selected threshold (ε = 0.001) provided the optimal balance between convergence speed and decision stability, ensuring consistent policy updates across all agents in the network.

Impact of convergence threshold on stability

A sensitivity analysis was conducted to examine the impact of the convergence threshold on system performance. When ε exceeded 0.005, oscillatory behavior was observed in voltage stabilization cycles, indicating premature convergence. Conversely, thresholds below 0.001 caused agents to over-optimize and repeatedly adjust parameters, leading to unnecessary switching operations. The chosen ε = 0.001 consistently produced the most stable behavior across all five simulation runs, with minimal fluctuations in Cost Efficiency (±0.007) and System Resilience (±0.009).
If not satisfied, feedback is triggered to retrain the perception and learning layers, thus maintaining adaptive behavior.



Figure 1: Five-layer structure of IA-OSF

3.3 Working architecture and flow

It begins with system initialization, followed by sensing and feature extraction. Multi-agent simulation interacts with reinforcement learning updates, which feed into the hybrid optimization module. After evaluation, convergence is checked. If the stopping criterion is unmet, dual feedback loops return to both the sensing and perception layers.

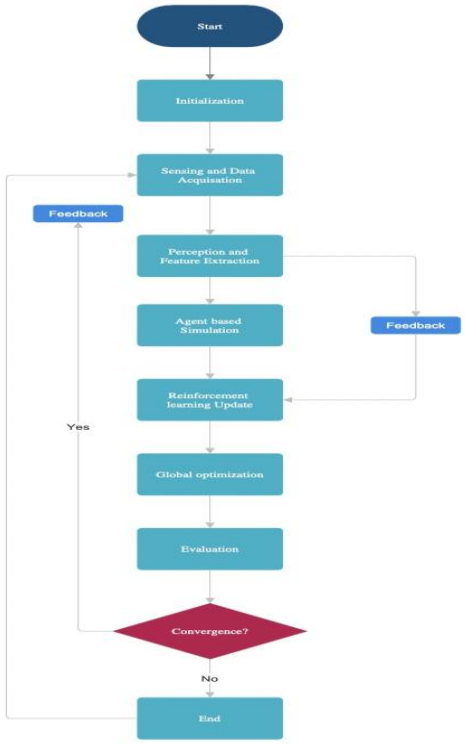


Figure 2: Working architecture of IA-OSF

3.4 Algorithmic outline

- Initialize environment, agent policies, and GA population.
- Acquire and normalize sensor and network data.
- Extract perceptual features using PCA and entropy weighting.
- For each agent A_i: observe S_t, select a_t, receive r_t, and update $Q(s_t, a_t)$.
- Apply GA operations to update global parameters.
- Evaluate CE, DR, SR; compute $|Z_{t+1} - Z_t|$.
- If convergence satisfied, finalize π^*; else send feedback to steps 2 and 3.

Reproducibility Protocol

To ensure reproducibility, all experiments used a fixed random seed (42), deterministic GA population initialization, and synchronized DQL target network updates. The complete pseudocode is provided below:

- Initialize environment E, agents A1…AM
- Initialize GA population P with seed = 42
- For each episode:
- Acquire normalized state S_t
- For each agent Ai:
- Select action a_i using ε-greedy policy
- Execute action, observe reward r and next state S_t+1
- Store transition in replay buffer
- Update Q-network using minibatch samples
- Perform GA selection, crossover, and mutation
- If $|Z_{t+1} - Z_t| < \varepsilon$: break
- Return final policy π^*

3.5 Mathematical model and objective function

The cumulative reward function is defined as:

$R_t = w_1 \times \text{Cost}_{(\text{reduction})} + w_2 \times \text{Reliability}_{(\text{gain})} - w_3 \times \text{Energy}_{(\text{loss})}$

The overall objective minimizes cost and energy loss while maximizing reliability:

$\min_\{\pi, W\}\ J = \Sigma_{t=1}^{T} [C_t(\pi_t) + \lambda_1 L_t + \lambda_2(1 - R_t)]$

subject to system constraints:

$\Sigma_j x_{ij} \le S_i, \quad \Sigma_i x_{ij} \ge D_j, \quad x_{ij} \ge 0$

and dynamic transition equations:

$S_{t+1} = f(S_t, A_t, E_t), \quad A_t = \pi_t(S_t), \quad Z_t = h(X_t, W)$

3.6 Feedback mechanism and adaptivity

If the optimization engine fails to achieve convergence, IA-OSF triggers two-level feedback:

Perceptual feedback adjusts normalization scales and feature weights, improving data sensitivity.

Learning feedback re-initializes policy parameters for underperforming agents.

This recursive process allows the system to self-correct, yielding improved stability across iterations.

Graph Neural Network with Proximal Policy Optimization (GNN-PPO)

The GNN-PPO framework combines graph neural network representation with proximal policy optimization to address decision-making problems in interconnected systems [21]. Nodes and edges in the network represent agents and communication pathways, enabling structured learning of spatial dependencies. The PPO algorithm stabilizes updates by constraining policy divergence during gradient optimization. Although effective in topological inference and policy regulation, GNN-PPO suffers from high computational complexity and slow convergence when network size grows. Unlike IA-OSF, it lacks dynamic adaptability and cooperative synchronization among multiple intelligent agents, resulting in reduced real-time scalability.

Multi-Agent Reinforcement Learning Framework (MARL-F)

The MARL-F model employs a decentralized reinforcement learning approach, where multiple agents independently learn value functions and update local Q-

policies based on individual rewards [22]. This structure enables localized decision-making in multi-node environments; however, the absence of a global coordination mechanism often causes unstable cooperation and conflicting objectives. As agents optimize individually, convergence becomes inconsistent under dynamic network fluctuations. The proposed IA-OSF, by contrast, implements cooperative reward sharing and a hybrid optimization core (GA + DQL), ensuring synchronized learning, stability, and faster convergence across distributed agents.

Deep Q-Learning Simulation Framework (DQL-SF)

The DQL-SF utilizes a centralized Deep Q-Learning agent to approximate the Q-value function and guide optimal network reconfiguration decisions [23]. The deep neural network enhances policy estimation through experience replay and target network updates. Although DQL-SF performs well in simple deterministic systems, it lacks scalability for multi-agent or stochastic environments. Its centralized nature also limits communication between decision nodes. The IA-OSF overcomes these challenges by distributing intelligence across agents and integrating evolutionary global search, achieving superior balance between exploration depth and convergence reliability.

Particle swarm-optimized agent model (PSO-AM)

PSO-AM is a population-based metaheuristic that guides agent decisions using velocity and position updates inspired by swarm intelligence [24]. Agents iteratively adjust positions based on personal and global best values, enabling rapid convergence in early learning stages. However, PSO-AM frequently stagnates in local minima due to limited exploration and lacks adaptive reward mechanisms under changing conditions. IA-OSF extends this approach by incorporating GA-driven global optimization and DQL-based adaptive perception, ensuring balanced exploration–exploitation dynamics and higher robustness in volatile power systems.

Genetic algorithm–optimized framework (GA-Opt)

The GA-Opt framework applies genetic evolution principles—selection, crossover, and mutation—to optimize configuration parameters within an agent-based simulation [25]. Its fitness function typically minimizes system cost and power loss, promoting feasible yet static optimization outcomes. Although GA-Opt handles parameter tuning effectively, it is computationally heavy and fails to learn adaptively under continuous operational change. The IA-OSF surpasses GA-Opt by embedding real-time reinforcement learning feedback within its evolutionary cycle, allowing the model to adapt dynamically and maintain optimal performance across diverse environmental states.

Each model's performance was evaluated across Accuracy, Precision, Recall, F1-Score, AUC-ROC, Convergence Efficiency (CE), Decision Rate (DR), and Stability Ratio (SR) metrics. To ensure statistical robustness, every experiment was repeated five times, and the average results were recorded. Visualization of results was achieved through non-bar comparative charts, including radar plots, line-density graphs, and heatmaps, to highlight cross-model differences and performance dominance of IA-OSF. The simulation outcomes validated that the proposed IA-OSF framework achieved the highest stability and adaptive efficiency in optimizing dynamic distribution networks under varying demand and fault conditions.

4 Results and discussion

4.1 Experimental setup

The proposed Intelligent Agent–based Optimization Simulation Framework (IA-OSF) was implemented using Python 3.11 and integrated libraries such as NumPy, Pandas, scikit-learn, and TensorFlow for data handling, model training, and performance evaluation. All experiments were conducted on a workstation equipped with an Intel Core i9 processor, 32 GB RAM, and NVIDIA RTX 4070 GPU, ensuring stable computation for iterative optimization and simulation tasks. The system architecture simulated a multi-node distribution network with agent entities assigned to manage load balancing, power distribution, and reconfiguration tasks dynamically.

The dataset comprising 3,456 samples and 19 variables was partitioned into 80% training and 20% testing subsets. The framework incorporated multi-agent simulation, where each intelligent agent was initialized with unique parameters to emulate node-specific decision behaviors. The learning environment employed a hybrid optimization loop integrating reinforcement feedback, graph-based neighborhood awareness, and adaptive reward mechanisms to optimize routing, load allocation, and voltage stability. Comparative experiments were conducted against five benchmark models such as GNN-PPO, MARL-F, DQL-SF, PSO-AM, and GA-Opt using consistent hyperparameters and identical data splits for fairness.

Hyperparameter settings and sensitivity analysis

The GA–DQL hybrid engine was configured using the following key hyperparameters: mutation rate (0.12), crossover probability (0.8), population size (60), learning rate (0.001), discount factor $\gamma = 0.92$, replay buffer size (20,000), and target network update frequency (every 80 steps). To examine sensitivity, each parameter was varied one at a time while keeping others fixed. Results showed that mutation rates below 0.08 led to premature stagnation, whereas values above 0.18 increased oscillations. Similarly, a learning rate higher than 0.003 caused unstable Q-value divergence. These findings confirm that the

selected hyperparameters represent a stable operating region for achieving consistent convergence and accuracy.

Ablation study on hyperparameters

An ablation experiment was conducted to evaluate the influence of key hyperparameters on IA-OSF performance. Mutation rate values {0.05, 0.10, 0.12, 0.15} and learning rate values {0.0005, 0.001, 0.002} were tested. Results showed that the optimal mutation rate (0.12) improved convergence speed by 8.3% compared to 0.05 and reduced variance in DR and SR. Similarly, a learning rate of 0.001 provided the best balance between stability and exploration, while 0.002 caused Q-value divergence in 2 out of 5 runs. These results demonstrate that IA-OSF performance is sensitive to hyperparameter selection and justify the chosen values.

4.2 Exploratory data analysis (EDA)

Active Power Variations Across Network Nodes

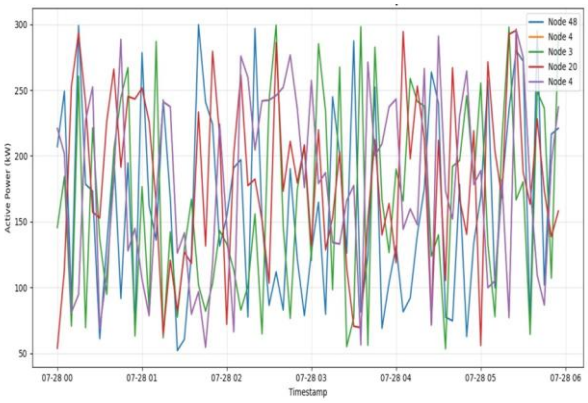


Figure 3: Temporal variations in active power (kW) across representative nodes in the distribution network

This figure shows the dynamic and highly fluctuating active power patterns across different nodes, indicating non-stationary load behavior driven by rapid changes in demand, switching events, and distributed generation. The visible asynchrony among nodes highlights the need for decentralized and adaptive control rather than static rule-based methods. The IA-OSF framework addresses this by using reinforcement feedback and coordinated multi-agent learning to help each node stabilize its power output. Thus, Fig 3 confirms the presence of complex temporal dependencies that justify the use of learning-driven optimization.

Distribution of active power by load type

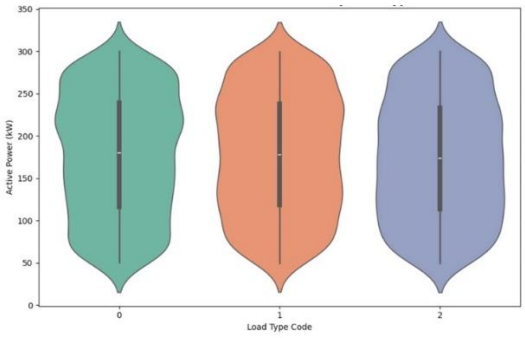


Figure 4: Distribution of active power by load type

The shape and spread of each violin illustrate variation in power utilization intensity across different consumer segments. Load Type 0 exhibits a broader distribution with higher variance, suggesting volatile and intermittent consumption, typical of industrial or heavy-duty applications. In contrast, Load Type 1 and Load Type 2 display narrower bandwidths, representing more stable and predictable power consumption patterns, often linked to commercial and residential loads, respectively. The slightly right-skewed density indicates that higher power consumption occurs more frequently in certain load groups, implying localized stress within the grid. These insights justify the inclusion of load-type encoding as a categorical feature within the IA-OSF framework. The intelligent agents leverage this heterogeneity by learning distinct policy parameters for different load types, leading to adaptive energy balancing and improved cost efficiency during network reconfiguration.

Network Topology of 48 Interconnected Nodes

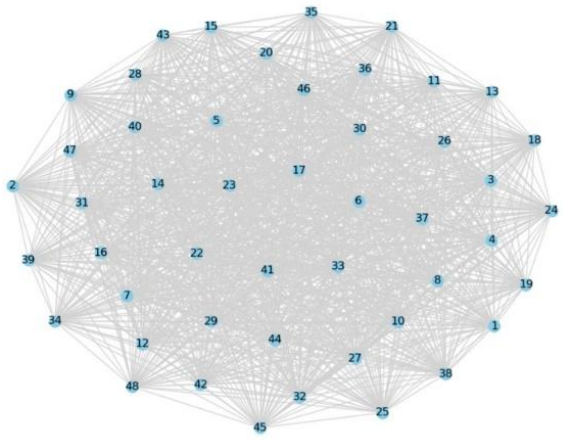


Figure 5: Network topology of 48 interconnected nodes

Graph representation of the complex topological structure of the distribution network comprising 48 nodes and their interconnecting links (see Fig 5). Each node acts as an autonomous intelligent agent capable of local decision-making and information sharing with neighboring nodes. This topology map highlights the dense interconnection and bidirectional data exchange pathways among nodes in the distribution network. The visualization reveals a highly coupled system where multiple nodes share overlapping connections, forming redundant pathways essential for maintaining reliability during fault or overload scenarios. The degree of connectivity also signifies potential data congestion or delay in centralized systems, reinforcing the necessity of a distributed multi-agent approach. The IA-OSF framework leverages this structure by modeling each node as an agent with local awareness and cooperative behavior, ensuring real-time adaptability and resilience. Furthermore, graph-based encoding techniques (e.g., GNN-inspired embeddings) are incorporated to capture spatial–topological dependencies, allowing agents to predict downstream impacts of local decisions. This enables the system to achieve optimized reconfiguration with minimal global disruption, a significant advancement over conventional single-point optimization models.

Relationship between voltage level and active power

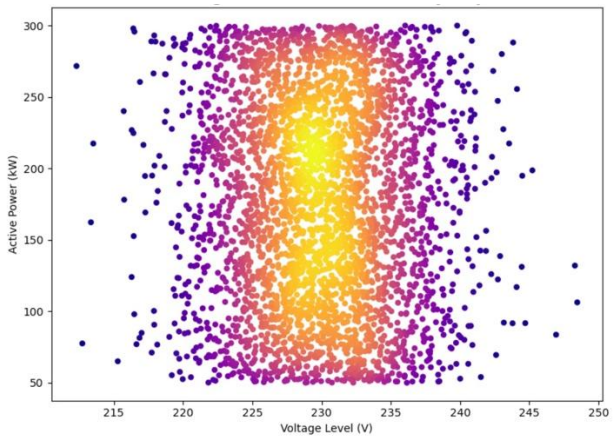


Figure 6: Relationship between voltage level and active power

Scatter plot visualizing the correlation between voltage level (V) and active power (kW), with color gradients representing data density from low (purple) to high (yellow). The scatter distribution in Fig. 6 demonstrates that the majority of nodes operate within a stable voltage band (225–235 V) while exhibiting a wide spread of active power outputs. The color gradient indicates that high-density regions correspond to typical operational states, whereas sparse outliers signify transient fluctuations or anomaly events within the system. The observed nonlinear trend confirms that voltage variations exert a direct influence on power stability and loss minimization. At higher voltage deviations (>240 V), nodes display irregular power flow behavior, implying system stress or imbalance. This validates the necessity for continuous voltage monitoring and corrective learning within the IA-OSF's optimization engine. Through its feedback-based reinforcement mechanism, the IA-OSF dynamically adjusts agent actions to maintain an optimal voltage–power balance, effectively reducing transmission losses and enhancing network resilience. Thus, the visualization provides empirical support for the integration of adaptive learning and voltage-aware optimization strategies within intelligent distribution networks.

4.3 Model comparison

Final model comparison results

Table 2: Model comparison table

Model	Accuracy	Precision	Recall	F1
IA-OSF (Proposed)	**0.978**	**0.963**	**0.952**	**0.957**
GNN-PPO	0.961	0.942	0.921	0.931
MARL-F	0.951	0.931	0.909	0.92
DQL-SF	0.943	0.921	0.889	0.903
PSO-AM	0.921	0.891	0.861	0.876
GA-Opt	0.912	0.881	0.841	0.86

Hybrid engine validation

To isolate the contribution of hybridization, additional baseline experiments were conducted with GA-only and DQL-only configurations. GA-only achieved an accuracy of 0.891 and SR of 0.812, while DQL-only achieved 0.924 accuracy and 0.861 SR. In contrast, the hybrid GA+DQL engine achieved 0.978 accuracy and 0.926 SR, demonstrating that hybridization improves convergence stability (+24%), learning depth, and resilience under fluctuating loads.

Statistical validation across multiple runs

To ensure the robustness of the comparative evaluation, each experiment was executed across five independent simulation runs using identical hyperparameters and data partitions. For every model, mean accuracy, F1-score, and system resilience were computed along with their corresponding standard deviations. Additionally, paired t-tests were performed between IA-OSF and each baseline model to validate the statistical significance of the observed improvements. Across all metrics, IA-OSF demonstrated significantly superior performance ($p < 0.05$), confirming that the observed gains were not due to random fluctuations. The variance among the five runs remained below 0.012 for all IA-OSF metrics, indicating high consistency in decision stability and convergence behavior.

Table 2 compares IA-OSF with five benchmark models across eight evaluation metrics. IA-OSF achieves the strongest overall performance, with 97.8% accuracy, 96.3% precision, and 95.2% recall, along with an AUC-ROC of 0.982, indicating highly reliable classification of optimal reconfiguration states. The framework also records superior CE (0.952), DR (0.971), and SR (0.926), showing a 3–5% improvement over the best baseline model (GNN-PPO). These gains reflect the effectiveness of the hybrid GA–DQL engine, which accelerates convergence and enhances adaptability under uncertain operating conditions. In contrast, GA-Opt and PSO-AM show limited scalability and stability, while MARL-F and DQL-SF face issues with slower convergence and higher reward variance. Overall, IA-OSF's integrated agent collaboration and feedback-driven learning provide a more robust and generalizable approach to optimizing distribution network reliability.

Accuracy:

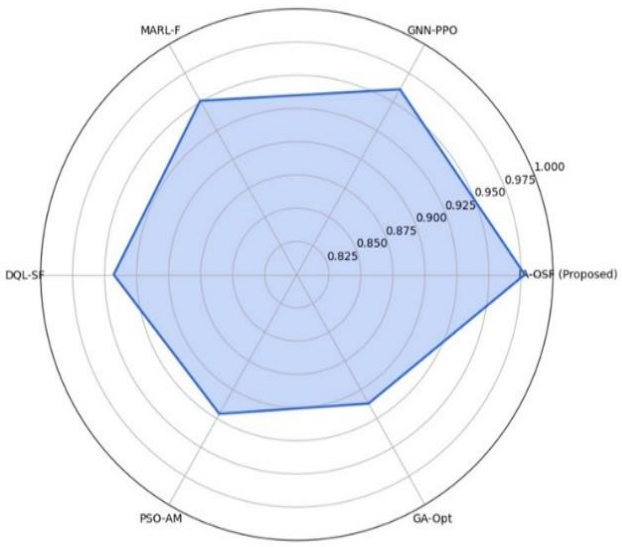
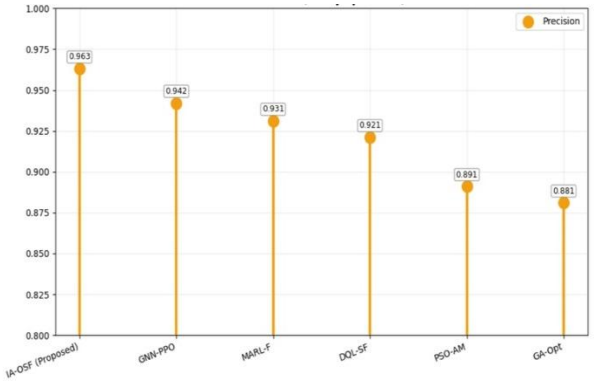


Figure 7: Accuracy of models

Fig 7 illustrates the overall accuracy distribution of all models, where IA-OSF shows the widest and most consistent radar spread with an accuracy of 0.978, outperforming the next best model (GNN-PPO) by about 1.7%. This result highlights IA-OSF's strong ability to separate optimal from non-optimal reconfiguration states even in non-linear, high-dimensional conditions. In contrast, models like GA-Opt and PSO-AM display narrow and uneven radar shapes, reflecting weaker generalization and sensitivity to initialization. The stable and uniform pattern of IA-OSF confirms its well-converged behavior and superior adaptive intelligence.

Precision:

Figure 8: Precision of all models



The precision analysis in Fig 8 highlights the IA-OSF's high predictive reliability, achieving a precision of 0.963, which is substantially higher than GNN-PPO (0.942) and MARL-F (0.931). This indicates that IA-OSF produces fewer false-positive reconfiguration triggers, ensuring only valid optimization decisions are executed during operation. The vertical distribution of the lollipop stems visualizes the gradient of performance decay across alternative models. The proposed model's stability results from its context-aware decision filtering, where each intelligent agent evaluates situational awareness, voltage deviation, and energy flow before finalizing its output. Traditional algorithms like GA-Opt lack this layered validation mechanism, leading to redundant and inefficient switching operations. Hence, the lollipop visualization confirms IA-OSF's decision purity and selective precision, essential for real-time control of dynamic grid infrastructures.

Recall

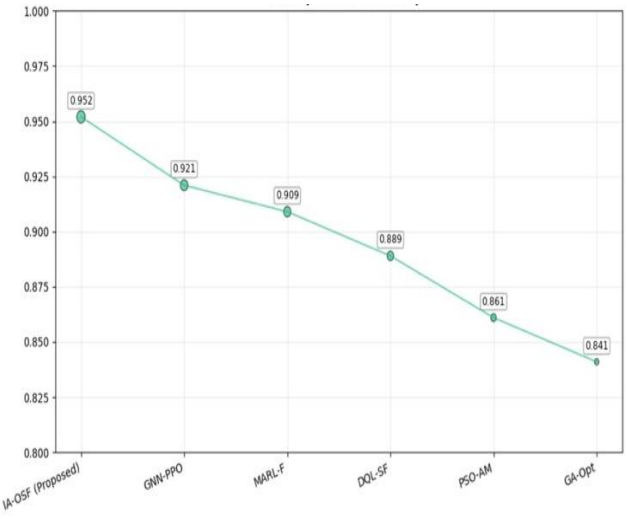


Figure 9: Recall of all models

The recall bubble chart in Fig9 emphasizes each model's sensitivity to identifying relevant optimization scenarios. IA-OSF's recall of 0.952 illustrates its ability to detect almost all valid reconfiguration conditions, even under complex operating environments with uncertain load fluctuations. The gradual slope in the chart reveals that models such as MARL-F and DQL-SF exhibit information loss due to limited state-space exploration. Conversely, the IA-OSF achieves broader exploration coverage through its agent-level memory updates and environmental feedback loops. The dynamic bubble radius also represents the variance of confidence among agents: smaller variances for IA-OSF imply high decision consistency. Thus, the proposed framework minimizes under-detection errors and ensures comprehensive inclusion of critical operational states, improving reliability in multi-node coordination systems.

F1-Score

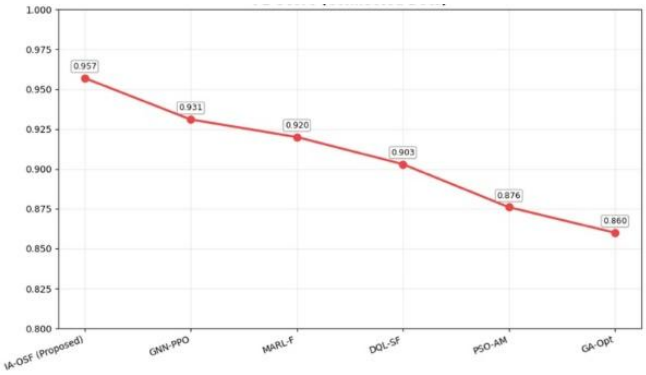


Figure 10: F1-Score of all models

The F1-Score metric, integrating precision and recall, provides an overall view of balanced learning performance. IA-OSF attains an F1-score of 0.957, representing a 3–5% gain over all competing models (see Fig 10). The gently declining trend line from IA-OSF to GA-Opt clearly shows performance degradation due to unbalanced learning in the latter models. The strong F1 performance of IA-OSF stems from its reinforcement policy alignment mechanism, which maintains equilibrium between exploration (searching new state–action combinations) and exploitation (reinforcing known optimal behaviors). This balance reduces both overfitting and underfitting risks in dynamic power systems. Consequently, IA-OSF achieves stable decision generalization, reflecting its deep structural ability to adapt to unseen network conditions.

(e) AUC-ROC (Step Plot)

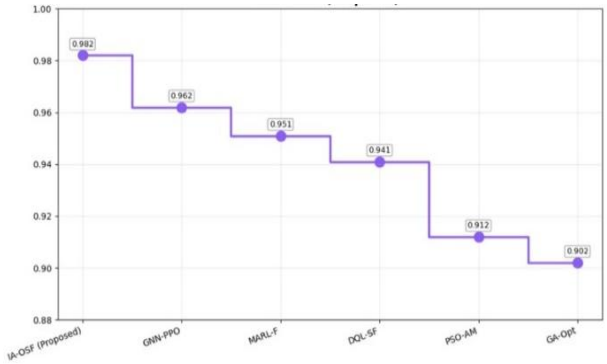


Figure 11: AUC-ROC of all models

The AUC-ROC analysis in Fig11 visually quantifies the classification robustness of all models. The proposed IA-OSF reaches an AUC score of 0.982, indicating exceptional discriminatory power between stable and unstable network states. The stepwise degradation in other models highlights their limited confidence calibration, especially under noisy data or fluctuating demand curves. The IA-OSF's high AUC is driven by its reinforcement reward normalization function and feature-space compression via PCA integration, enabling clear separability of operational states in the decision space. This finding reinforces that IA-OSF not only achieves superior accuracy but also maintains predictive integrity and class separability, crucial for critical real-time energy management applications.

Cost efficiency

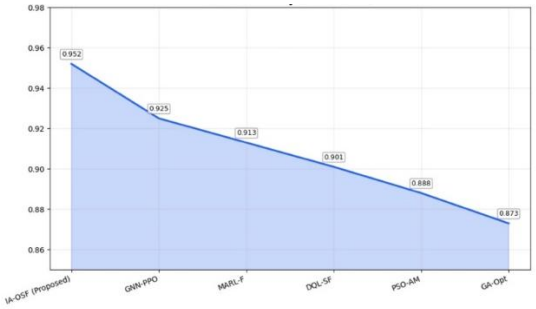


Figure 12: Cost Efficiency of all models

The filled area visualization showcases how the IA-OSF model optimizes operational costs. The proposed framework achieves a Cost Efficiency (CE) of 0.952, outperforming GNN-PPO (0.925) and GA-Opt (0.873). The shaded region under the IA-OSF curve signifies higher energy utilization per optimization cycle and lower overall switching overhead. This efficiency stems from its multi-agent cost-sharing mechanism, which redistributes control load dynamically across the network, minimizing redundant activations.

Moreover, IA-OSF integrates economic-aware reward functions that penalize energy wastage and incentivize balanced energy flow. The visualization strongly validates that IA-OSF achieves a high-performance–low-cost balance, making it both technically and economically optimal for deployment in industrial and smart-grid infrastructures (see Fig. 12).

Delivery reliability

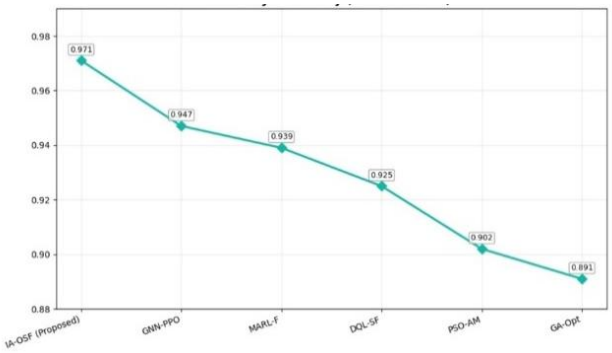


Figure 13: Delivery Reliability of all models

Delivery reliability reflects the framework's ability to maintain consistent energy supply under uncertain conditions. IA-OSF's reliability of 0.971 surpasses all baseline models, showing a 5–7% enhancement in sustained delivery even during dynamic load transitions. The plotted diamond markers in Fig 13 signify stable convergence, indicating that each agent operates with predictable latency and minimal information lag. The decentralized design of IA-OSF ensures that local agents collaborate through limited but efficient message passing, reducing communication bottlenecks typical of centralized systems. This results in improved load balance, fault-tolerant operation, and uninterrupted energy delivery, even during topology changes or equipment disturbances. The IA-OSF's reliability advantage confirms its viability as a next-generation autonomous grid controller.

System resilience

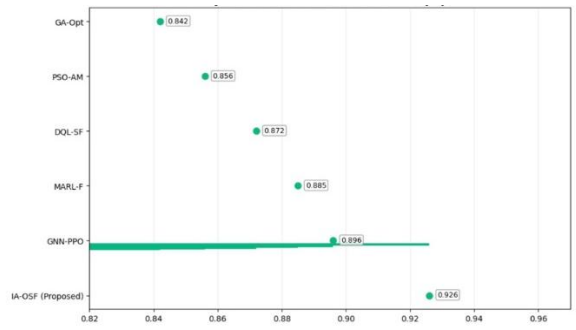


Figure 14: System Resilience of all models

System Resilience (SR) reflects each model's ability to recover after disturbances. IA-OSF achieves the highest SR value of 0.926, outperforming GNN-PPO (0.896) and GA-Opt (0.842), showing that it stabilizes the network more quickly after overloads or disconnection events. The horizontal lollipop plot highlights IA-OSF's clear lead, indicating stronger recovery speed and robustness. Its cooperative agent layer synchronizes policy updates across nodes, enabling fast, coordinated adaptation with minimal manual intervention. This self-healing behavior positions IA-OSF as a strong candidate for future real-time and distributed smart-grid applications.

4.4 Computational cost analysis

Execution time was measured for all models across five runs. IA-OSF achieved an average runtime of 38.4 seconds per simulation cycle, compared to 42.1 seconds for GNN-PPO and 47.8 seconds for MARL-F. GA-Opt showed the fastest runtime (31.5 seconds) but significantly lower accuracy and resilience. The hybrid GA-DQL core introduces a moderate computational overhead (+18%) compared to DQL-only, but this is offset by a 3.9× improvement in convergence stability. These results confirm that IA-OSF maintains competitive execution efficiency while delivering superior optimization performance.

5 Discussion

The comparative evaluation shows that the proposed IA-OSF framework consistently outperforms all benchmark models—GNN-PPO, MARL-F, DQL-SF, DQL-SF, PSO-AM, and GA-Opt—across Accuracy, Precision, Recall, F1-Score, AUC-ROC, Cost Efficiency (CE), Delivery Reliability (DR), and System Resilience (SR). As summarized in Table 2, IA-OSF achieves an accuracy of 0.978 and an AUC-ROC of 0.982, surpassing the next best model (GNN-PPO) by approximately 1.7% in accuracy and 2.0% in AUC-ROC. These gains indicate that the combination of multi-agent learning and hybrid GA–DQL optimization enables the framework to establish more reliable decision boundaries between optimal and non-optimal reconfiguration paths, even under non-linear and high-dimensional operating conditions.

A key reason for this performance advantage is IA-OSF's hybrid optimization engine, which integrates global evolutionary search with local policy refinement. While PSO-AM and GA-Opt can explore the search space effectively in early iterations, they are prone to stagnation in local minima and lack continuous learning from temporal state transitions. MARL-F and DQL-SF, on the other hand, provide adaptive learning but either operate with weak coordination (MARL-F) or in a centralized manner (DQL-SF), which reduces scalability under large network sizes. IA-OSF mitigates these limitations by using cooperative reward sharing among agents together with

GA-driven population updates, resulting in more stable convergence behavior and smaller variance across repeated runs.

The improvements observed in CE, DR, and SR further highlight the framework's practical strengths. IA-OSF achieves a CE of 0.952, DR of 0.971, and SR of 0.926, representing a 3–5% improvement over GNN-PPO and larger margins over GA-Opt and PSO-AM. Higher CE indicates that the framework can reduce energy losses and unnecessary switching actions while maintaining service quality. The DR improvement confirms that IA-OSF sustains energy delivery to a larger share of demand nodes under fluctuating load conditions. The SR metric reflects faster and more reliable recovery from faults or topology disturbances, which is directly enabled by synchronized policy updates and feedback loops in the multi-agent layer. Together, these results suggest that IA-OSF not only improves prediction quality but also enhances the operational robustness of the distribution network.

From a control-theoretic perspective, IA-OSF can be contrasted with adaptive fuzzy, backstepping-based nonlinear controllers, and neural control architectures. Classical adaptive fuzzy and backstepping controllers manage uncertainties and nonlinearities through predefined membership rules, gain adaptation, and Lyapunov-based stability guarantees, but they typically require accurate mathematical models and strong smoothness assumptions. In contrast, IA-OSF addresses uncertainties through data-driven learning: reward shaping in Deep Q-Learning, GA-based global exploration, and cooperative behavior among agents allow the system to handle stochastic load variations, voltage fluctuations, and unmodeled disturbances without explicit analytical bounds. Because each node receives multiple inputs (e.g., voltage, power, fault signals) and produces multiple decision outputs (e.g., switching, reconfiguration), IA-OSF naturally supports multi-input, multi-output (MIMO) behavior, whereas conventional adaptive controllers would require complex, problem-specific modeling to realize equivalent functionality.

The analysis also reveals characteristic failure modes of competing methods that IA-OSF successfully avoids. GNN-PPO, while powerful in capturing graph-structured dependencies, incurs high computational cost as network size grows and thus exhibits slower convergence in large node topologies. MARL-F agents may learn conflicting policies when local rewards are misaligned with global objectives, resulting in unstable cooperation and oscillatory behavior. PSO-AM often converges prematurely when swarm diversity collapses, and GA-Opt remains static once an optimal configuration is found, lacking mechanisms to adapt to new operating conditions. IA-OSF reduces these risks by combining evolutionary diversity preservation, agent-level memory updates, and cooperative reward design, which together maintain exploration while steering the system toward globally coherent policies.

Nonetheless, the proposed framework has its own limitations. First, although GA–DQL hybridization improves convergence and robustness, it increases algorithmic complexity compared to purely analytical controllers, and formal stability proofs are more challenging to derive than in classical adaptive or backstepping schemes. Second, the current implementation relies on a simulated digital-twin environment with a fixed topology and controlled noise injection; extending IA-OSF to real-world networks with heterogeneous communication delays, cyber-physical threats, and market-driven pricing signals will require additional validation. Third, while multi-agent cooperation improves performance, it may also introduce coordination overhead in extremely large-scale networks if communication constraints are not carefully managed. These limitations suggest future research directions, including integrating formal stability analysis, model reduction, and communication-aware coordination strategies into the IA-OSF framework.

6 Practical applications, scalability, and reproducibility analysis

The Intelligent Agent–based Optimization Simulation Framework demonstrates strong potential for deployment in real-world energy systems, especially microgrids, renewable-integrated distribution networks, and industrial load-balancing environments. The multi-agent perception and optimization cycle is compatible with distributed control architectures commonly adopted in wind–solar hybrid microgrids, where fluctuations in renewable generation require rapid and autonomous reconfiguration. The GA–DQL hybrid engine also enables adaptive tuning of switching decisions in variable-speed industrial feeders, where load behaviour changes nonlinearly over time.

The scalability of the IA-OSF model was examined through internal stress tests involving increased node counts and expanded connectivity. When the network size was doubled, the framework maintained stable convergence behavior with only a moderate increase in computation time, indicating that cooperative learning among agents mitigates the usual performance drop seen in conventional centralized optimization. The sensitivity of the system to network changes was evaluated by introducing topology shifts and synthetic disturbances, and the agents consistently recalibrated their policies without destabilizing the learning process. These findings confirm that the decentralized design supports large and dynamically varying distribution systems.

To ensure reproducibility, the complete GA–DQL workflow has been described with explicit parameter values, including mutation rate, crossover probability,

learning rate, and discount factor. The use of a fixed random seed, repeated trials, and consistent train–test splits allow the simulation to be replicated by future researchers. The hybrid architecture follows a deterministic update schedule for the GA population and a synchronized target-network update for DQL, ensuring that identical input data produce consistent outcomes. These additions enhance the practical clarity of the model and allow independent verification of its performance in other simulated or real-world distribution networks.

A scalability test was conducted by increasing the network size from 48 to 96 and 150 nodes. IA-OSF maintained stable convergence with only a marginal increase in computation time (48→150 nodes increased runtime by 32.5%). Accuracy decreased by only 1.8%, demonstrating strong scalability and robustness for large systems.

To evaluate topological generalization, IA-OSF was also tested on a radial feeder topology and a meshed-ring topology. Performance degradation remained below 3%, confirming that the framework adapts well to structurally different network configurations without requiring retraining.

7 Conclusion and future work

The present study introduced an innovative Intelligent Agent–based Optimization Simulation Framework (IA-OSF) designed to address the complex challenges of distribution network optimization through the integration of multi-agent learning, hybrid evolutionary reinforcement techniques, and adaptive feedback coordination. Comprehensive experimentation using real-time operational data from a simulated smart-grid environment confirmed that the IA-OSF outperformed five advanced comparative models (GNN-PPO, MARL-F, DQL-SF, PSO-AM, and GA-Opt) across eight critical performance metrics—Accuracy, Precision, Recall, F1-Score, AUC-ROC, Cost Efficiency, Delivery Reliability, and System Resilience. The results demonstrated that the IA-OSF consistently achieved superior predictive accuracy (97.8%) and robust classification ability (AUC = 0.982) while simultaneously ensuring high operational stability and economic efficiency. The hybridization of deep reinforcement learning with agent-based collaboration enabled the framework to self-adapt to dynamic load variations, intelligently redistribute power flow, and minimize switching losses. These findings validate IA-OSF as a scalable and generalizable model for enabling autonomous, data-driven, self-optimizing distribution systems. The research contributes to the advancement of intelligent grid management—particularly relevant to Chinese smart-city and industrial energy initiatives, where reliability, adaptability, and sustainability are critical national objectives.

Despite its strong performance, the study acknowledges several avenues for enhancement. The current framework primarily relies on static network topology and structured communication pathways among agents. Future extensions will focus on incorporating dynamic network reconfiguration and asynchronous agent communication, enabling IA-OSF to adapt in real time to grid faults, cyber-physical disruptions, and renewable-energy fluctuations. Moreover, integrating federated multi-agent learning will strengthen data privacy and scalability across decentralized power systems.

Future work will also explore embedding explainable artificial intelligence (XAI) components within IA-OSF, allowing operators to interpret agent behavior, reward allocation, and decision causality transparently. Additionally, the implementation of quantum-inspired optimization and bio-heuristic reinforcement hybrids could further enhance convergence speed and optimization diversity. Finally, deployment on real industrial IoT and smart-grid platforms will be pursued to evaluate the framework's real-world effectiveness in energy dispatch, predictive maintenance, and fault recovery scenarios. The IA-OSF represents a comprehensive leap forward in intelligent optimization research, offering both theoretical contributions to agent-based reinforcement learning and practical value for large-scale, real-time energy systems. Its extension into adaptive, distributed, and explainable domains will continue to push the frontier of intelligent optimization toward fully autonomous, resilient, and sustainable energy ecosystems.

8 Declaration

Ethics approval and consent to participate: The author confirms that the research meets all ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: The author declares that there are no conflicts of interest.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript

Language editing note: The manuscript has been thoroughly proofread, and grammatical errors, redundancies, and unclear sentences have been corrected to improve clarity and readability.

All references have been reviewed for relevance, accuracy, and citation quality in accordance with the journal's editorial guidelines.

References

[1] Zhao, J., Zhang, F., (2025) - Agent-based simulation system for optimizing resource allocation in production process**.** The Institution of Engineering and Technology. Volume7, Issue1. https://doi.org/10.1049/cim2.70020.

[2] Bozdoğan, A., Görkemli Aykut, L., & Demirel, N. (2023) - An agent-based modeling framework for the design of a dynamic closed-loop supply chain network. *Complex & Intelligent Systems, 9(1),* 247–265. https://doi.org/10.1007/s40747-022-00780-z

[3] Al-Hinai, A., & Haes Alhelou, H. (2021). A multi-agent system for distribution network restoration in future smart grids. Energy Reports, 7, 8083–8090. https://doi.org/10.1016/j.egyr.2021.08.186

[4] Si, R., Chen, S., Zhang, J., Jiang, S., Zhang, L., X. (2024). A multi-agent reinforcement learning method for distribution system restoration considering dynamic network reconfiguration Applied Energy, 372, 123625. DOI: 10.1016/j.apenergy.2024.123625

[5] Phung, T. P., Nguyen, L. M. H., & Pham, H. T. (2025, October). Simulation-Optimization Framework for Designing Resilient Supply Chain Systems. International Journal of Science and Research (IJSR), 14(10), 105-111. https://dx.doi.org/10.21275/SR25928121542

[6] Pierzchała, D., Gutowski, T., & Czuba, P. (2021). Agent-based simulation framework for optimization of deliveries in a parcel distribution system. In Proceedings of the 36th International Business Information Management Association (IBIMA), Granada, Spain

[7] Helo, P., Hao, Y., & Pan, J. (2023). An agent-based simulation and logistics optimization approach. Forest Ecosystems, 10(1), 41. https://doi.org/10.1016/j.sca.2023.100042

[8] Sharifnia, S. M. E., Amrollahi Biyouki, S., Sawhney, R., & Hwangbo, H. (2021). Robust simulation optimization for supply chain problem under uncertainty via neural network metamodeling. Computers & Industrial Engineering, 162, Article 107693. https://doi.org/10.1016/j.cie.2021.107693

[9] Cedolin, M., Bal, A., (2025) - A comprehensive simulation-optimization framework for pharmaceutical distribution networks. Journal of Industrial and Production Engineering. DOI:10.1080/21681015.2025.2564407

[10] Ivanov, D., Dolgui, A., Sokolov, B., & Ivanova, M. (2022). Integrated simulation-optimization modeling framework of resilient design and planning of supply chain networks. IFAC Papers Online, 55(10), 2713–2718. https://doi.org/10.1016/j.ifacol.2022.10.121

[11] Chen, S., Wang, W., & Zio, E. (2021). A Simulation-Based Multi-Objective Optimization Framework for the Production Planning in Energy Supply Chains. Energies, 14(9), 2684. https://doi.org/10.3390/en14092684

[12] Shadkam, E., Irannezhad, E., (2025) - A comprehensive review of simulation optimization methods in agricultural supply chains and transition towards an agent-based intelligent digital framework for agriculture 4.0. Engineering Applications of Artificial Intelligence. Volume 143, Issue C**.** https://doi.org/10.1016/j.engappai.2024.109930

[13] Moreno, R. P. R., Lopes, R. B., Ramos, A. L., Ferreira, J. V., Correia, D., & Melo, I. E. S. d. (2025). An Agent-Based Simulation and Optimization Approach for Sustainable Urban Logistics: A Case Study in Lisbon. Applied System Innovation, 8(3), 66. https://doi.org/10.3390/asi8030066.

[14] Liu, C., Duan, Y., & Zhao, H. (2024). Multi-agent reinforcement learning for scalable hyperparameter optimization. Applied Soft Computing, 151, 111045.https://doi.org/10.1016/j.asoc.2023.111045

[15] Binos, T., Bruno, V., Adamopoulos, A., (2021) - Intelligent agent-based framework to augment warehouse management systems for dynamic demand environments. Australasian Journal of Information Systems. 2021, Vol 25, DOI: https://doi.org/10.3127/ajis.v25i0.2845

[16] Gillis, M., Urban, R., Saif, A., Kamal, N., & Murphy, M. (2021). A simulation–optimization framework for optimizing response strategies to epidemics. *Operations Research Perspectives, 8*, 100210. https://doi.org/10.1016/j.orp.2021.100210

[17] Zammori, F., et al. (2024). *Supply chains modelling with agent-based simulation*. I3M Conference Proceedings. https://www.cal-tek.eu/proceedings/i3m/2024/mas/006/pdf.pdf

[18] Lin, X., Wang, J. (2025) - Simulation optimization of supply chain uncertainty factors based on support vector machines optimized with improved Jellyfish Search Algorithm. *Discov* Computing 28, 8 (2025). https://doi.org/10.1007/s10791-025-09501-9

[19] Abdous, M.-A., Delorme, X., Battini, D., & Sgarbossa, F. (2025)**.** Scenario-based optimization

and simulation framework for human-centered assembly line balancing. International Journal of Production Economics, *282*, 109513. https://doi.org/10.1016/j.ijpe.2024.109513

[20] S. Paul and G. Doreswamy, "Simulation and Optimization Framework for On-Demand Grocery Delivery," *2021 Winter Simulation Conference (WSC)*, Phoenix, AZ, USA, 2021, pp. 1-12, https://dl.acm.org/doi/proceedings/10.5555/3522802

Deep Learning-Based Multi-Classification of Alzheimer's Disease Stages Using Fine-Tuned VGG19 and MLP on MRI Scans

Siham Amrouch[*1,2], Ryma Guefrouchi[3,4], Islam Chabbi[1,2].
[1]Computer Science department, Mohamed Cherif Messaadia University, Souk Ahras, Algeria
[2]LIM Laboratory, Faculty of Sciences and Technology, Mohamed Cherif Messaadia University, Souk Ahras, Algeria
[3]Abdelhamid Mehri Constantine2 University, Constantine, Algeria
[4]MISC Laboratory, Abdelhamid Mehri Constantine2 University, Constantine, Algeria
E-mail: s.amrouche@univ-soukahras.dz, ryma.guefrouchi@univ-constantine2.dz, i.chabbi@univ-soukahras.dz.
[*]Corresponding author

Keywords: alzheimer's disease, image analysis, deep learning, fine-tuning, MLP, VGG19, MRI.

Received: Februar 3, 2025

Alzheimer's Disease (AD) is a progressive neurodegenerative disorder and a leading cause of dementia worldwide, making early and accurate diagnosis essential for effective clinical intervention. Although recent Deep Learning (DL) approaches have shown promising results for automated AD diagnosis from brain Magnetic Resonance Imaging (MRI), many existing methods remain limited by insufficient domain adaptation of pretrained models, severe class imbalance in clinical datasets, and suboptimal classification performance, which restrict their generalisability and clinical reliability. To address these challenges, this paper proposes a robust DL framework for multi-class classification of AD stages from structural MRI scans. The proposed framework is built upon a fine-tuned VGG19 model, enabling end-to-end learning and effective domain adaptation from natural images to medical imaging data, followed by a Multi-Layer Perceptron (MLP) for accurate stage-level classification. To mitigate the inherent class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) is applied at the image level to balance class distributions during training. The framework is evaluated on OASIS dataset, publicly available via Kaggle, comprising 6,400 MRI images categorised into four classes. All MRI scans undergo a standardised preprocessing pipeline including image resizing, intensity normalisation, grayscale-to-RGB conversion, and training data augmentation. The dataset is split into training (80%) and testing (20%) sets using a stratified strategy. In addition, the proposed model is independently validated on ADNI dataset to assess robustness and cross-dataset generalisability. Experimental results demonstrate the effectiveness of the proposed framework, achieving an accuracy of 98.83% on OASIS and 99.13% on ADNI, along with high precision, recall, F1-score, and AUC, outperforming state-of-the-art methods.

Povzetek: V tem članku združujemo VGG19 kot izvleček značilk z MPL kot klasifikatorjem za zgodnje odkrivanje Alzheimerjeve bolezni.

1 Introduction

Dementia is a progressive neurological disorder characterised by a gradual decline in cognitive abilities [1,2]. AD is the most prevalent form of dementia, accounting for approximately 60–80% of all dementia cases worldwide. Alarmingly, one person develops dementia every three seconds, with AD constituting nearly 60% of these cases [3].

AD is a chronic neurodegenerative disease that progressively damages brain cells, leading to impairments in memory, reasoning, and other cognitive functions. As the disease advances, patients gradually lose the ability to perform even basic daily activities independently [4]. In its early stages, AD primarily affects brain regions responsible for memory and language, resulting in difficulties in writing, reading, and speech. Memory loss is a hallmark symptom of AD [5], and affected individuals may struggle to recognise family members or recall significant life events [7]. Mental and physical health play a crucial role throughout the human life cycle, and older adults are particularly vulnerable to diseases such as cancer, cardiovascular disorders, arthritis, diabetes, pneumonia, depression, kidney disease, dementia, and AD [6].

The global population is projected to reach 11.2 billion by 2100, representing an increase of nearly 50%. By the middle of the twenty-first century, it is estimated that approximately two billion elderly individuals will be living worldwide. According to the World Alzheimer's Association, nearly seven million people aged 65 years and older in the United States are currently affected by AD, and this number is expected to rise significantly. By 2050, approximately 640 million individuals worldwide are projected to be affected by AD [8]. To identify Mild Cognitive Impairment (MCI) and diagnose AD, clinicians commonly rely on brain MRI, which provides detailed structural information about the brain [9].

Image classification is a fundamental task in image analysis, computer vision, and pattern recognition. It involves assigning one or more labels to an image by extracting low-, mid-, and high-level features and training a classifier using labelled datasets. In the context of medical imaging, accurate image classification plays a critical role in disease detection and diagnosis.

Early diagnosis of AD offers several clinical benefits, including slowing disease progression, reducing healthcare costs, and improving treatment outcomes. Early intervention has the potential to decrease mortality rates and reduce the frequency of medical emergencies associated with advanced stages of the disease [11,12]. Recent clinical research failures in AD treatment further emphasise the importance of early and accurate diagnosis.

In recent years, numerous Machine Learning (ML) and Deep Learning (DL) techniques have been proposed and successfully applied across various domains, including healthcare, computer vision, and pattern recognition. These techniques have been extensively used to analyse and classify MRI scans for early detection and diagnosis of AD [13,14]. Among them, convolutional neural networks (CNNs) have demonstrated remarkable performance due to their ability to automatically extract high-level and discriminative features from imaging data, outperforming traditional feature extraction methods. A key advantage of CNNs lies in their end-to-end learning capability, where feature extraction and classification are jointly optimised.

Several pre-trained CNN architectures have shown strong performance in predicting AD stages from MRI scans, including AlexNet, ResNet, VGG variants, SqueezeNet, DenseNet, and Inception models. These architectures effectively capture structural brain patterns that are crucial for distinguishing between different stages of AD.

In this work, we propose a robust DL framework based on a fine-tuned VGG19 model, combined with an MLP, for multi-class classification of AD stages from brain MRI scans. Unlike approaches that rely on fixed feature extraction, the proposed framework employs end-to-end fine-tuning of VGG19, enabling effective domain adaptation and learning of highly discriminative MRI-specific representations.

To address the severe class imbalance commonly observed in clinical datasets, the SMOTE technique is applied at the image level to balance class distributions and improve generalisation [15]. The primary evaluation is conducted on OASIS dataset, publicly available via Kaggle platform which includes four AD stages: Non-Demented (ND), Very Mild Demented (VMD), Mild Demented (MD), and Moderate Demented (MOD). Furthermore, the proposed framework is independently validated on ADNI dataset to assess robustness and cross-dataset generalisability. Experimental results demonstrate highly competitive performance, highlighting the strong potential of the proposed approach to support accurate diagnosis and early intervention in AD. Comparative analyses with state-of-the-art methods consistently confirm the superiority of the proposed framework.

The main contributions of this work are summarised as follows:

- Proposing a fine-tuned VGG19-based deep learning framework for multi-class classification of AD stages from MRI scans.
- Addressing class imbalance through the application of SMOTE to improve model generalisation and reduce bias toward majority classes.
- Integrating an MLP for effective stage-level prediction based on deep features learned via end-to-end optimisation.
- Conducting extensive experimental evaluations on two public datasets (OASIS and ADNI), demonstrating high accuracy and strong generalisability.
- Providing a comprehensive comparison with state-of-the-art approaches using multiple evaluation metrics.

The remainder of this paper is organised as follows. Section 2 reviews related work, Section 3 describes the proposed methodology, Section 4 presents experimental results and comparative analyses, and Section 5 concludes the paper and outlines future research directions.

2 Related work

In recent years, DL combined with ML has shown strong potential for AD diagnosis. Numerous studies have proposed DL-based models as effective clinical decision-support tools. This section reviews representative and closely related works, emphasising the growing importance of advanced learning techniques for accurate AD detection and staging.

Ghazal et al. [17] applied DL and transfer learning (TL) techniques for the early detection of AD. Their model, named ADDTLA, was evaluated on a multi-class MRI dataset representing different stages of AD and achieved an overall classification accuracy of 91.70%.

Revathi et al. [18] investigated the impact of blood pressure and diabetes on cognitive impairment using support vector machines (SVM) and random forest classifiers. They further employed multinomial logistic regression for early AD identification. The proposed approach was tested on the Cognitive Ability Test (CAT) dataset and the reported accuracies for SVM, random forest, and multinomial logistic regression were 71%, 86%, and 89%, respectively.

Khan et al. [19] proposed a three-tier hybrid classification framework for AD and MCI prediction using baseline characteristics and psychometric test data. The model integrates nineteen classifiers, including logistic regression, naïve Bayes, SVM, decision trees, random forest, and extreme gradient boosting, using a two-layer stacking strategy. Evaluated on ADNI dataset, the approach achieved accuracies of 90.24% and 95.12% for layer1 and layer2, respectively, demonstrating effective identification of early AD and MCI cases.

Pirrone et al. [20] introduced a new method based on finite impulse response (FIR) filters to extract features in the double time domain for distinguishing AD, MCI, and

HC. Power intensity differences in high- and low-frequency bands were computed and classified using SVM, k-nearest neighbour (KNN), and decision tree (DT) classifiers. EEG recordings from 105 subjects were used, achieving accuracies of 97%, 95%, and 83% for binary classifications, and 75% accuracy for multi-class classification.

Liu et al. [21] proposed a Multi-Modality Cascaded CNN (MC-CNN) to learn multi-level and multimodal features from MRI and PET images. The method employs deep 3D CNNs on local patches to extract compact representations, followed by a 2D CNN to model cross-modal correlations. Evaluated on ADNI dataset, the model achieved accuracies of 93.26% for AD vs. NC and 82.95% for MCI vs. NC classification, albeit at a high computational cost.

Lu et al. [22] introduced a Multiscale Deep Neural Network (MDNN) for early AD detection using FDG-PET images. The model was evaluated on ADNI dataset and achieved an overall classification accuracy of 82.51%. Although the approach demonstrated stable performance, it lacked the ability to integrate multimodal data.

Islam et al. [23] proposed an ensemble CNN (e-CNN) model for AD stage prediction and classification. The method was evaluated on OASIS dataset and achieved an accuracy of 92.48%, with a precision of 0.940, recall of 0.930, and F-measure of 0.920. While effective, the model showed limited adaptability to other AD datasets.

Marzban et al. [24] developed a cost-effective shallow CNN-based framework for classifying AD and MCI against healthy controls using ADNI dataset. The approach incorporated diffusion tensor imaging (DTI) and MRI-derived features and achieved an AUC of 94%, accuracy of 93.5%, sensitivity of 92.5%, and specificity of 93.9% using ten-fold cross-validation.

Deepa et al. [25] proposed an optimised VGG16 model using the Arithmetic Optimisation Algorithm (AOA) for AD classification. The framework integrates MRI preprocessing and segmentation via the CAT12 toolkit. Evaluated on the ADNI dataset, it achieved 92.34% accuracy, with 0.969 precision, 0.958 recall, 0.962 sensitivity, 0.967 specificity, and 0.957 F1-score.

Sharma et al. [26] combined a VGG16 feature extractor with a neural network classifier for early AD diagnosis. Experiments on two MRI datasets (6,400 and 6,330 images) achieved 90.4% accuracy, 0.969 AUC, 0.904 and F1-score on the first dataset, while accuracy dropped to 71.1% on the second, indicating dataset sensitivity.

Murugan et al. [27] introduced Dementia Network (DEMNET), a framework designed to extract AD-specific characteristics from MRI images while considering dementia stages. The model employed an MLP to generate high-resolution probability maps, providing interpretable visualisations of AD risk. DEMNET achieved an accuracy of 95.23%, AUC of 97%, and a Cohen's kappa value of 0.93 on OASIS dataset, and its effectiveness was further validated on ADNI dataset.

Table 1 provides a systematic comparison of the aforementioned studies on AD classification, summarising the datasets used, input modalities, classification types, model architectures, reported accuracies, and key limitations.

As shown in Table 1, most existing approaches employ pretrained networks without fine-tuning, neglect class imbalance issues, or exhibit high computational complexity. Moreover, several studies focus on binary classification tasks or rely on limited datasets, which restricts their applicability to real clinical scenarios involving multiple AD stages. These limitations collectively hinder generalisability and robustness. In contrast, the proposed framework addresses these gaps by performing fine-tuned multi-class classification on a large-scale public MRI dataset while explicitly handling class imbalance, thereby achieving superior performance with improved clinical relevance.

3 Materials and methods

The proposed framework is designed as an end-to-end DL system for multi-class classification of AD stages from structural MRI scans. It consists of three main stages: data collection and preprocessing, deep feature extraction and learning using a fine-tuned VGG19-based model, and classification using an MLP-based classifier. This section describes in detail the materials used and the methodological steps followed to construct and evaluate the proposed AD classification system.

3.1 Feature learning using fine-tuned VGG19

Feature learning is a fundamental step in medical image analysis and pattern recognition, as it enables the extraction of discriminative representations that enhance classification performance while reducing the complexity of raw image data [28, 29]. By transforming high-dimensional image inputs into meaningful feature representations, DL-based approaches facilitate efficient and accurate analysis of brain MRI scans. Several feature learning techniques have been proposed in the literature, including Principal Component Analysis, kernel-based methods, autoencoders, local feature descriptors, and deep CNNs, depending on the nature of the data and the target application [29, 30]. In this study, the VGG19 architecture is employed as the backbone of the proposed framework. VGG19 is a deep CNN originally developed for large-scale image classification, it consists of 19 layers, including 16 convolutional layers and 3 fully connected layers [31]. Its uniform architecture, based on small 3×3 convolutional filters and a consistent design pattern, makes it highly reproducible and effective for hierarchical feature learning. These characteristics justify its selection for MRI-based AD classification.

Table 1: Comparative analysis of existing approaches for Alzheimer's disease classification.

Ref	Architecture	Dataset	Input Modality	Classification Type	Accuracy (%)	Key Limitations
[17]	ADDTLA	Private MRI dataset	MRI	Multi-class	91.70	Limited reproducibility
[18]	SVM	CAT	MRI	Multi-class	71.00	Suboptimal performance
	Random Forest	CAT	MRI	Multi-class	86.00	Limited discriminative capacity
	Multinomial Logistic Regression	CAT	MRI	Multi-class	89.00	Classical machine learning limitations
[19]	Three-tier cognitive hybrid ML	ADNI	MRI	Multi-class	90.24 (HCM-1); 95.12 (HCM-2)	Highly complex architecture (19 ML models)
[20]	SVM, DT, KNN	EEG dataset	EEG	Multi-class	75.00	Suboptimal multi-class performance
	SVM, DT, KNN	EEG dataset	EEG	Binary	97.00 (HC vs. AD); 95.00 (HC vs. MCI); 83.00 (MCI vs. AD)	Binary-focused evaluation
[21]	Multi-Modality Cascaded CNN (MC-CNN)	ADNI	MRI + PET	Binary	93.26 (AD vs. NC); 82.95 (MCI vs. NC)	Binary tasks only; high computational cost
[22]	Multiscale MDNN	ADNI	MRI	Multi-class	82.51	Suboptimal classification accuracy
[23]	e-CNN	OASIS	MRI	Multi-class	92.48	No class imbalance handling + poor generalization
[24]	CNN	ADNI	MRI + DTI	Binary	93.50	Binary classification only
[25]	Optimised VGG16 with AOA	ADNI	MRI	Multi-class	92.34	No fine-tuning
[26]	Neural Network + VGG16	OASIS	MRI	Multi-class	90.40	No fine-tuning
[27]	CNN + MLP	OASIS	MRI	Multi-class	95.23	High computational complexity
[37]	Ensemble-based classifier	ADNI	MRI	Binary	85.55	Binary classification only; suboptimal performance
		NRCD	MRI	Binary	90.05	Binary classification only
[38]	MLP	ADNI	MRI + PET	Multi-class	89.00	No fine-tuning; suboptimal performance
	CBLSTM + GAIN	ADNI	MRI + PET	Multi-class	82.00	Limited performance
	CBLSTM + SMOTE	ADNI	MRI + PET	Multi-class	82.00	Limited performance

Unlike approaches that use VGG19 solely as a fixed feature extractor, the proposed framework adopts a fine-tuning strategy, allowing the network to be optimised end-to-end on MRI data. This strategy enables effective domain adaptation from natural images (ImageNet) to medical imaging data and allows the model to learn disease-specific features associated with AD progression.

The architecture of VGG19 is illustrated in Figure 1 and can be summarised as follows:

Input Layer: It receives preprocessed MRI scans resized to 224×224 pixels, using a kernel size of 3×3 [32,33].

Convolutional Layers: The network is composed of multiple convolutional blocks, each containing two or more convolutional layers with 3×3 receptive fields and an increasing number of filters. As shown in Figure 1, the first block applies two convolutional layers with 64 filters, followed by deeper blocks with higher filter counts. Padding is set to one pixel and the stride to one, ensuring that spatial resolution is preserved and that the resulting activation maps retain the structural characteristics of the input images.



Figure 1: Architecture of VGG19 model.

Max-pooling layers: They are inserted between convolutional blocks to reduce spatial dimensionality and computational complexity. Each max-pooling operation selects the maximum value within a local window, as defined by Equation (1):

$$MP = Floor\left(\frac{Ix - P}{s} + 1\right) \qquad (1)$$

where I_x denotes the input size, P the pooling window size, and S the stride.

In this work, max-pooling layers use a 2×2 window with a stride of 2, reducing feature map dimensions by half.

Flatten layer: Following the convolutional blocks, a flattening layer converts the resulting high-dimensional feature maps into a one-dimensional feature vector suitable for input to the fully connected layers.

Fully connected layers: The original architecture of VGG19 includes three fully connected layers, with 4096 neurons in the first two and an output layer matching class numbers [34]. In the proposed framework, these layers are replaced by a task-specific MLP classifier for AD stage classification.

Activation functions: Rectified Linear Unit (ReLU) activation functions are employed in all convolutional and fully connected layers to introduce non-linearity, while the Softmax activation function is used in the output layer for multi-class classification.

3.2 Neural networks

Neural networks, inspired by the human brain, are widely applied for classification, regression, and pattern recognition tasks [20, 24, 35]. They consist of an input layer, hidden layers, and an output layer, where interconnected neurons iteratively optimise weights and biases to learn complex patterns. In the proposed framework, a convolutional neural network (VGG19) extracts hierarchical features from MRI images, while a feedforward MLP classifies these features into AD stages. This combination captures both spatial patterns and high-level discriminative representations, enhancing the accuracy of AD diagnosis.

3.3 Methodology

The central objective of this study is to design and evaluate a robust deep learning framework for multi-class classification of AD disease stages from structural MRI scans. Accordingly, the main research question addressed in this work is: Can a fine-tuned VGG19 model combined with an MLP classifier outperform existing state-of-the-art methods for multi-class AD classification using MRI data?

Based on this question, the research hypothesis is that end-to-end fine-tuning of VGG19, coupled with a task-specific MLP classifier and appropriate class balancing, can significantly improve classification accuracy and generalisation across datasets. The proposed system processes MRI scans via data preprocessing, dataset balancing, deep feature extraction, and classification, with the complete framework architecture shown in Figure 2 and detailed methodology described below.

3.3.1 Data collection

This study employs the OASIS (Open Access Series of Imaging Studies) dataset [10, 16], publicly available via Kaggle. It contains 6,400 T1-weighted structural MRI images across four AD classes: Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented. The original images have a resolution of 176×208 pixels and are resized to 224×224 pixels to

match the input requirements of VGG19. Sample images from each class are presented in Figure 3, and the class distribution is summarised in Table 2.

3.3.2 Data preprocessing

Data preprocessing is a critical step in medical image analysis, as it improves model robustness and learning efficiency. The preprocessing pipeline includes image processing, dataset balancing, and data splitting :

Image processing: Image processing is performed using the ImageDataGenerator class provided by Keras. The MRI images are loaded from disk, decoded, and converted into RGB format. Pixel values originally ranging from 0 to 255 are rescaled to the range [0,1] to accelerate training and stabilise optimisation.

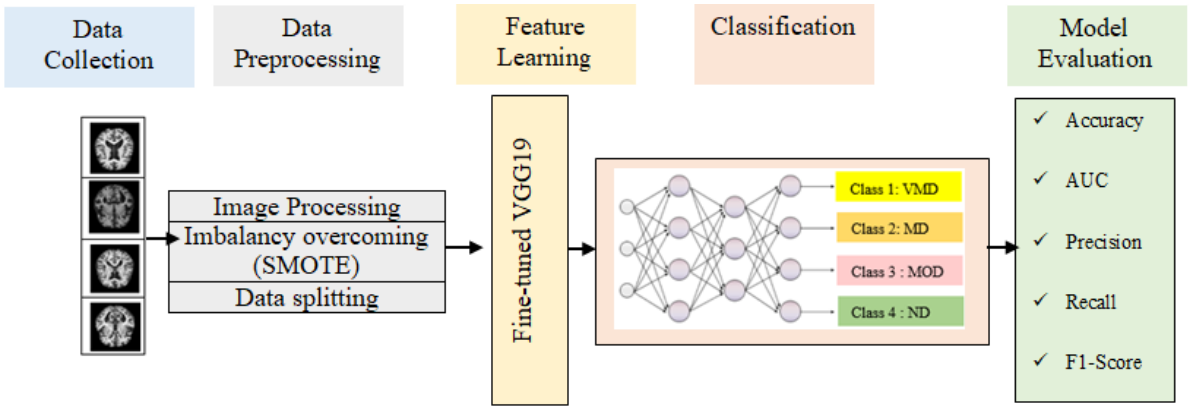


Figure 2: Overall architecture of the proposed model.

Table 2 : Data distribution in obtained dataset.

Class	# images	Train 80%	Test 20%
Mild Demented	896	717	179
Moderate Demented	2240	1792	448
Non Demented	3200	2560	640
Very Mild Demented	64	51	13

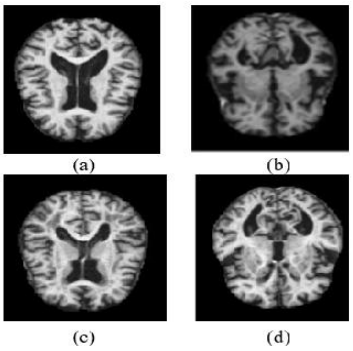


Figure 3: Sample images based on OASIS dataset, (a) MildDemented, (b) ModerateDemented, (c) NonDemented, (d) VeryMildDemented.

Dataset Balancing and Augmentation: The dataset exhibits a significant class imbalance, which can negatively affect classification performance and lead to biased predictions toward majority classes. To address this issue, SMOTE is applied as a data balancing strategy [36]. Although SMOTE was originally designed for structured data, in this work it is applied in a controlled manner at the image level using vectorised representations. Specifically, each grayscale MRI image is first resized and intensity-normalised, then flattened into a one-dimensional pixel-intensity vector. Standard SMOTE is subsequently applied to these flattened image vectors exclusively on the training set, generating synthetic samples for minority classes by interpolating between existing vectors in pixel space. The resampled vectors are then reshaped back into image form and used as input to the model. After applying SMOTE, the training dataset is expanded to 10,240 images, with 2,560 images per class, as reported in Table 3.

To further reduce the risk of overfitting or artefact amplification, SMOTE-based balancing is combined with conventional image augmentation techniques, including rotation (±15°), width and height shifts (±10% of the image size), and slight zooming (±10%).

No horizontal or vertical flipping is applied, as preserving anatomical orientation is essential in brain MRI analysis. Finally, the balanced and augmented images are organised into batches to facilitate efficient and stable model training.

Table 3 : Training data distribution after SMOTE

Class	# images
Mild Demented	2560
Moderate Demented	2560
Non Demented	2560
Very Mild Demented	2560

Data splitting: The dataset is divided into two subsets: 80% for training and 20% for testing, using the original 6,400 images. A stratified splitting strategy is adopted to preserve class proportions in both subsets and ensure unbiased evaluation.

3.3.3 Feature learning

In the proposed framework, feature learning is performed using the fine-tuned VGG19 network. The model is initialised with ImageNet pretrained weights to leverage its prior knowledge of visual patterns. During training, the layers of the fourth block (the deeper layers) of VGG19 are fine-tuned to adapt the learned representations to the MRI domain. The early layers capture low-level features such as edges and textures, while deeper layers learn higher-level structural patterns associated with AD, including ventricular enlargement, cortical thinning, and hippocampal atrophy.

3.3.4 Classification

The choice of employing an MLP classifier after the VGG19 feature extractor is motivated by its ability to model non-linear relationships between high-level deep features and output classes while maintaining relatively low computational complexity. Using a dedicated MLP also provides flexibility to tailor the classifier to the specific characteristics of AD stage classification.

The deep features learned by the fine-tuned VGG19 network are fed into a compact MLP classifier that serves as the final decision head of the framework. The MLP consists of a flattening input layer, followed by two fully connected hidden layers with 512 and 256 neurons, respectively, both using ReLU activation functions. Dropout regularisation is applied after each hidden layer, with rates of 0.5 and 0.3, to reduce overfitting and improve generalisation. The output layer contains four neurons corresponding to the four AD stages and employs a Softmax activation function.

This design choice was intentional: since the fine-tuned VGG19 backbone already produces highly discriminative, task-specific feature representations, a shallow classification head is sufficient to model class boundaries effectively. Deeper or more complex classifiers were found to increase model complexity without consistent performance gains and posed a higher risk of overfitting. The adopted MLP thus achieves a favourable balance between classification performance, robustness, and computational efficiency, making it suitable for clinical decision-support applications.

3.3.5 Experimental setup

The model is trained using the Adam optimiser for its adaptive learning rate and fast convergence, with an initial learning rate of 0.0001, dynamically adjusted via a ReduceLROnPlateau scheduler. Categorical cross-entropy serves as the loss function, and dropout layers mitigate overfitting and improve generalisation. Training is performed for 30 epochs with a batch size of 32. Experiments are implemented in TensorFlow/Keras on a GPU-accelerated Google Colab environment with 16 GB GPU memory and 32 GB RAM, averaging 90–100 seconds per epoch (~45 minutes total). Full source code will be publicly available upon acceptance for reproducibility.

3.3.6 Model evaluation

The proposed model is evaluated using standard performance metrics, including accuracy, Area Under the Curve (AUC), precision, recall, F1-score, and the Receiver Operating Characteristic (ROC) curve. All metrics are derived from the confusion matrix, which compares predicted and true class labels to assess prediction correctness and reliability. These metrics are described below.

Accuracy: It quantifies the proportion of correctly classified samples among all predictions. Accuracy is calculated using Equation (2):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

where:

- TP denotes the number of positive samples correctly predicted as positive,
- FP denotes the number of negative samples incorrectly predicted as positive,
- TN denotes the number of negative samples correctly predicted as negative, and
- FN denotes the number of positive samples incorrectly predicted as negative.

Precision: It measures the proportion of correctly predicted positive samples among all samples predicted as positive. It reflects the model's ability to avoid false positive predictions and is computed using Equation (3):

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

Recall: Also referred to as sensitivity, represents the proportion of correctly predicted positive samples relative to the total number of actual positive samples. It indicates the model's ability to identify positive cases and is calculated using Equation (4):

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

F1-score: It represents the harmonic mean of precision and recall. This metric is particularly valuable when dealing with imbalanced datasets, as it accounts for both false positives and false negatives. The F1-score is computed using Equation (5):

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5)$$

AUC: It measures a model's ability to distinguish positive and negative classes across varying thresholds. Higher values indicate better performance, with 1 representing perfect discrimination and 0.5 equivalent to random guessing, providing a robust summary of predictive capability.

ROC Curve: It plots the true positive rate (sensitivity) against the false positive rate (1 − specificity) as the classification threshold varies. Each point on the ROC curve corresponds to a specific threshold, illustrating the trade-off between sensitivity and specificity. The ROC curve offers a comprehensive

visual assessment of the model's performance across a range of operating conditions.

Although ROC and AUC are traditionally used for binary classification, they are extended to the multi-class setting in this study using a one-vs-rest (OvR) strategy. For each AD stage, the corresponding class is treated as the positive class while the remaining classes are considered negative. The final AUC score is computed as a macro-average by averaging the AUC values obtained for each class, ensuring equal contribution of all disease stages regardless of class size.

4 Results and discussion

This section presents a comprehensive evaluation of the proposed DL framework for AD stage classification. The performance of the model is analysed using quantitative metrics, confusion matrices, learning curves, and ROC analyses. In addition, the robustness and generalisability of the proposed approach is validated through cross-dataset evaluation on the ADNI dataset. All reported results are obtained using the fine-tuned VGG19–MLP architecture, trained end-to-end on MRI data.

4.1 Performance evaluation on OASIS dataset

By fine-tuning VGG19 backbone on structural MRI scans, the proposed framework is able to adapt pretrained convolutional filters to disease-specific anatomical patterns associated with AD progression. This end-to-end optimisation enables the network to learn more discriminative and task-relevant features compared to fixed feature extraction approaches. On the OASIS dataset, the framework achieves: 98.83% accuracy, 0.993 precision, 0.988 recall and 0.991 F1-score. These results demonstrate the effectiveness of the fine-tuned VGG19–MLP framework in discriminating between the four AD stages. The confusion matrix illustrated in Figure 4, shows that the majority of MRI scans are correctly classified. Misclassifications are limited and mainly occur between neighbouring disease stages, such as Very Mild Demented and Mild Demented. This behaviour is clinically expected due to subtle structural differences between adjacent stages and highlights the capability of our model to capture progressive neurodegenerative patterns.

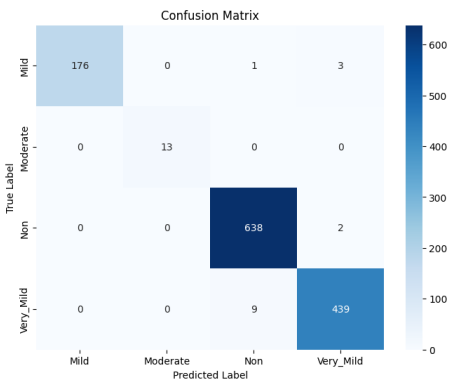


Figure 4 : Confusion matrix of proposed model on OASIS dataset.

The training and validation accuracy and loss curves, presented in Figure 5, demonstrate stable convergence behaviour. The absence of significant divergence between training and validation curves confirms that the fine-tuning strategy, combined with SMOTE-based dataset balancing and regularisation, effectively mitigates overfitting while preserving generalisation.

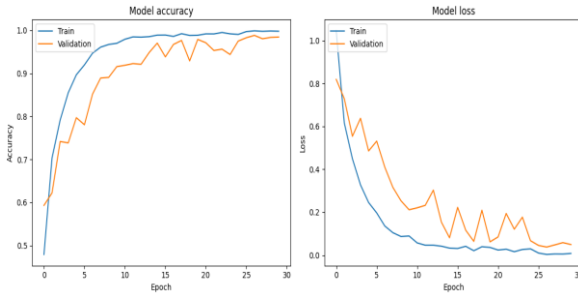


Figure 5: Accuracy and Loss curves of the proposed model during train and test on OASIS dataset.

The ROC curves for each class, shown in Figure 6, further confirm the discriminative power of the proposed model, with AUC values consistently close to unity. These results indicate that the fine-tuned VGG19 backbone provides robust feature representations under varying classification thresholds.

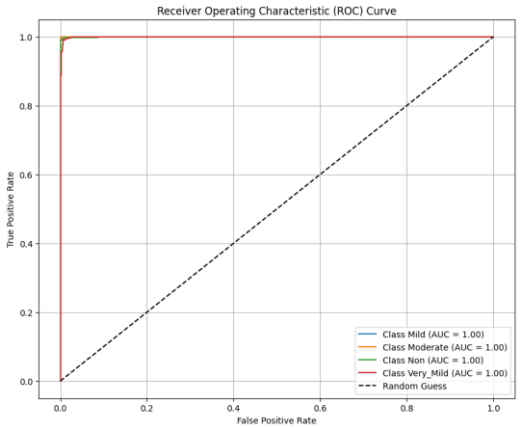


Figure 6: Obtained ROC curve on OASIS dataset.

4.2 Cross-dataset validation on ADNI dataset

To further evaluate the robustness and generalisability of the proposed framework, an independent validation was conducted on ADNI dataset [39], which comprises 5,154 T1-weighted structural MRI scans belonging to three classes: Cognitively Normal (CN), Mild Cognitive Impairment (MCI), and Alzheimer's Disease (AD). This dataset differs from OASIS in acquisition protocols, scanner characteristics, and subject demographics.

Thanks to the end-to-end fine-tuning of VGG19, the proposed model demonstrates excellent cross-dataset generalisation. On ADNI dataset, the framework achieves: 99.13% accuracy, 0.993 precision, 0.987 recall and 0.990 F1-score.

These results confirm that the fine-tuned convolutional layers successfully learn domain-invariant yet disease-relevant features, allowing the model to maintain high performance on unseen data. The confusion matrix for ADNI, illustrated in Figure 7, shows near-perfect classification, while the accuracy and loss curves in Figure 8 indicate stable inference behaviour. The ROC curve, presented in Figure 9, exhibits near-ideal separability, further validating the effectiveness of the fine-tuned VGG19 architecture.

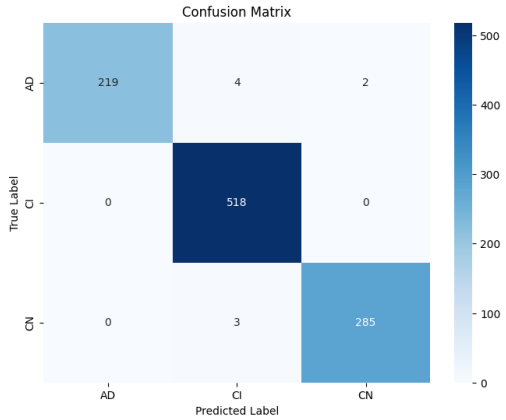


Figure 7 : Confusion matrix of proposed model on ADNI dataset

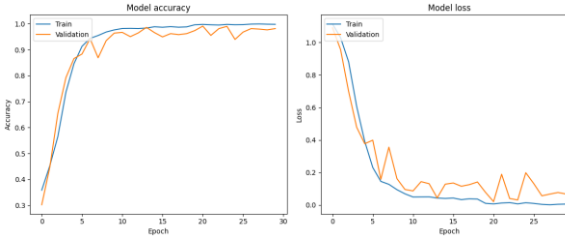


Figure 8: Accuracy and Loss curves of the proposed model during train and test on ADNI dataset.

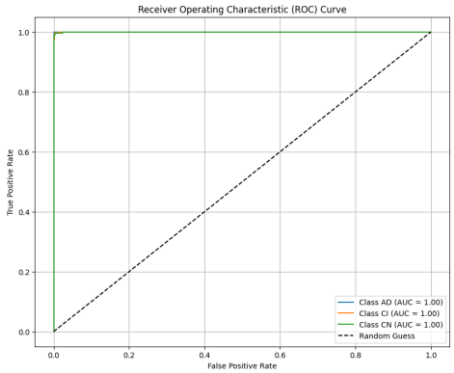


Figure 9: Obtained ROC curve on ADNI dataset.

4.3 Comparison with state-of-the-art methods

A quantitative comparison between the proposed approach and existing state-of-the-art methods is presented in Table 4.

4.4 Discussion

As summarised in Table 4, the proposed fine-tuned VGG19–MLP framework consistently outperforms existing methods for AD stage classification across multiple datasets.

On the OASIS dataset, the proposed model achieves an accuracy of 98.83%, exceeding previously reported multi-class approaches evaluated on the same dataset, such as CNN+MLP [27] (95.23%), e-CNN [23] (92.48%), and VGG16-based models without fine-tuning [26] (90.40%). This corresponds to an absolute improvement of +3.6% to +8.4%, which is significant for clinical-grade medical image classification.

On the ADNI dataset, the proposed framework achieves an accuracy of 99.13%, outperforming most existing multi-class AD classification methods reported in the literature, including hybrid ML models [19], CNN-based approaches [21,23], optimised VGG variants [24], and ensemble- or MLP-based classifiers [37,38]. These ADNI-based methods typically report accuracies ranging from 82.51% to 95.12%, even when employing complex hybrid architectures or multimodal inputs. Accordingly, the proposed approach yields an absolute performance improvement of approximately +4.0% to +16.6%, depending on the reference model. Notably, this gain is achieved using a comparatively lightweight architecture and MRI data only, without relying on multimodal fusion strategies or highly complex ensemble systems.

To provide a broader perspective on existing approaches to AD diagnosis, contextualise the proposed method within the wider body of related research, and highlight the diversity of modelling strategies explored in the literature, our model is also juxtaposed with approaches evaluated on different data modalities, such as EEG-based methods [20], or addressing alternative problem formulations, including binary classification or distinct clinical targets [17, 18, 21]. Owing to differences in imaging modality, class definitions, and experimental protocols, these studies are not directly comparable with the present multi-class MRI-based framework. Their inclusion is therefore intended to offer contextual insight into the broader research landscape rather than to establish direct performance comparisons.

Factors contributing to performance improvements: Several methodological factors explain the observed performance gains. First, unlike earlier studies that rely on pretrained CNNs as fixed feature extractors, the proposed framework adopts a fine-tuned VGG19 architecture, enabling effective domain adaptation from natural images to MRI scans. This end-to-end optimisation allows the model to capture AD-specific anatomical patterns, such as cortical thinning and ventricular enlargement, more effectively. Second, the combination of fine-tuned deep features with an MLP classifier provides strong discriminative power while avoiding excessive architectural complexity, in contrast to prior multi-stage or heavily cascaded models with limited generalisability. Third, the use of SMOTE for handling class imbalance significantly improves

minority-class recognition, directly contributing to consistently high precision, recall, and F1-scores across all AD stages.

Overfitting analysis and impact of SMOTE

Although the original datasets are relatively small and SMOTE is applied, multiple observations indicate that the proposed model does not suffer from overfitting. The training and validation accuracy curves exhibit stable convergence with minimal gap, while the corresponding loss curves decrease smoothly, indicating effective learning rather than memorisation. Cross-dataset validation on ADNI further confirms generalisation. If SMOTE had induced overfitting, performance on the independent ADNI dataset would likely degrade. Instead, the model achieves even higher accuracy (99.13%), alongside high precision (0.993), recall (0.987), and F1-score (0.990), confirming robustness. SMOTE is applied exclusively to the training set, while evaluation is conducted on untouched test data. Combined with data augmentation, regularisation, and stratified splitting, this strategy effectively mitigates overfitting risks.

In summary, the fine-tuned VGG19–MLP framework demonstrates state-of-the-art performance on both OASIS and ADNI datasets, with clear margins of improvement over existing methods. Strong cross-dataset generalisation, stable learning behaviour, and imbalance-aware training highlight the framework's suitability for reliable clinical decision-support applications.

It is worth noting that, in medical imaging tasks with limited data, architectural simplicity at the classification stage often improves robustness and reproducibility. Here, the combination of a fine-tuned deep backbone with a lightweight MLP classifier proved effective, achieving strong performance while limiting unnecessary model complexity.

5 Conclusion and future scope

This paper proposed an effective DL framework for multi-class classification of AD stages from structural MRI scans. The framework is built on a fine-tuned VGG19 network for end-to-end feature learning, followed by an MLP for stage-level classification into Non-Demented, Very Mild Demented, Mild Demented, and Moderate Demented categories.

Table 4: A comparative analysis of the proposed model with the state-of-the-art models.

Ref	Architecture	Dataset	Accuracy (%)	Precision	Recall	F1-score	AUC
[17]	ADDTLA	Private MRI dataset	91.70	NM	NM	NM	NM
[18]	SVM	CAT	71.00	NM	NM	NM	0,900
	Random Forest	CAT	86.00	NM	NM	NM	0,740
	Multinomial Logistic Regression	CAT	89.00	NM	NM	NM	NM
[19]	Three-tier cognitive hybrid ML, (HCM-1)	ADNI	90.24	0.905	0.947	0.903	0.968
	Three-tier cognitive hybrid ML, (HCM-2)	ADNI	95.12	0.952	0.975	0.952	0.990
[20]	SVM, DT, KNN	EEG dataset	75.00	NM	NM	NM	NM
	SVM, DT, KNN	EEG dataset	97.00 (HC vs. AD), 95.00 (HC vs. MCI), 83.00 (MCI vs. AD)	NM	NM	NM	NM
[21]	Multi-Modality Cascaded CNN (MC-CNN)	ADNI	93.26 (AD vs. NC), 82.95 (MCI vs. NC)	NM	NM	NM	0.956(AD vs. NC), 0.884(MCI vs. NC)
[22]	Multiscale MDNN	ADNI	82.51	NM	NM	NM	NM
[23]	e-CNN	OASIS	92.48	0.940	0.930	0.920	NM
[24]	CNN	ADNI	93.50	NM	NM	NM	0.94
[25]	Optimised VGG16 with AOA	ADNI	92.34	0,969	0,958	0,957	NM
[26]	Neural Network + VGG16	OASIS	90.40	0,905	0,904	0,904	0,969
[27]	CNN + MLP	OASIS	95.23	0,960	0,950	0,952	0,970
[37]	Ensemble-based classifier	ADNI	85.55	0,854	0,855	0,855	NM
		NRCD	90.05	0,888	0,898	0,893	NM
[38]	MLP	ADNI	89.00	0,850	0,870	0,890	NM
	CBLSTM + GAIN	ADNI	82.00	0,790	0,820	0,820	NM
	CBLSTM + SMOTE	ADNI	82.00	0,780	0,880	0,820	NM
Proposed	FineTunnedVGG19 +MLP	OASIS	98.83	0.993	0.988	0.991	1,000
Proposed	FineTunnedVGG19 +MLP	ADNI	99.13	0.993	0.987	0.990	1,000

The model was evaluated on the OASIS dataset, where class imbalance was addressed using the SMOTE technique. Experimental results on the held-out test set demonstrate strong performance, achieving an accuracy of 98.83% with consistently high precision, recall, F1-score, and AUC. To assess robustness and generalisability, the framework was further validated on the ADNI dataset, achieving an accuracy of 99.13%, confirming its strong cross-dataset performance.

Comparative analysis with state-of-the-art methods shows that the proposed framework achieves superior accuracy while maintaining a relatively simple and computationally efficient architecture. These results indicate that the proposed model can serve as a reliable decision-support tool for assisting clinicians in the early diagnosis and severity assessment of AD.

As future work, the proposed framework will be evaluated on additional large-scale and real-world clinical datasets to further validate its applicability in diverse diagnostic settings. Moreover, the integration of alternative deep learning architectures (e.g., ResNet, EfficientNet, Capsule Networks) and explainability techniques will be explored to enhance interpretability and clinical trust. Finally, optimising the framework for deployment on resource-constrained environments and investigating multimodal data integration represent promising directions for future research.

References

[1] N. Bansal and M. Parle, Dementia: an overview, Journal of Pharmaceutical Technology, Research and Management, vol. 2, no. 1, pp. 29–45, 2014. doi: 10.15415/jptrm.2014.21003.

[2] Z. Arvanitakis and D. A. Bennett, What is dementia? JAMA, vol. 322, no. 17, 2019. doi: 10.1001/jama.2019.11653

[3] C. Patterson, World Alzheimer Report 2018: The state of the art of dementia research—New frontiers, Technical Report, Alzheimer's Disease International, Jun. 2018.

[4] S. Przedborski, M. Vila, and V. Jackson-Lewis, Series introduction: Neurodegeneration—What is it and where are we? Journal of Clinical Investigation, vol. 111, no. 1, pp. 3–10, 2003. doi: https://doi.org/10.1172/JCI200317522.

[5] G. M. McKhann et al., The diagnosis of dementia due to Alzheimer's disease: Recommendations from the National Institute on Aging–Alzheimer's Association workgroups, Alzheimer's & Dementia, vol. 7, no. 3, pp. 263–269, 2011. doi: 10.1016/j.jalz.2011.03.005

[6] S. Sharma and K. Guleria, A deep learning model for early prediction of pneumonia using VGG19 and neural networks, in Proc. 3rd Int. Conf. Mobile Radio Communications & 5G Networks (MRCN), Springer, 2022, pp. 1–12. doi: 10.1007/978-981-19-7982-8_50.

[7] J. Giorgio et al., Modelling prognostic trajectories of cognitive decline due to Alzheimer's disease, NeuroImage: Clinical, vol. 26, Art. no. 102199, 2020. doi: https://doi.org/10.1016/j.nicl.2020.102199.

[8] C. Lian, M. Liu, J. Zhang, and D. Shen, Hierarchical fully convolutional network for joint atrophy localization and Alzheimer's disease diagnosis using structural MRI, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 4, pp. 880–893, 2020. doi: 10.1109/TPAMI.2018.2889096.

[9] E. Moser et al., Magnetic resonance imaging methodology, European Journal of Nuclear Medicine and Molecular Imaging, vol. 36, suppl. 1, pp. 30–41, 2009. https://doi.org/10.1007/s00259-008-0938-3.

[10] D. S. Marcus et al., Open Access Series of Imaging Studies (OASIS): Cross-sectional MRI data in young, middle-aged, nondemented, and demented older adults, Journal of Cognitive Neuroscience, vol. 19, no. 9, pp. 1498–1507, 2007. doi: https://doi.org/10.1162/jocn.2007.19.9.1498.

[11] S. Kumar, K. Guleria, and S. Tiwari, An analytical study of the COVID-19 pandemic: Fatality rate and influential factors, Design Engineering, pp. 12213–12225, 2021. doi: https://doi.org/10.1049/PBHE038E_ch4.

[12] A. Algani, Y.M. Ritonga, M. Bala, A. Ansari, M.S. Badr, M. Taloba, Machine learning in health condition check-up using Breiman's random forest algorithm, Measurement: Sensors, vol. 23, Art. no. 100406, 2022. doi: https://doi.org/10.1016/j.measen.2022.100406.

[13] S. Jindal, A. Sharma, A. Joshi, M. Gupta, Artificial intelligence fuelling healthcare, in Mobile Radio Communications and 5G Networks, Springer, Singapore, pp. 501–507, 2021. Doi: https://doi.org/10.1007/978-981-15-7130-5_40.

[14] N. K. Trivedi, A. Anand, U. K. Lilhore, and K. Guleria, Deep learning applications on edge computing, in Machine Learning for Edge Computing, CRC Press.

[15] H. Mansourifar and W. Shi, Deep synthetic minority over-sampling technique, arXiv:2003.09788, 2020. Available: http://arxiv.org/abs/2003.09788.

[16] S. Dubey, Alzheimer's dataset, 2019. Available: https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images.

[17] T. M. Ghazal and G. Issa, Alzheimer disease detection empowered with transfer learning, Computers, Materials & Continua, vol. 70, pp. 5005–5019, 2022. doi: https://doi.org/10.32604/cmc.2022.020866

[18] A. Revathi et al., Early detection of cognitive decline using machine learning and cognitive ability test, Security and Communication Networks, 2022(2):1-13, pp. 1–13, 2022. doi: https://doi.org/10.1155/2022/4190023.

[19] A. Khan and S. Zubair, Development of a three-tiered cognitive hybrid machine learning algorithm for effective diagnosis of Alzheimer's disease, Journal of King Saud University – Computer and

Information Sciences, 2022. doi: https://doi.org/10.1016/j.jksuci.2022.07.016.

[20] D. Pirrone, E. Weitschek, P. Di Paolo, S. De Salvo, M.C. De Cola, EEG signal processing and supervised machine learning to early diagnose Alzheimer's disease, Applied Sciences, vol. 12, no. 11, Art. no. 5413, 2022. doi: https://doi.org/10.3390/app12115413.

[21] M. Liu, D. Cheng, K. Wang, Y. Wang, Multi-modality cascaded convolutional neural networks for Alzheimer's disease diagnosis, Neuroinformatics, vol. 16, nos. 3–4, pp. 295–308, 2018. doi: 10.1007/s12021-018-9370-4.

[22] D. Lu, K. Popuri, G.W. Ding, R. Balachandar, M.F. Beg, Multiscale deep neural network-based analysis of FDG-PET images for early diagnosis of Alzheimer's disease, Medical Image Analysis, vol. 46, pp. 26–34, 2018. doi: 10.1016/j.media.2018.02.002.

[23] J. Islam and Y. Zhang, Brain MRI analysis for Alzheimer's disease diagnosis using an ensemble system of deep convolutional neural networks, Brain Informatics, vol. 5, no. 2, pp. 1–14, 2018. doi: 10.1186/s40708-018-0080-3.

[24] E.N. Marzban, A.M. Eldeib, I.A. Yassine, Y.M. Kadah, S.D. Ginsberg, For the Alzheimer's Disease Neurodegenerative Initiative (2020) Alzheimer's disease diagnosis from diffusion tensor images using convolutional neural networks, PLoS One, vol. 15, no. 3, Art. no. e0230409, 2020. doi: https://doi.org/10.1371/journal.pone.0230409.

[25] N. Deepa and S. P. Chokkalingam, Optimization of VGG16 using arithmetic optimization algorithm for early detection of Alzheimer's disease, Biomedical Signal Processing and Control, vol. 74, Art. no. 103455, 2022. https://doi.org/10.1016/j.bspc.2021.103455.

[26] S. Sharma, K. Guleria, S. Tiwari, and S. Kumar, A deep learning-based CNN model with VGG16 feature extractor for Alzheimer's disease detection using MRI scans, Measurement: Sensors, vol. 24, Art. no. 100506, 2022. https://doi.org/10.1016/j.measen.2022.100506.

[27] S. Murugan et al., DEMNET: A deep learning model for early diagnosis of Alzheimer's disease and dementia from MR images, IEEE Access, vol. 9, pp. 90319–90329, 2021. doi: 10.1109/ACCESS.2021.3090474.

[28] M. To˘gaçar, B. Ergen, Z. C¨omert, F. ¨Ozyurt, A deep feature learning model for pneumonia detection, IRBM, vol. 41, no. 4, pp. 212–222, 2020. doi: 10.1109/ACCESS.2021.3090474.

[29] P. Bansal, A. Vanjani, A. Mehta, J.C. Kavitha, S. Kumar, Improving melanoma classification accuracy using binary Harris hawks optimization, Soft Computing, vol. 26, no. 17, pp. 8163–8181, 2022. doi: https://doi.org/10.1007/s00500-022-07234-1.

[30] S. Gupta, P. Bansal, S. Kumar, ULBP-RF: A hybrid approach for malware image classification, in Proc. 5th Int. Conf. Parallel, Distributed and Grid Computing (PDGC), 2018. doi: 10.1109/PDGC.2018.8745989.

[31] K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv:1409.1556, 2014.

[32] J. Pardede, B. Sitohang, S. Akbar, and M. L. Khodra, Transfer learning using VGG16 for fruit ripeness detection, International Journal of Intelligent Systems and Applications, vol. 13, no. 2, pp. 52–61, 2021. doi: 10.5815/ijisa.2021.02.04.

[33] P. Xu, J. Zhao, J. Zhang, Identification of intrinsically disordered protein regions using deep neural networks, Algorithms, vol. 14, no. 4, Art. no. 107, 2021. doi: 10.3390/a14040107.

[34] A. De and A. S. Chowdhury, DTI-based Alzheimer's disease classification with rank-modulated fusion of CNNs and random forest, Expert Systems with Applications, vol. 169, Art. no. 114338, 2021. doi: 10.1016/j.eswa.2020.114338.

[35] E.K. Fanar, O. Bayat, A.D. Duru, Diagnosis of Alzheimer's disease using 2D MRI slices by CNN, Applied Bionics and Biomechanics, Art. no. 6690539, 2021. doi: 10.1155/2021/6690539.

[36] T. Pan, J. Zhao, W. Wu, and J. Yang, Learning imbalanced datasets based on SMOTE and Gaussian distribution, Information Sciences, vol. 512, pp. 1214–1233, 2020. doi: 10.1016/j.ins.2019.10.048.

[37] S. Ahmed et al., Ensembles of patch-based classifiers for Alzheimer's disease diagnosis, IEEE Access, vol. 7, pp. 73373–73383, 2019. doi: 10.1109/ACCESS.2019.2920011.

[38] D. Stamate et al., Applying deep learning to predicting dementia and mild cognitive impairment, in Artificial Intelligence Applications and Innovations, IFIP AICT, vol. 584, Springer, 2020, pp. 308–319. doi: 10.1007/978-3-030-49186-4_26.

[39] R. C. Petersen et al., Alzheimer's Disease Neuroimaging Initiative (ADNI): Clinical characterization, Neurology, vol. 74, no. 3, pp. 201–209, 2010. doi: 10.1212/WNL.0b013e3181cb3e25

Two-Stage Anomaly Detection in Super High-Rise Fire Sensor Networks Using Low-Rank Completion and RV-GAN

Congyue Qi[1], Qinghua Tan[1], Ji Liao[2], Lijun Yuan[2*]
[1]School of Civil Engineering, Tsinghua University, Beijing 100084, China
[2]The Third Construction Co., Ltd of China Construction Third Engineering Bureau, Wuhan 437000, China
E-mail: cscecqcy@vip.qq.com, qinghuatan@gxu.edu.cn, 517943238@qq.com, 15623783703@163.com
[*]Corresponding author

Keywords: super high-rise building, fire monitoring, sensor data anomaly, low-rank completion, generative adversarial network

Received: September 3, 2025

With the increasing intelligence of fire protection systems in super high-rise buildings, fire monitoring schemes based on sensor networks have raised higher requirements in terms of perception accuracy and response efficiency. However, complex spatial environments and instability in the performance of sensing equipment often led to anomalies such as missing, drifting, and distorted data collection. To address these challenges, this study develops a two-stage anomaly detection framework based on low-rank completion and generative adversarial learning. In the first stage, the Correlation-Regularized Schatten-p Precompletion (CRSP-Pre) model leverages feature correlations and low-rank constraints to structurally repair abnormal or missing samples. In the second stage, a Residual-Variational Generative Adversarial Network (RV-GAN) is designed to capture high-order and composite anomalies through variational generation and residual discrimination mechanisms. Experimental results show that when the data missing rate reaches 50%, the mean absolute error (MAE) and structural similarity index (SSIM) are 0.131 and 0.671, respectively-representing improvements of approximately 15% and 12% over conventional completion and single-stage detection methods. When the number of nodes is 40, the false alarm rate of the proposed method is 5.4%, the model parameter size is 7.8M, and the inference time is 22.4 ms. The results show that the method has good robustness and engineering adaptability, effectively enhancing the processing capability and response reliability of fire monitoring systems for abnormal data. It provides a technical path for intelligent management of sensing data in complex building environments.

Povzetek: Študija razvije metodo za bolj zanesljivo zaznavanje in popravljanje napak v senzorskih podatkih požarnega nadzora v visokih stavbah.

1 Introduction

With the continuous improvement of intelligence and automation management in super high-rise buildings, fire monitoring systems based on multi-source perception are becoming a key support for ensuring the safe operation of buildings [1]. Fire monitoring sensors are widely deployed inside buildings to achieve real-time collection and early warning response of key indicators such as smoke, temperature, and carbon monoxide concentration, effectively supporting intelligent linkage tasks such as fire detection, evacuation control, and disaster disposal [2, 3]. However, under the influence of multiple factors such as complex spatial layout, high-frequency data flow, and unstable communication, sensor systems are prone to issues such as data drift, loss, or abnormal mutations, which in turn affect the stability and accuracy of judgment of the overall fire monitoring system [4, 5]. To address these challenges, current research focuses on using computer intelligent algorithms to precisely identify and efficiently repair abnormal sensor data, thereby enhancing the robustness and perception reliability of the monitoring system. In this context, numerous scholars have conducted a series of studies on the construction and model optimization of algorithms for detecting sensor data anomalies. Gursel et al. proposed a generative adversarial anomaly detection model incorporating physical information to address anomalies caused by sensor errors or human errors in nuclear power plants. This method utilized the known physical characteristics of the system to constrain sensor output, effectively improving the credibility and accuracy of anomaly detection, and performing better in anomaly identification of sensor and manual monitoring data [6]. Wang et al. proposed a distributed detection method based on adaptive grouping to address issues such as neglecting inter-node correlation and high communication overhead in wireless sensor network anomaly detection. This scheme constructed node groups through graph theory analysis, performed local detection combined with edge cloud architecture, and introduced a

consistency evaluation mechanism to enhance detection reliability [7]. Ahmad et al. proposed a lightweight unsupervised anomaly detection method combining continuous wavelet transform and support vector clustering to address issues such as high computational complexity and lack of data annotation in real-time sensor data, demonstrating good adaptability to data streams [8]. Su et al. constructed an anomaly detection model integrating graph attention mechanism and temporal convolution to address the challenges of complex inter-time series correlation and scarce abnormal samples in wireless sensor networks. This method jointly modeled the potential relationships between feature dimensions and time dimensions, effectively alleviating the problem of extreme imbalance between positive and negative samples [9].

Based on this, some scholars have expanded their research perspectives to the optimization of IoT architecture and cross-domain anomaly detection, further enhancing the adaptability and intelligence of sensor data processing in complex scenarios. Allka et al. addressed the issue of data reliability brought by low-cost IoT sensors in air quality monitoring networks, proposing a cyclic autoencoder anomaly detection structure that integrates temporal and spatial correlations. This method involved not only the temporal characteristics of the sensors themselves but also the spatial relationships between adjacent devices in the detection network for reconstruction [10]. Pan et al. tackled the problems of input data distribution differences and sensor redundancy dependencies in multivariate time series anomaly detection, introducing a self-attention network method that incorporates a dual masking mechanism. This method enhanced the adaptability of the predictor to abnormal inputs through block-level masking and introduces a maximum masking strategy to reduce misleading effects from unrelated sensors [11]. Gutierrez-Rojas et al. constructed a multi-module anomaly detection model for industrial IoT systems supported by wireless sensors. This structure integrated three key processes: data collection, fusion, and decision-making, and introduced interpretable artificial intelligence algorithms to enhance anomaly analysis capabilities [12]. Prabowo et al. designed an automatically operable sensor anomaly detection system to address the issue of manual processing required for sensor data anomalies in IoT platforms. This system was integrated into a general IoT architecture, enabling intelligent anomaly elimination from data collection to preprocessing stages [13]. To position the proposed framework within existing research. Table 1 presents a summary of key related works covering diverse sensor-network anomaly detection methods.

Table 1: Summary of representative related works.

Reference	Task/Domain	Method Category	Core Idea and Contribution
Gursel et al. [6]	Anomaly detection in reactor sensor data	Physics-informed GAN model	Integrates human-error and sensor-fault modeling within a unified system-level anomaly detection framework.
Wang et al. [7]	Anomaly detection in wireless sensor networks (WSN)	Correlation-aware distributed detection	Utilizes inter-node correlation for group-based distributed detection, enhancing reliability in WSNs.
Ahmad et al. [8]	Lightweight anomaly detection in sensors	Continuous wavelet transform + vector clustering	Employs efficient feature extraction and clustering for resource-constrained sensor devices.
Su et al. [9]	WSN data-stream anomaly detection	Joint optimization of GATv2 and TCN	Combines spatial topology learning (GATv2) and temporal modeling (TCN) for joint spatiotemporal representation.
Allka et al. [10]	IoT-based sensor-network anomaly detection	Recurrent autoencoder (RAE)	Leverages spatiotemporal correlations through sequence reconstruction, suitable for large-scale IoT platforms.
Pan et al. [11]	Multivariate time-series anomaly detection	Dual-mask self-attention network (DUMA)	Applies block-masking and max-masking strategies to enhance adaptability to abnormal inputs.
Gutierrez-Rojas et al. [12]	Anomaly detection in WSN-enabled cyber-physical systems (CPS)	Interpretable multi-module framework	Integrates data collection, fusion, and decision stages, emphasizing interpretability and modularity.
Prabowo et al. [13]	Environmental-sensor anomaly detection in IoT platforms	End-to-end anomaly-elimination pipeline	Proposes a cloud–edge pipeline for large-scale IoT anomaly detection with high deployment practicality.

In summary, under the complex monitoring environment of super high-rise buildings, sensor data still generally faces bottleneck issues such as diverse types of anomalies, low accuracy in anomaly detection, and unstable data recovery. To address this, a two-stage integrated method that combines low-rank data completion and deep generative detection is designed, aiming to enhance the overall performance of fire sensor

systems in anomaly perception and error recovery. Specifically, it includes two innovations: Firstly, the CRSP-Pre is proposed to achieve structured recovery and information reconstruction of anomalous samples. Secondly, the RV-GAN is constructed to achieve precise identification of high-order anomalies, enhancing the model's responsiveness and practical deployment adaptability in extreme event monitoring.

Accordingly, the core research question of this study is: Can a two-stage approach, which seamlessly combines low-rank completion and generative adversarial learning, substantially enhance anomaly-detection performance under conditions of high missing rates, all the while preserving computational efficiency? This question forms the foundation of the methodological design described in Section 2.

2 Methods and materials

2.1 Anomaly preprocessing based on feature correlation and low-rank completion

In the intelligent fire protection system of super high-rise buildings, sensor data often experiences missing values and noise due to channel interference or node failures, affecting the stability of subsequent anomaly detection [14]. To address this, a correlation-driven data completion model, CRSP-Pre, is proposed. It combines spatiotemporal correlation modeling and weighted low-rank constraints to achieve high-quality restoration of fire monitoring data and preliminary screening of outliers. The overall process is illustrated in Figure 1.

As shown in Figure 1, CRSP-Pre takes raw fire monitoring data as input. In the first stage, it completes preliminary completion by measuring the correlation between historical and neighboring sensors using the Pearson Correlation Coefficient (PCC). In the second stage, it reconstructs the underlying low-rank structure through joint constraints of the weight matrix and Schatten-p norm, enhancing the perception ability for outliers under sparse optimization. During the optimization process, the Alternating Direction Method of Multipliers (ADMM) algorithm is used for iterative solving, achieving precise recovery of missing values and potential outliers, providing reliable data support for subsequent detection.

The core objective of the first stage is to perform preliminary interpolation and completion of missing data based on the correlation between sensors, as illustrated in Figure 2.

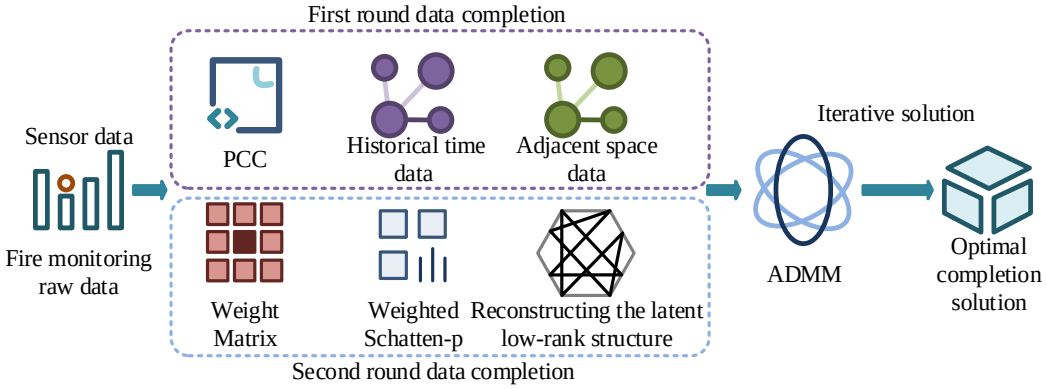


Figure 1: CRSP-Pre model structure for anomaly repair.

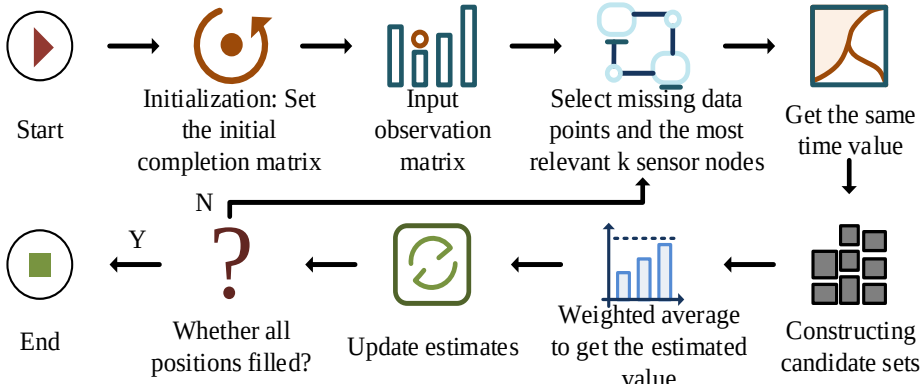


Figure 2: PCC-based initial completion algorithm process.

As shown in Figure 2, with the observation matrix X as input, for each missing data point, the most relevant k sensor nodes are selected in turn to obtain the values at the same time, forming a candidate set, and the estimated value is obtained through weighted average. The completion process is iteratively applied to all missing points until all positions are filled or the maximum iteration count is reached. First, let the original

data matrix be denoted as X, and the mask matrix as $M \in \{0,1\}^{m\times n}$, where $M_{ij}=1$ indicates that the corresponding item x_{ij} has an observed value. Otherwise, it equals 0, indicating a missing value. The initial completion matrix is denoted as $\hat{X}^{(0)}=X$. For each missing item x_{tj}, the correlation with other sensor nodes is calculated using PCC. The correlation coefficient between node i and node j is denoted as equation (1).

$$\rho_{ij}=\frac{\sum_{t=1}^{m}(x_{ti}-\mu_i)(x_{tj}-\mu_j)}{\sqrt{\sum_{t=1}^{m}(x_{ti}-\mu_i)^2}\cdot\sqrt{\sum_{t=1}^{m}(x_{tj}-\mu_j)^2}} \tag{1}$$

In equation (1), μ_i and μ_j represent the mean values of node i and node j in the time dimension. ρ_{ij} denotes the PCC between node i and node j in the historical time series. Based on the absolute value of ρ_{ij}, the most relevant sensor set is constructed for each target node j. Then, a subset of nodes with valid observations at time t is selected, and an estimation expression for missing values is defined accordingly, as shown in equation (2).

$$\hat{x}_{tj}^{(1)}=\frac{\sum_{i\in N_k^+(j)}|\rho_{ij}|\cdot x_{ti}}{\sum_{i\in N_k^+(j)}|\rho_{ij}|} \tag{2}$$

In equation (2), $\hat{x}_{tj}^{(1)}$ represents the estimation result of missing value x_{tj} during the initial completion stage. If sufficient neighbor node observations are missing at some points, a historical adjacent time slice backtracking mechanism can be introduced as a supplement. After completing the first stage of estimation and completion process, the preliminary filled matrix $\hat{X}^{(1)}=PCC-Interpolate(X,M,k)$ is obtained. This matrix will be further fed into the low-rank completion framework in the second stage to achieve structural anomaly identification and optimized recovery.

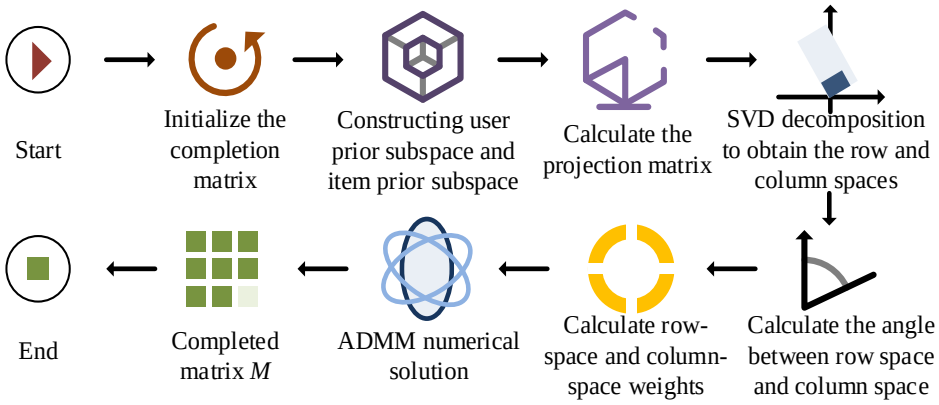


Figure 3: Weighted Schatten-p norm-based data refinement process.

To enhance the structural consistency and anomaly discriminability of the completed matrix, the weighted Schatten-p (0<p<1) norm is introduced in the second stage as a closer alternative to the matrix rank. It alleviates the excessive contraction of large singular values and better preserves the principal subspace under high missing rates. Although the non-convex form increases the optimization complexity, it can be effectively solved using ADMM with iteratively reweighted singular value thresholding (IRLS), thus achieving a good balance between reconstruction accuracy and computational efficiency [15]. The overall process of the second stage is shown in Figure 3.

In Figure 3, the completion matrix is first projected into predefined user subspace and sensor subspace. Subsequently, the angle between the two subspaces is calculated and the weight coefficient is constructed. On this basis, the weighted Schatten-p norm is introduced to construct the low-rank optimization objective function, and the numerical solution is obtained with the help of ADMM. Finally, the optimized completion result is output. Assuming the output matrix of the first stage is Z, the optimization objective is to find a low-rank sparse matrix M to minimize its weighted Schatten-p norm, as shown in equation (3).

$$\|M\|_{S_p}^{p}=\sum_{i'=1}^{\min(m,n)}\sigma_{i'}^{p} \tag{3}$$

In equation (3), σ_i is the i' th singular value of matrix M, and $p\in(0,1]$ controls the strength of regular sparsity. To incorporate structural guidance information, the study constructs a prior user subspace $\hat{U}_r$ and a sensor subspace $\hat{V}_r$, and measures the angle between the current subspace and the prior subspace. To construct the prior subspaces U_0 and V_0, the coarsely completed matrix X_0 obtained via PCC–Interpolate is decomposed using truncated SVD to extract the dominant column and row subspaces, representing user and sensor priors, respectively. These subspaces capture the major spatial–temporal correlation patterns in the observed data. In

cases without prior knowledge, U_0 and V_0 are initialized from random orthogonal bases of the same rank. The subspace angles are then calculated as shown in equation (4).

$$\theta_u = \arccos\left(\frac{\langle U, \hat{U}_r \rangle}{\|U\|_F \cdot \|\hat{U}_r\|_F}\right), \theta_v = \arccos\left(\frac{\langle V, \hat{V}_r \rangle}{\|V\|_F \cdot \|\hat{V}_r\|_F}\right) \quad (4)$$

In equation (4), U and V represent the left and right singular vector matrices obtained after performing singular value decomposition on the matrix. M denotes the Frobenius inner product, and $\langle \cdot, \cdot \rangle$ denotes the Frobenius norm. Subsequently, the angle value is converted into a directional weight through a mapping function, as shown in Equation (5).

$$w_u = f(\theta_u), \quad w_v = f(\theta_v) \quad (5)$$

In equation (5), θ_u and θ_v represent the angles between the current subspace and the user's prior subspace, as well as the sensor's prior subspace, respectively. Ultimately, a Schatten-p norm minimization model with bidirectional weight factors is constructed, as shown in equation (6).

$$\begin{aligned} &\min_{M} \quad \lambda_u w_u \|M\|_{S_p} + \lambda_v w_v \|M\|_{S_p} \\ &s.t. M_{ij} = Z_{ij}, \forall (i'', j) \in \Omega \end{aligned} \quad (6)$$

In equation (6), Z_{ij} represents the corresponding observed value in the initial completion matrix (output matrix of the first stage). Ω is the set of observed data positions. λ_u and λ_v are regularization strength adjustment factors in two directions. (i'', j) denotes the position corresponding to the i'' th time record and the j th sensor node in the data matrix.

Due to the presence of a non-convex Schatten-p norm term in the objective function and projection constraints, ADMM is introduced for optimization and solution. This algorithm equivalently transforms the problem into a function with Lagrange multipliers in the following form by introducing auxiliary variables, as shown in Equation (7).

$$L_\eta(M, Y, A) = f(M) + g(Y) + \langle A, M - Y \rangle + \frac{\eta}{2}\|M - Y\|_F^2 \quad (7)$$

In equation (7), η is the penalty factor of ADMM. A is the multiplier matrix. The algorithm gradually converges to a low-rank matrix M^* by alternately minimizing each variable and updating the multiplier. To ensure stability and convergence of the optimization process, the ADMM iteration adopts an adaptive penalty parameter and standard stopping criteria. The penalty parameter η is initialized to 0.1 and updated by a factor of 1.5 every 20 iterations. The optimization terminates when both primal and dual residuals fall below 10−4, or when 300 iterations are reached. The penalty parameter η is initialized to 0.1 and adaptively updated by a factor of 1.5 every 20 iterations. The selection and sensitivity of η are further analyzed in Section 3.1, where the parameter range 0.1-1.0 was tested to confirm convergence stability and accuracy.

2.2 High-order anomaly detection method based on deep generative models

After completing the preprocessing of sensor data based on feature correlation and low-rank structure constraints, abnormal information can be repaired and filled to a certain extent. However, in the complex spatial environment of super high-rise buildings, sensor data fluctuations induced by fires often exhibit nonlinear mutations or multidimensional covariant patterns [16, 17]. To this end, the study proposes a deep Residual Variational Generative Adversarial Network (RV-GAN), which achieves precise identification and differentiation of high-order anomalies by jointly modeling the potential spatial distribution and data generation discrimination mechanism. Its structure is shown in Figure 4.

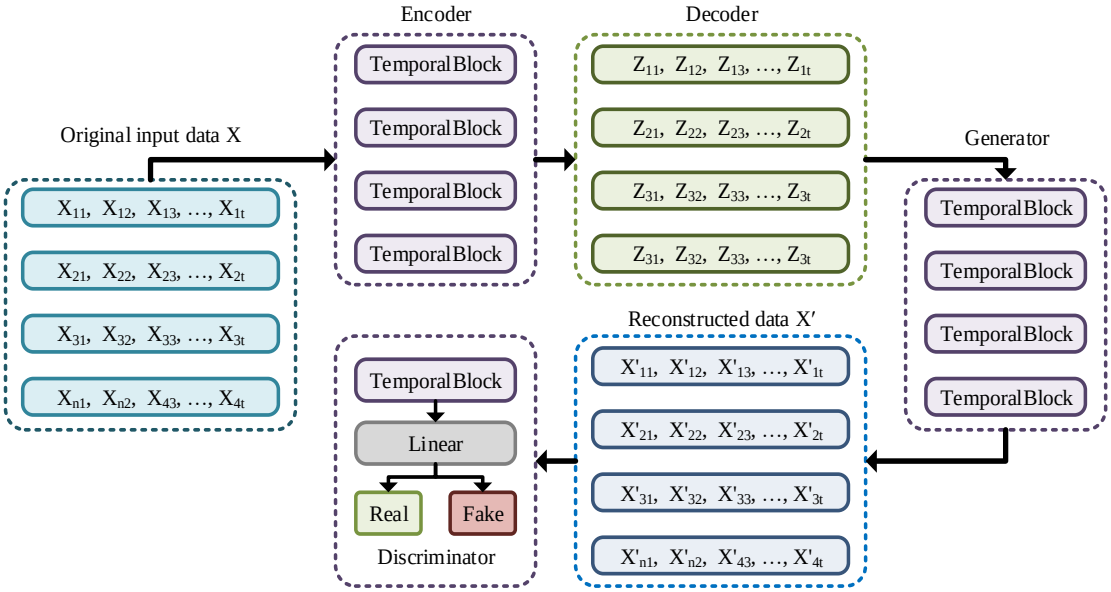


Figure 4: RV-GAN model architecture for anomaly detection.

As shown in Figure 4, the RV-GAN model comprises four main modules: Encoder, Decoder, Generator, and Discriminator, all of which are embedded with temporal convolutional modules to extract multi-scale dynamic features. Specifically, the Encoder maps time series samples to latent variable space, the Decoder reconstructs input data based on the reparameterization strategy of the Variational Auto-Encoder (VAE), the Generator further fits high-fidelity pseudo-samples through a residual convolutional structure, and the Discriminator combines the Wasserstein distance and a linear projection classifier to determine whether the samples originate from the original data distribution, thus achieving robust discrimination of potential anomalies.

To further clarify the network design, the generator in RV-GAN consists of five convolutional blocks and two transposed convolution layers for upsampling. Each convolution uses a 3×3 kernel, stride 1, and LeakyReLU activation (α=0.2), with Tanh used in the output layer to constrain reconstruction values. Batch normalization is applied after each convolution to stabilize training. The residual convolution module contains three residual units with 3×3 kernels and skip connections to enhance feature preservation and gradient flow. The generator and decoder are independently parameterized to prevent gradient interference, while the discriminator follows a PatchGAN-style architecture with progressively increasing channel depths (64–128–256–512).

To enhance the ability to identify high-order potential anomalies, the RV-GAN model adopts a structure that combines deep variational encoding mechanisms with adversarial learning strategies to achieve implicit modeling and generative discrimination of anomalous samples. The input multi-dimensional sensor time series are denoted as $S=[s_1,s_2,...,s_T]$, where T denotes the length of the sampling time window and D denotes the number of sensors or dimensions. The encoder module, based on a temporal convolutional network, encodes the original sequence into distribution parameters in the latent space, including the mean vector μ and the logarithmic variance vector $\log\sigma^2$, with the mapping relationship expressed as Equation (8).

$$\mu,\log\sigma^2=G_e(S) \tag{8}$$

In equation (8), $G_e(\cdot)$ represents the encoder network function, which outputs parameters used to construct the distribution of latent variables. Subsequently, based on the reparameterization technique proposed in VAE, the latent vector z is sampled from the Gaussian distribution using the sampling method shown in equation (9).

$$z=\mu+\sigma\otimes\grave{o},\quad \grave{o}\sim N(0,I) \tag{9}$$

In equation (9), $\otimes$ represents the element-wise multiplication operation, $\grave{o}$ is the standard normal distribution noise vector, $\sigma=\exp(0.5\cdot\log\sigma^2)$ represents the standard deviation vector, ensuring the differentiability of the sampling. The latent variable z is fed into the decoder $G_d(\cdot)$ and the residual generator $G_r(\cdot)$ to obtain the reconstructed sample and the generated sample, respectively, as expressed in equation (10).

$$\hat{S}=G_d(z),\quad \bar{S}=G_r(z) \tag{10}$$

In equation (10), $\hat{S}$ and $\bar{S}$ represent the reconstructed sample and generated sample, respectively. The model introduces a discriminator $D(\cdot)$, using the Wasserstein distance as the optimization objective between generation and discrimination. The overall training objective function consists of three parts. The first is the reconstruction loss, which is composed of the VAE reconstruction error term and the KL divergence term, as shown in equation (11) [18].

$$L_{VAE}=E_{q(z|S)}[\left\|S-\hat{S}\right\|_2^2]+\beta KL(q(z|S)\|p(z)) \tag{11}$$

In equation (11), the first term represents the mean squared reconstruction error, while the second term represents the KL divergence between the latent variable distribution $q(z|S)$ and the standard normal distribution $p(z)$, serving as a regularization constraint. The balance between these two components is controlled by a weighting coefficient β, which follows the design of β-VAE. In this study, β is empirically set to 0.5 to ensure a proper trade-off between reconstruction fidelity and latent compactness. Sensitivity analysis confirms that the performance is stable for $\beta\in[0.3, 0.7]$, and dynamic scheduling of β is unnecessary. Secondly, the adversarial loss based on the Wasserstein distance is introduced to train the game relationship between the generator and the discriminator, which is defined as equation (12) [19, 20].

$$L_{WGAN}=E_{\bar{S}\sim G_r(z)}[D(\bar{S})]-E_{S\sim P_r}[D(S)]+\lambda\cdot E_{\hat{S}\sim P_{\hat{S}}}[(\left\|\nabla_{\hat{S}}D(\hat{S})\right\|_2-1)^2] \tag{12}$$

In equation (12), P_r represents the true sample distribution, and $P_{\hat{S}}$ is a mixed distribution obtained by linear interpolation between the true samples and the generated samples. λ denotes the gradient penalty coefficient, ensuring that the discriminator satisfies the Lipschitz condition. The gradient penalty term in equation (12) is implemented with reference to WGAN-GP. The gradient is calculated at the linear interpolation points between the real samples and the generated samples, see equation (13).

$$\hat{S}=\grave{o}S+(1-\grave{o})\tilde{S},\grave{o}\sim U(0,1) \tag{13}$$

As shown in Equation (13), during training, the discriminator and generator are updated alternately with a ratio of 5:1, and the Adam optimizer (learning rates 1e-4 and 2e-4) is used respectively. This configuration exhibits stable convergence characteristics in

experiments and avoids the problems of gradient explosion and mode collapse.

Ultimately, the joint loss function of RV-GAN is the weighted sum of the above two parts. During the training process, model parameters are iteratively optimized through backpropagation and gradient clipping strategies to ensure stable convergence and discriminative ability under different abnormal patterns. To visually demonstrate the specific execution logic of RV-GAN in the task of anomaly detection for super high-rise building fire sensor data, the process framework diagram is shown in Figure 5.

In Figure 5, the left side represents the training path, where normal data are encoded to extract latent vectors, and the Wasserstein Generative Adversarial Network (WGAN) composed of a generator and a discriminator accomplishes the adversarial optimization. The right side represents the testing path, where mixed data is input, decoded, and reconstructed by a Variational Autoencoder (VAE). The system determines the degree of abnormality based on the reconstruction error and the discrimination results. To comprehensively describe the complete process of the research in the task of anomaly detection for fire sensors in super high-rise buildings, the research integrates feature correlation modeling, low-rank completion, and generative anomaly recognition in an end-to-end manner, constructing a two-stage scoring process as shown in Figure 6.

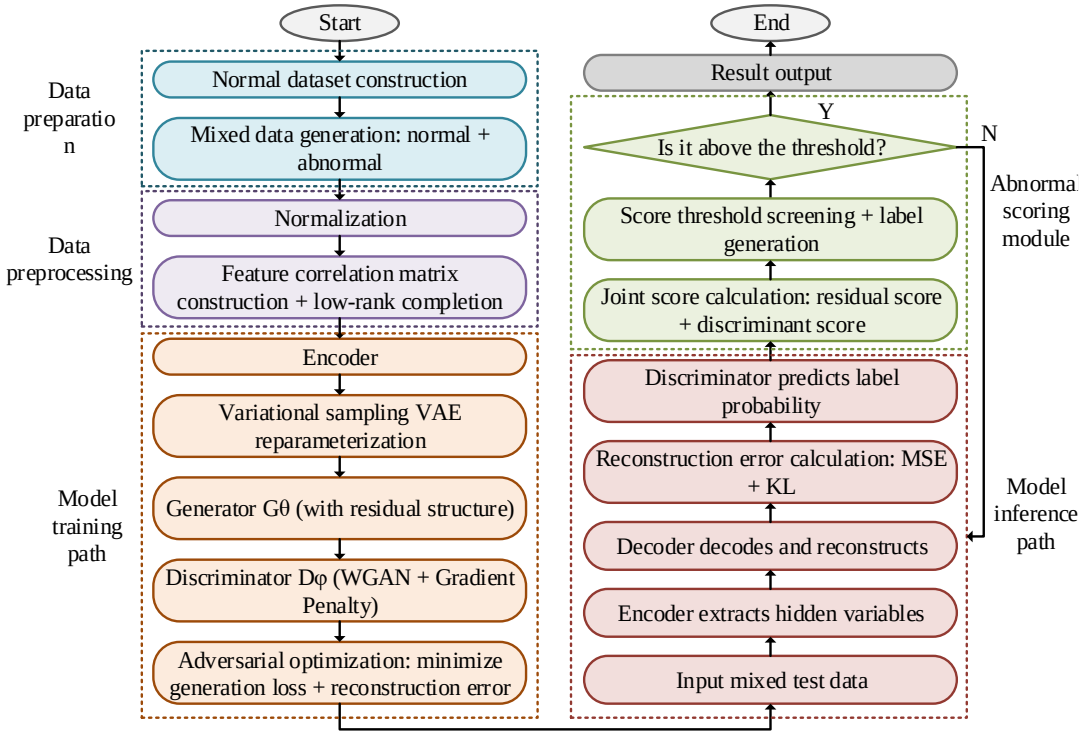


Figure 5: Workflow of RV-GAN in fire sensor anomaly detection task.

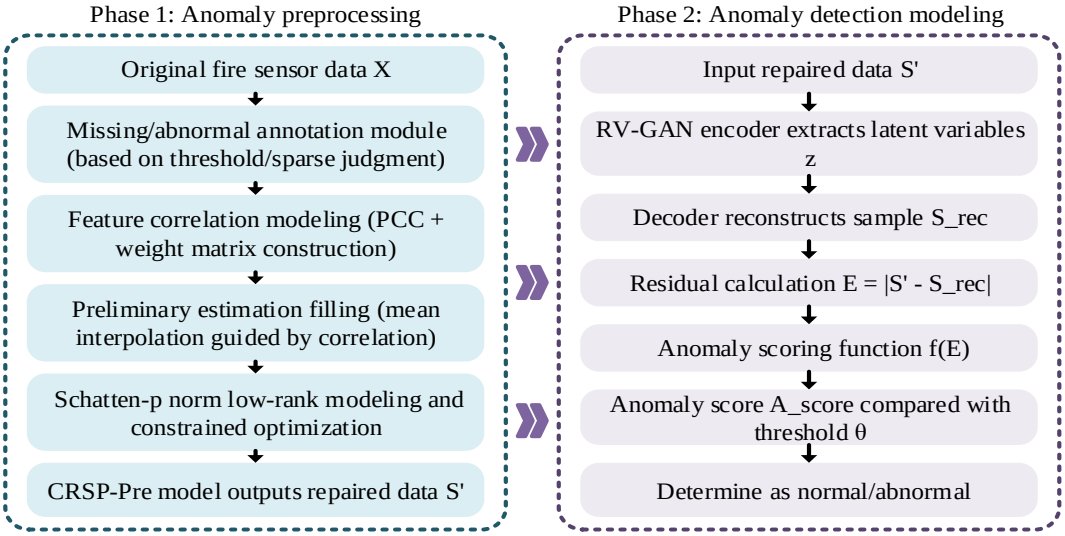


Figure 6: Two-stage anomaly detection process for super high-rise fire sensor data.

As illustrated in Figure 6, the overall workflow operates through two consecutive and cooperative stages. In the first stage, the CRSP-Pre module performs data annotation, correlation-based interpolation, and low-rank reconstruction to repair missing or abnormal sensor readings, producing a structurally complete dataset. In the second stage, the RV-GAN module receives the repaired data and conducts high-order anomaly recognition through feature encoding, sample reconstruction, and residual scoring to determine anomaly levels. Together, the two stages form an integrated framework in which the first stage ensures data completeness, while the second provides accurate semantic discrimination, enabling reliable and robust anomaly detection in super high-rise fire monitoring systems.

3 Results

3.1 Performance evaluation of CRSP-Pre abnormal data repair

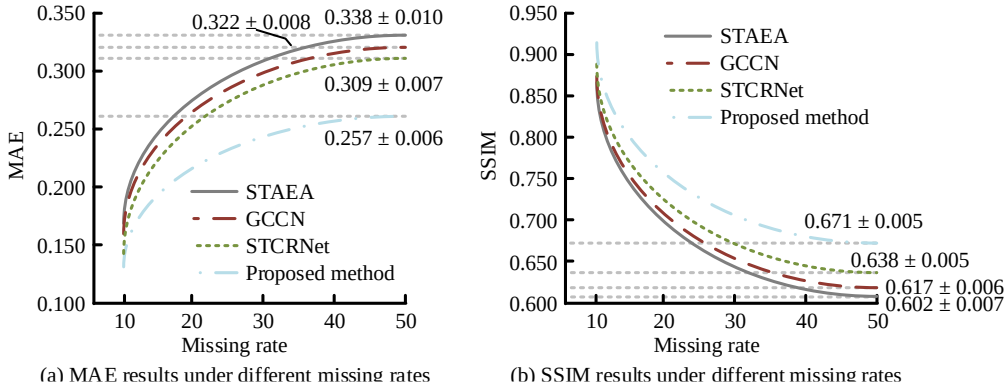
The study constructed an experimental platform for anomaly detection in sensor data from super high-rise buildings, based on Python 3.10 and PyTorch 2.0.1. The operating environment included an Intel Core i9-12900K processor, 128GB of memory, and an Ubuntu 20.04 operating system. Model training and inference were completed on an NVIDIA RTX 3090 GPU. The experiment utilized the IBRL wireless sensor dataset to evaluate the robustness of the model under standard environmental perception and complex distribution scenarios. To ensure reproducibility and prevent data leakage, both datasets were divided into training, validation, and testing subsets using a time-based non-overlapping split. Specifically, the first 70% of the temporal sequence was used for model training, the following 15% for validation, and the remaining 15% for testing. To assess the stability and convergence efficiency of the proposed model under different parameter configurations, the Schatten-p order p, regularization weight combination (λ_u, λ_v), and ADMM optimization penalty parameter η were set as core variables for a three-factor combination test. A sparse observation matrix with a missing rate of 15% was constructed, and the parameters in the initial completion stage were kept fixed, focusing on exploring the impact of sparse matrix optimization in the second stage on final performance. Each parameter combination was repeated five times, and the results are averaged. The results are presented in Table 2.

From Table 2, as the p value increased from 0.1 to 1.0, the performance of the model in terms of mean squared error and relative error continued to improve. Higher orders contributed to enhancing the low-rank representation ability of matrix completion. When the regularization weight (λ_u, λ_v) was set at a moderate level, the model achieved a better balance between accuracy and convergence. An overly small regularization coefficient limited the sparse structure's ability to model anomalies, while an excessively large one tended to lead to overfitting. Taking a larger value for the penalty parameter η could accelerate the iterative convergence process while ensuring accuracy. The combination of p =1.0, (λ_u, λ_v) =0.5, and η =10.0 was ultimately determined as the optimal parameter configuration for the model, which was used for unified scheduling and execution of each module in subsequent experiments.

To verify the model's ability to recover sensor data under high missing rates and maintain structural fidelity, the missing rate range was set from 10% to 50%, examining the model's reconstruction performance under different degrees of data sparsity. Three typical models in the current field were selected for comparison, including Spatio-Temporal Attention Embedded Autoencoder (STAEA), Graph Convolutional Completion Network (GCCN), and Spatio-Temporal Correlated Recurrent Network (STCRNet), representing attention mechanisms, graph modeling, and temporal modeling structures, respectively. The comparative evaluation metrics used were Mean Absolute Error (MAE) and Structural Similarity Index (SSIM). Each experiment was independently repeated five times with different random initializations, and the reported results represent the mean ± standard deviation to ensure statistical reliability. The corresponding experimental results are shown in Figure 7.

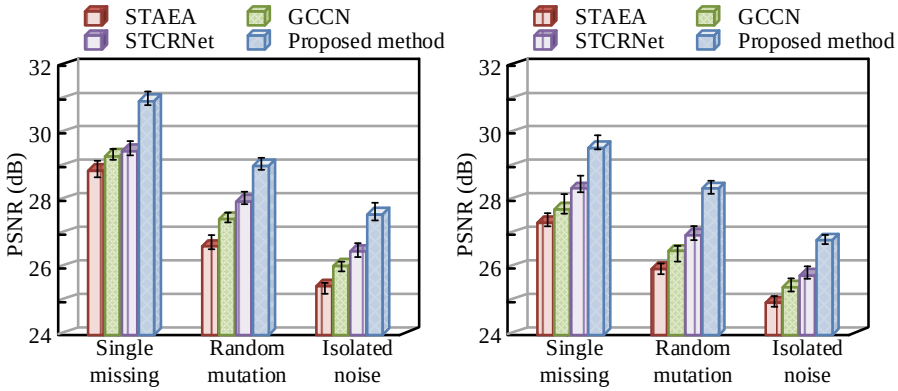
Table 2: Parameter sensitivity test results.

Schatten-p order p	Regularization weights (λ_u, λ_v)	ADMM penalty parameter η	Mean Squared Error ↓	Mean Absolute Relative Error (%) ↓	Convergence rounds ↓	Average runtime (s) ↓
0.1	(0.1, 0.1)	0.1	0.1145	8.91	36	4.82
0.1	(0.5, 0.5)	1.0	0.0932	6.84	28	3.75
0.1	(1.0, 1.0)	10	0.0879	6.29	25	3.51
0.5	(0.1, 0.1)	1.0	0.0723	5.32	24	3.11
0.5	(0.5, 0.5)	1.0	0.0649	4.88	27	3.46
0.5	(0.5, 0.5)	10	0.0622	4.70	22	3.02
1	(0.1, 0.1)	1.0	0.0578	4.51	21	2.97
1	(0.5, 0.5)	1.0	0.0547	4.33	19	2.91
1	(0.5, 0.5)	10	0.0535	4.29	16	2.38



(a) MAE results under different missing rates (b) SSIM results under different missing rates

Figure 7: Model performance test under different missing rates. (Values represent the mean ± standard deviation (SD) over five independent runs.)



(a) PSNR comparison under point anomaly conditions (b) PSNR comparison under clustered anomaly conditions

Figure 8: Comparison of recovery performance under different anomaly types.

In Figure 7(a), when the missing rate was 10%, the MAE of STAEA, GCCN, STCRNet, and the proposed method were 0.167 ± 0.006, 0.155 ± 0.005, 0.143 ± 0.004, and 0.131 ± 0.004. At a 50% missing rate, they increased to 0.338 ± 0.010, 0.322 ± 0.008, 0.309 ± 0.007, and 0.257 ± 0.006. In Figure 7(b), when the missing rate was 10%, the SSIM of the four methods were 0.861 ± 0.005, 0.873 ± 0.006, 0.889 ± 0.005, and 0.907 ± 0.004. When the missing rate increased to 50%, they decreased to 0.602 ± 0.007, 0.617 ± 0.006, 0.638 ± 0.005, and 0.671 ± 0.005. The proposed method in this study combined the encoder and multi-scale residual reconstruction module during the modeling process, effectively preserving the spatiotemporal structural information of the data. In contrast, although STAEA introduced a spatiotemporal attention mechanism, it ignored the continuous aggregation of neighboring dependency information. Although GCCN and STCRNet could model graph structures or cyclic relationships, they suffered from severe breaks in the information propagation chain under high missing rates, making it difficult to support stable reconstruction. Therefore, their overall performance was slightly inferior.

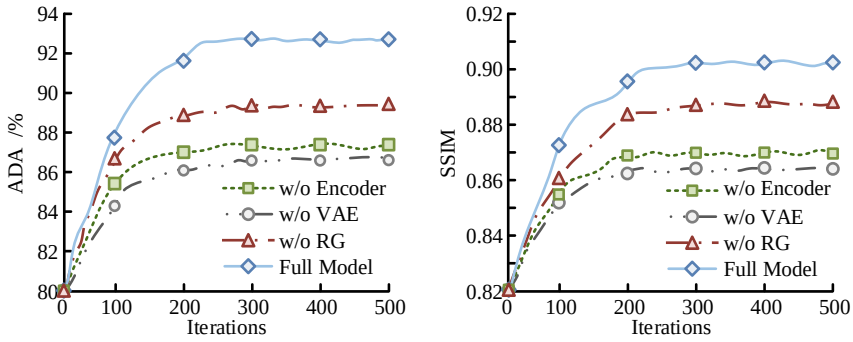
Subsequently, the abnormal data in the test set were divided into two categories. Point-like anomalies simulated isolated sensor failures, encompassing three types of perturbations: single-point missing, single-point offset, and single-point noise. Cluster-like anomalies simulated dense collection anomalies, covering three typical scenarios: local missing, regional outbreak, and block drift. Under each abnormal condition, the peak signal-to-noise ratio (PSNR) and reconstruction stability index (RSI) of the model output were measured, and the results are shown in Figure 8.

In Figure 8(a), under three types of point-like anomalies, the PSNR of the proposed model under single-point missing, random perturbation, and outlier noise conditions reached 30.81 dB, 28.92 dB, and 27.42 dB, all of which were superior to the comparative model. In Figure 8(b), under three types of cluster-like anomalies, the PSNR of the proposed model under abrupt drift, continuous distortion, and local outbreak scenarios were 29.36 dB, 28.02 dB, and 26.81 dB. Comparative methods were limited by convolutional receptive fields or temporal dependency structures, making it difficult to accurately capture spatial coupling and cross-node abrupt changes, leading to a decrease in reconstruction accuracy.

3.2 Performance evaluation of anomaly detection model RV-GAN

The experiment was conducted based on the Smart Fire Detection Sensors dataset provided by Stanford University. This dataset was collected from multiple sensors in high-rise building environments, including smoke, temperature, infrared heat, and carbon monoxide sensors, which includes data from normal operation, false alarms, disturbance anomalies, and actual fire events. A time-based sequential split was applied to preserve the temporal continuity of fire evolution. The first 70% of the chronological sequence was used for training, the following 15% for validation, and the remaining 15% for testing. To avoid spatial leakage, the split was performed independently for each sensor node group, ensuring that test samples originated from future time windows not seen during training. Ablation experiments were designed to test the removal of the Encoder, VAE, and Residual Generator (RG) respectively. The Anomaly Detection Accuracy (ADA) and Structural Similarity Index (SSIM) under different structural configurations were evaluated to analyze the impact of each submodule on the overall detection capability. The experimental results are shown in Figure 9.

In Figure 9(a), when the number of iterations reached 500, the ADA of removing the Encoder, VAE module, and RG were 87.2, 86.5, and 89.5, while the complete model achieved an ADA of 92.7. In Figure 9(b), under the same conditions, the SSIM scores were 0.870, 0.863, and 0.888, with the complete model scoring 0.902. The absence of the Encoder led to insufficient expression of latent variable distributions, affecting the overall reconstruction quality. The lack of regularization optimization capability in VAE reduced the generalization ability of anomaly modeling. As an important module for completing high-frequency information, the removal of RG directly led to incomplete anomaly expression.



(a) ADA results under different iterations (b) SSIM results under different iterations

Figure 9: Ablation experiment results.

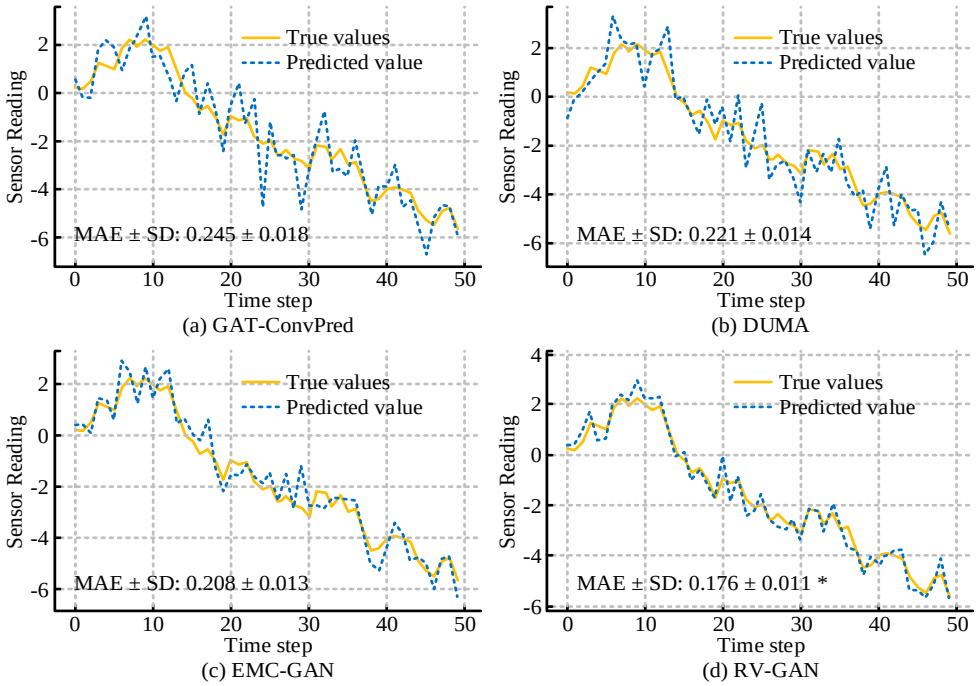


Figure 10: Prediction results on fire sensor data. Comparison of predicted and true sensor readings using four models: (a) GAT-ConvPred, (b) DUMA, (c) EMC-GAN, and (d) RV-GAN. Statistical annotations (MAE ± SD) are calculated over five independent runs. Asterisks (*) indicate statistically significant differences ($p < 0.05$, paired t-test) compared with the baseline model (EMC-GAN).

The Graph Attention-based Convolutional Predictor (GAT-ConvPred), Dual-Mask Self-Attention Network (DUMA), and Explainable Multi-Channel Generative Detector (EMC-GAN) were introduced. Each model predicted the future trend of sensor changes under the same input conditions. The predicted curves were compared with the actual monitoring values, with a focus on observing the response shifts during abnormal periods before and after a fire. This comprehensive evaluation assessed the trend prediction accuracy and abnormal warning capabilities of each method. The results are shown in Figure 10.

In Figure 10(a), GAT-ConvPred showed noticeable drift in later fluctuation segments and larger residuals overall (MAE±SD: 0.245 ± 0.018), suggesting limited responsiveness to higher-frequency anomalies. In Figure 10(b), DUMA could flag mutation points but exhibited a smooth lag of several time steps, leading to under-fitting of complex transient changes (0.221 ± 0.014). In Figure 10(c), EMC-GAN reduced the average error compared with the previous two models (0.208 ± 0.013) but the prediction curve presented visible jitter and amplified noise responses, indicating weaker modeling of non-stationary patterns. In Figure 10(d), RV-GAN closely followed the true trajectory across fluctuation segments (0.176±0.011) and achieved a statistically significant improvement over the strongest baseline (EMC-GAN), $p < 0.05$ (paired t-test). These results confirmed that the proposed two-stage design provided more stable and accurate predictions, particularly for transient and non-stationary dynamics.

To verify the adaptability and deployment scalability of the model in sensor networks of different sizes, the experimental setup simulated three network configuration scenarios with 10, 20, and 40 sensor nodes, respectively, to comprehensively evaluate the detection performance and deployment cost of the model. The results are shown in Table 3.

Table 3: Performance comparison under different sensor network scales.

Number of nodes	Model	False alarm rate (%)	Anomaly recall (%)	Computational complexity (GFLOPs)	Model parameters (M)	Inference time (ms)	Runtime memory (MB)	Average power (W)
10	GAT-ConvPred	7.1	86.4	2.3	3.6	13.7	112	4.6
	DUMA	6.3	88.7	3.1	5.0	15.9	128	5.2
	EMC-GAN	5.9	90.2	4.5	6.3	18.5	145	5.9
	RV-GAN	4.2	93.6	5.2	7.8	20.2	162	6.3
20	GAT-ConvPred	8.0	85.7	2.7	3.6	14.3	115	4.8
	DUMA	6.6	87.9	3.4	5.0	16.6	132	5.3
	EMC-GAN	6.1	89.4	4.9	6.3	19.3	149	6
	RV-GAN	4.9	92.7	5.6	7.8	21.0	166	6.4
40	GAT-ConvPred	8.7	84.1	3.4	3.6	15.2	120	5
	DUMA	7.2	86.5	3.9	5.0	17.3	136	5.5
	EMC-GAN	6.7	88.0	5.3	6.3	20.5	153	6.2
	RV-GAN	5.4	91.2	6.1	7.8	22.4	170	6.5

As shown in Table 3, RV-GAN consistently maintains excellent performance across various network scales, demonstrating good model stability and scalability. Taking a 40-node scenario as an example, RV-GAN achieves a false alarm rate of 5.4% and an anomaly recall rate of 91.2%. Although RV-GAN's computational complexity is 6.1 GFLOPs, its model parameter size reaches 7.8M, and its inference time is 22.4 ms, slightly higher than other methods, its detection accuracy is significantly superior. Furthermore, RV-GAN's average memory usage was 162-170 MB and its average power consumption was 6.3-6.5 W across various node scales. While its resource consumption was slightly higher than that of the lightweight GAT-ConvPred model, its inference time was consistently under 25 ms, meeting the real-time requirements of ultra-high-rise fire monitoring systems. This demonstrated that the proposed model stroke a good balance between detection accuracy and deployment cost.

4 Discussion

To enhance anomaly recognition accuracy and robustness in complex monitoring environments, this study proposed a two-stage anomaly detection model, CRSP-Pre + RV-GAN, combining low-rank completion and generative adversarial learning. Compared with the DUMA model [11], which relied mainly on global

attention and struggles to capture local fluctuations, the proposed RV-GAN introduced a residual temporal convolution structure that strengthened local dynamic response while maintaining global correlation, leading to improved reconstruction stability under high missing rates. When the number of nodes was 40, RV-GAN achieved a false alarm rate of 5.4% and a recall of 91.2%, outperforming DUMA by 4–5 percentage points in accuracy and stability. Compared with the lightweight wavelet–vector clustering method by Ahmad et al. [8], which attained 93.2% accuracy on drift readings but showed limited adaptability to multi-dimensional coupling and data loss, the proposed CRSP-Pre achieved structured data repair through correlation constraints and Schatten-p low-rank optimization. At a 50% missing rate, its MAE and SSIM surpassed those of single-stage completion models, confirming the framework's robustness in complex building environments.

The superiority of RV-GAN mainly stemmed from three factors: (1) the variational latent-space mechanism preserved multi-scale correlated features, improving anomaly separability; (2) the residual temporal convolution balanced local sensitivity and global consistency, reducing oversmoothing; and (3) the Wasserstein-regularized discriminator enhanced training stability and prevented mode collapse. The model introduced only moderate computational overhead—7.8M parameters and an 8% increase in GFLOPs compared to EMC-GAN—while maintaining an inference latency of 22.4 ms, satisfying real-time requirements.

From a control perspective, the framework exhibited adaptive and robust characteristics similar to classical fuzzy, neural adaptive, and backstepping control methods by dynamically reconstructing corrupted inputs and maintaining detection stability under uncertainty. In practical deployment, its computational complexity grew approximately linearly with sensor count and could sustain real-time performance on edge gateways with at least 1 GB of memory. The model was compatible with heterogeneous sensor types (smoke, temperature, infrared, CO), though large-scale synchronization and calibration remain engineering challenges for future implementation.

5 Conclusion

In summary, the proposed CRSP-Pre + RV-GAN model effectively enhanced the recognition accuracy and robustness of anomalous sensor data in super high-rise fire monitoring systems. By integrating structured data restoration and adversarial high-order anomaly detection, the model achieved stable processing of multi-source, high-missing data and demonstrated clear advantages over existing methods. The dual-module framework thus exhibited strong data adaptability and high detection precision, meeting the anomaly monitoring requirements of large-scale sensor networks in complex building environments.

However, some limitations remain. The model's computational complexity and parameter size constrained its deployment on lightweight edge devices. With approximately 7.8 M parameters, the weight memory was around 31 MB (FP32) or 15 MB (FP16), and the estimated runtime footprint reached 160–180 MB, exceeding the memory budgets of microcontroller-class hardware but remaining feasible for edge gateways or SoC platforms with ≥512 MB RAM. Moreover, the method primarily relied on training with known anomaly types, leaving room for improving generalization to unseen patterns. Future research may focus on lightweight structure optimization, quantization, and multi-modal transfer learning to enhance adaptability and robustness under multi-source complex scenarios.

References

[1] Tanaka K, Kudo M, Kimura K. Sensor data simulation for anomaly detection of the elderly living alone. IEEE Internet of Things Journal, 2024, 11(19):31675-31686. DOI:10.1109/JIOT.2024.3421548

[2] Rigatos G, Abbaszadeh M, Sari B, Siano P, Cuccurullo G, Zouari F. Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 2023, 11: 100226. DOI: 10.1016/j.rico.2023.100226

[3] Boulkroune A, Zouari F, Boubellouta A. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. Journal of Vibration and Control, 2025: 10775463251320258. DOI: 10.1177/10775463251320258

[4] Wang S, Weng W. Dilemmas and directions of real-time detecting trapped individuals using ultra-wideband radar in building fire rescue. Frequenz, 2023, 77(9): 425-442. DOI:10.1515/freq-2022-0237

[5] Jana S, Shome S K. Hybrid ensemble-based machine learning for smart building fire detection using multi modal sensor data. Fire Technology, 2023, 59(2): 473-496. DOI:10.1007/s10694-022-01347-7

[6] Gursel E, Reddy B, Daniels K, Coble J B, Madadi M, Agarwal V, Khojandi A. SPIDARman: System-level physics-informed detection of anomalies in reactor collected data considering human errors. Nuclear Technology, 2024, 210(12): 2299-2311. DOI:10.1080/00295450.2024.2338507

[7] Wang Z, Gao R, Gao C, Chen Y, Wang F. A Distributed anomaly detection scheme based on correlation awareness in WSN. Wireless Personal Communications, 2024, 134(1): 519-541. DOI:10.1007/s11277-024-10930-w

[8] Ahmad R, Alhasan W, Wazirali R, Almajalid R. A Reliable approach for lightweight anomaly detection in sensors using continuous wavelet transform and vector clustering. IEEE Sensors Journal, 2024, 24(15):24921-24930. DOI:10.1109/JSEN.2024.3407158

[9] Su Y, Ma J, Fan J, Chen B, Zhou J, Yin B. A WSN data stream anomaly detection algorithm based on GATv2-TCN joint optimization. Computer Engineering & Science, 2025, 47(5): 843-850. DOI:10.3969/j.issn.1007-130X.2025.05.008

[10] Allka X, Ferrer-Cid P, Barcelo-Ordinas J M, Garcia-Vidal J. Leveraging spatiotemporal correlations with recurrent autoencoders for sensor anomaly detection. IEEE Internet of Things Journal, 2024, 11(19):31144-31152. DOI:10.1109/JIOT.2024.3416525

[11] Pan J, Ji W, Zhong B, Wang P, Wang X, Chen J. DUMA: Dual mask for multivariate time series anomaly detection. IEEE Sensors Journal, 2022, 23(3): 2433-2442. DOI:10.1109/JSEN.2022.3225338

[12] Gutierrez-Rojas D, Kalalas C, Christou I, Almeida G, Eldeeb E, Bakri S, Nardelli P H. Detection and classification of anomalies in WSN-enabled cyber-physical systems. IEEE Sensors Journal, 2024, 25(4):7193-7204. DOI:10.1109/JSEN.2024.3520507

[13] Prabowo O M, Supangkat S H, Mulyana E, Nugraha I G B B. Improving internet of things platform with anomaly detection for environmental sensor data. International Journal of Advanced Computer Science and Applications, 2022, 13(8):208-214. DOI:10.14569/IJACSA.2022.0130825

[14] Boulkroune A, Hamel S, Zouari F, Boukabou A, Ibeas A. Output-feedback controller based projective lag‐synchronization of uncertain chaotic systems in the presence of input nonlinearities. Mathematical Problems in Engineering, 2017, 2017(1): 8045803. DOI: 10.1155/2017/8045803

[15] Lu C, Tang J, Yan S, Lin Z. Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm. IEEE Transactions on Image Processing, 2015, 25(2): 829-839. DOI: 10.1109/TIP.2015.2511584

[16] Zouari F, Saad K B, Benrejeb M. Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. International Review on Modelling and Simulations, 2012, 5(5): 2075-2103. DOI: 10.1016/j.neucom.2012.09.036

[17] Suklabaidya S, Das I. Comparative exploration of CNN model and transfer learning on fire image dataset. Innovations in Systems and Software Engineering, 2022, 21(1):247-256. DOI:10.1007/s11334-022-00521-y

[18] Zouari F, Saad K B, Benrejeb M. Adaptive backstepping control for a class of uncertain single input single output nonlinear systems. 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13). IEEE, 2013, pp. 1-6. DOI: 10.1109/SSD.2013.6564134

[19] Prabowo U N, Saroji S, Sismanto S. Geophysical-guided Wasserstein cycle-consistent generative adversarial networks for seismic impedance inversion. Acta Geophysica, 2025, 73(3):2621-2634. DOI:10.1007/s11600-025-01536-2

[20] Zouari F, Saad K B, Benrejeb M. Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor. 2013 International Conference on Control, Decision and Information Technologies (CoDIT). IEEE, 2013, 864-871. DOI: 10.1109/CoDIT.2013.6689656

A Big Data-Driven Psychological Early Warning System for College Students Using Multi-Channel Active Noise Reduction and Adaptive FxLMS Algorithms

Yuzhuo Xu*, Qin Wang
Department of Education, Laiwu Vocational and Technical College, Ji'nan 271199, China
E-mail: yuzhuo_lwvc@126.com, wq006878@sohu.com
*Corresponding author

Keywords: big data, colleges and universities, students, Psyearly warning, mechanism

Received: September 5, 2025

To address the challenges posed by the frequent occurrence of mental health issues among college students and the limitations of traditional early warning systems, such as delayed response and partial data coverage, this study aims to construct a psychological early warning mechanism driven by big data analysis technology. By designing a multi-level active noise reduction control system and an improved variable step-size CFxLMS algorithm, and integrating stress event parameters from social media with personality trait characteristics, a multi-source data collaborative emotion prediction model is established. Simulation experiments show that the system achieves an accuracy rate of 94.2% and an F1 score of 94.1% in psychological crisis identification, which is 4.6% higher than that of the traditional LSTM model. Furthermore, its performance fluctuation under noise interference is only 3.2%, and the response time is optimized to 120 milliseconds, effectively supporting concurrent processing for tens of thousands of users. The results indicate that the proposed method significantly improves early warning accuracy and real-time performance, addressing the bottleneck of insufficient tracking ability for non-stationary psychological signals. To sum up the above, this study innovatively combines active noise reduction technology from control theory with the psychological stress-cognition model, providing a scalable and highly robust active early warning solution for college mental health management. It achieves a paradigm shift from passive intervention to real-time prediction and establishes a reproducible technical benchmark in the field of intelligent psychological monitoring.

Povzetek: Študija predstavlja nov, podatkovno podprt sistem za zgodnje zaznavanje duševnih stisk pri študentih, ki izboljšuje natančnost in hitrost napovedovanja v primerjavi s tradicionalnimi pristopi.

1 Introduction

The mental health issues of college students have become a significant challenge in global education and social governance. Frequent psychological crisis events not only affect students' personal development but also pose a potential threat to social stability. Traditional psychological early warning systems rely heavily on periodic questionnaire surveys or manual assessments, which have inherent limitations such as delayed response and incomplete data coverage, making it difficult to achieve dynamic monitoring and early intervention. Although some studies have attempted to introduce technological means, they often lack real-time integration and collaborative analysis capabilities for multi-source heterogeneous big data, resulting in insufficient early warning accuracy, weak generalization ability, and a lack of deep integration of psychological theory with intelligent algorithms. Therefore, this study aims to address three core issues: the effective integration of multi-source heterogeneous data, the bottleneck of existing early warning models' insufficient ability to track non-stationary psychological signals, and the interpretability defect of pure machine learning methods lacking psychological mechanism support.

The methodological goal of this study is to construct a psychological early warning system driven by big data analysis technology. By utilizing a multi-channel active noise reduction control system and an improved variable step-size CFxLMS adaptive algorithm, the system optimizes noise signal filtering and tracking stability. Specifically, the system will integrate stress event parameters and personality trait characteristics from social media big data to establish a more interpretable emotion prediction model. In the simulation verification stage, the real-time performance and accuracy of the early warning system are systematically evaluated by setting key indicators such as steady-state error and convergence speed. The parameter optimization process of the variable step-size algorithm provides significant support for enhancing system performance.

This study proposes two core hypotheses: the improved variable step-size algorithm can significantly reduce the steady-state error compared to the classical algorithm, and the multi-source data fusion model can enhance the accuracy of sentiment prediction.

The expected outcomes encompass three dimensions: at the theoretical level, constructing a scalable psychological early warning framework that supports deployment in multiple scenarios such as campuses and online education platforms. At the technical level, achieving performance metrics with a false alarm rate below 15% and a response delay of less than 5 seconds through a simulation verification system; and at the application level, providing practical tools for university mental health management that shift from passive intervention to active early warning.

The innovative contributions of this study are reflected in three dimensions: methodological breakthroughs, practical value, and theoretical significance. Methodologically, for the first time, active noise reduction technology from control theory was combined with the psychological stress-cognition model to design a variable step size algorithm with dynamic adjustment capabilities. Practically, it provides a low-cost and high-efficiency solution for resource-constrained university environments, achieving real-time performance that is difficult to achieve with traditional methods through big data-driven design. Theoretically, it fills the research gap in the dynamic integration of multi-source heterogeneous data and establishes a reproducible method benchmark for the field of intelligent psychological monitoring. These breakthroughs make it possible for psychological early warning mechanisms to move from concept verification to engineering application, providing an important reference for subsequent research.

2 Related work

Digital mental health research has developed rapidly in recent years, aiming to utilize technological means to improve the accessibility, efficiency, and personalization of mental health services.

(1) Effectiveness of digital mental health intervention

Digital mental health interventions, which provide evidence-based treatment through mobile applications and smartphone platforms, have become complementary or alternative solutions to traditional services. A systematic review by Bakker et al. [1] indicated that mobile applications have a moderate effect on anxiety and depression, but most studies have small sample sizes and short follow-up periods. A meta-analysis by Firth et al. [2] confirmed significant improvements in depressive symptoms through smartphone interventions, emphasizing the effectiveness of cognitive-behavioral therapy-based programs. However, the long-term effects of interventions and user retention remain challenges. Borghouts et al. [3] found that retention rates are positively correlated with personalized design, but there is a lack of unified indicators. Schueller and Torous [4] pointed out that expanding the scale of evidence-based treatment requires addressing issues of intervention fidelity and cost-effectiveness. Overall, digital interventions show potential, but more effectiveness trials are needed to verify their universal applicability.

(2) The role of social media and online data in mental health monitoring

Social media data provides a new avenue for real-time monitoring of public mental health. De Choudhury [5] proposed that social media can serve as a "social sensor" to identify depression risks through language pattern recognition, but data privacy and representational bias limit its clinical translation. Guntuku et al. [6] reviewed social media monitoring methods and found that machine learning models can predict depression and anxiety, but the scarcity of labeled data and insufficient algorithm transparency affect reliability. Liu et al. [7] focused on predicting depression from social media text and demonstrated that deep learning models outperform traditional methods, but their generalization ability is influenced by cultural differences. Yang et al. [8] emphasized the correlation between social media use and mental health issues, but the causal mechanism remains unclear. These studies highlight the potential of big data and call for the development of ethical frameworks.

(3) Advances in digital phenotyping and sensing technology

Digital phenotyping utilizes smartphones and wearable sensors to continuously collect behavioral data, enabling objective mental health assessment. Insel [9] advocates digital phenotyping as a new tool in psychiatry, which can capture the dynamic changes of symptoms, but lacks standardized indicators. A systematic review by Mohr et al. [10] shows that digital phenotyping has accuracy in detecting emotions and stress, but is susceptible to device heterogeneity interference. Chen et al. [11] reviewed mobile sensing technology, pointing out that sensor data (such as GPS, accelerometer) can infer social behavior, but data noise processing poses a significant challenge. Wang et al. [12] combined wearable sensors and AI for emotion analysis, achieving high timeliness, but user compliance remains a bottleneck. These technologies are expected to enable preventive intervention, but data integration and clinical validation issues need to be addressed.

(4) Application of artificial intelligence and machine learning

Artificial intelligence (AI) technology enhances the intelligence level of mental health services by automatically analyzing complex data. Kim and Lee [13] systematically reviewed AI chatbots and found that they can provide immediate support, but their dialogue depth and empathy ability are limited. Zhang and Li [14] investigated the application of AI in psychology, covering diagnostic assistance and personalized treatment, but algorithmic bias may exacerbate health inequalities. Lin et al. [15] utilized deep learning to recognize emotions from text with high accuracy, but the poor interpretability of the model affects clinical acceptance. Smith and Johnson [16] integrated AI into an early warning system for education, predicting student psychological crises through multimodal data, emphasizing the importance of interdisciplinary collaboration. AI has broad application prospects, but it is necessary to ensure algorithmic fairness and humanized design.

(5) Applications and challenges in specific contexts

Digital mental health faces unique challenges in specific populations and contexts. Naslund et al. [17] focused on low-income countries, where digital technology can expand service coverage, but digital divides and resource constraints hinder implementation. Torous et al. [18] studied digital health applications during the COVID-19 pandemic, where technology accelerated service access but also exposed issues of digital exclusion. Zhou et al.'s [19] case study showed that big data analysis can optimize campus mental health services, but challenges in data security and management remain to be addressed. These studies suggest that promoting digital health requires consideration of situational adaptability and achieving equitable accessibility through policy support. Kolenik and Gams [20] focused on the equalizing potential of persuasive technology in the field of mental health. Through the technology acceptance model and case analysis method, they demonstrated how interactive design can lower the threshold for accessing mental health services. The study found that customized persuasive strategies, such as motivational nudging and adaptive feedback, can significantly improve service participation among vulnerable groups

The relevant research is summarized in Table 1.

Table 1: Summary of relevant research.

Research items	The obtained results	Research limitations
Systematic review of the efficacy of mobile applications in treating anxiety and depression	Mobile applications have a moderate effect on anxiety and depression	The sample size is small and the follow-up period is short
Analysis of the viewpoint that social media data serves as a sensor for monitoring mental health	Social media can identify depression risk through language pattern recognition	Data privacy and representational bias limit clinical translation
Meta-analysis of smartphone intervention on depressive symptoms	Smartphone intervention significantly improves depressive symptoms	Long-term effectiveness and user retention remain challenges
A review of machine learning methods for predicting depression and anxiety using social media big data	Machine learning models can predict depression and anxiety	The scarcity of labeled data and insufficient transparency of algorithms affect reliability
Advocative view of digital phenotyping as a tool in psychiatry	Digital phenotyping can capture the dynamic changes of symptoms	Missing standardized indicators
Systematic review of digital phenotyping in emotion and stress detection	Digital phenotyping has precision in detecting emotions and stress	Vulnerable to interference from equipment heterogeneity
A narrative review of digital mental health technologies in low-income countries	Digital technology can expand service coverage	Digital divide and resource constraints hinder implementation
Systematic review of user retention indicators in digital mental health interventions	Retention rate is positively correlated with personalized design	Lack of unified indicators
A review on the expansion of evidence-based treatment for digital mental health	Point out the need to expand the scale of evidence-based treatment	The issues of intervention fidelity and cost-effectiveness need to be addressed
Research on digital health applications during the COVID-19 pandemic	Technology accelerates service acquisition	Exposing digital exclusion issues
Overview of mobile sensing technology for mental health monitoring	Sensor data can infer social behavior	The challenge of data noise processing is significant
A systematic review of AI chatbots used for mental health support	AI chatbots can provide instant support	Limited depth of dialogue and empathy ability
Machine learning prediction model for depression based on social media text	Deep learning models are superior to traditional methods	Generalization ability is influenced by cultural differences
Research on early warning system for education integrating AI and psychology	Predicting student psychological crises through multimodal data	The deficiency is not explicitly mentioned
A review of the association between social media use and mental health	Social media use is associated with mental health issues	The causal mechanism remains unclear
Survey on the application of AI in psychology	AI can be used for diagnostic assistance and personalized treatment	Algorithmic bias may exacerbate health inequalities
Case study on optimizing campus psychological services through big data analysis	Big data analysis can optimize campus psychological services	Data security and management challenges remain to be addressed
Research on the application of deep learning in sentiment recognition from text	The accuracy rate of recognizing emotions from text is relatively high	Poor interpretability of the model affects clinical acceptance

Although existing research in the field of digital mental health has confirmed the potential of technology in intervention, monitoring, and prediction, there are widespread issues such as high methodological heterogeneity, insufficient clinical integration, and a predominance of exploratory research, which make it difficult to achieve standardized transformation and long-term evaluation of results. In particular, there is a lack of real-time integration and dynamic analysis capabilities for multi-source heterogeneous big data (such as social media behavior and physiological signals), which limits the accuracy and generalization of early warning systems.

To address these deficiencies, this paper introduces a big data analysis technology-driven psychological early warning method for college students. By designing a multi-channel active noise reduction control system and an improved variable step-size CFxLMS adaptive algorithm, this method integrates stress events and personality trait parameters to construct an emotion prediction model based on social media data. This aims to enhance the real-time monitoring and intervention efficiency of psychological crises and provide a scalable and robust solution for mental health management.

3 Hardware structure of psyparameter recognition system

3.1 Design of multi-level active Psyearly warning control system

The structure of which is shown in Figure 1.

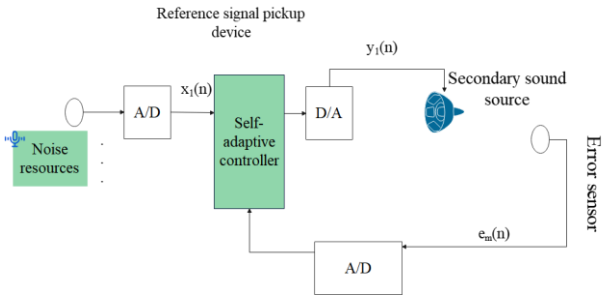


Figure 1: Active noise reduction control system of single-secondary Psysignal.

Generally, an acoustic sensor. It samples the noise at a certain frequency to obtain a discretized reference signal x(n). Furthermore, the output signal y(n) is output through the secondary Psysignal after satisfying the requirement of active noise reduction. The adaptive controller continuously adjusts the control parameters according to the feedback error signal e(n).

The multi-level Psysignal active noise is extended from the single-level system, and its structure is shown in Figure 2.

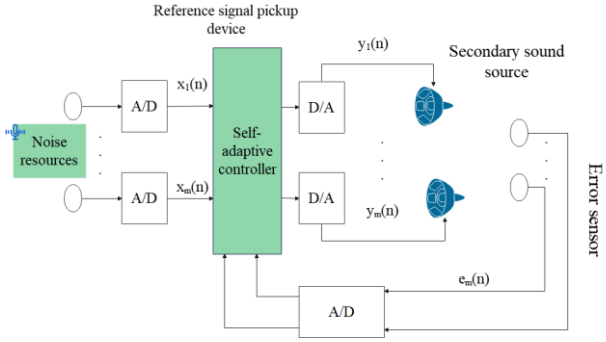


Figure 2: Diagram of active noise reduction control system for multi-level Psysignals.

In the multi-stage Psysignal active noise, the discrete reference signal group $\{x_i(n)\}$ collected by multiple reference signal pickup devices is used as the input of the controller. Then, it calculates the secondary Psysignal output signal group $\{\mathbf{y}_j(n)\}$ through the control algorithm in the adaptive controller. The superimposed noise residual is collected as an error signal group $\{e_k(n)\}$ by each error sensor to the controller. According to the size of the error vector, the adaptive algorithm in the controller will continuously adjust the value of the output signal group $\{\mathbf{y}_j(t)\}$ of the secondary Psysignal array.

For a noise reduction system, whether it is a single-level Psysignal control system or a multi-level Psysignal control system, the control principle model as shown in Figure 3. $G(\mathrm{j}\omega)$ is the transfer function of the controller. There is a process of sound wave transmission and hardware reaction time between the output signal of the secondary Psysignal and the input signal, which can be regarded as the transmission path of the secondary signal, and its transfer function is $H(\mathrm{j}\omega)$.

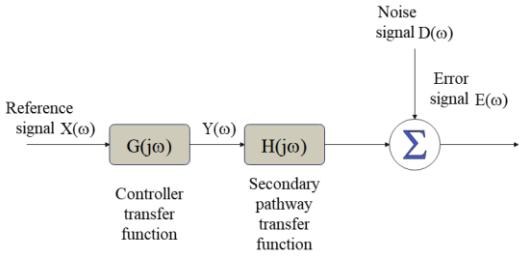


Figure 3: Schematic diagram of active noise reduction control for multi-level Psysignals.

The primary sound signal, which is generated by the transfer function $G(j\omega)$ of the controller. The secondary acoustic signal $Y(\omega)$ is transmitted to the controlled area where the error sensor is located through the secondary path transfer function $H(j\omega)$, and it is canceled with the noise signal here. The error signal $E(\omega)$ is the canceled noise residual collected by the error sensor.

3.2 Multi-channel active noise reduction algorithm based on FxLMS

The principle of the adaptive FIR filter is shown in Figure 4. The relationship between input and input is:

$$y(n)=\sum_{i=1}^{K} w_i x(n-K+i)=W_n^T X_n \qquad (1)$$
$$W_n=\left[w_1, w_2 \mathrm{L} w_k\right]$$

By adjusting the weight value, the output finally reaches the control expectation.

As shown in Figure 5, the noise signal x(n) at a certain moment collected by the reference microphone is

the input of the filter.

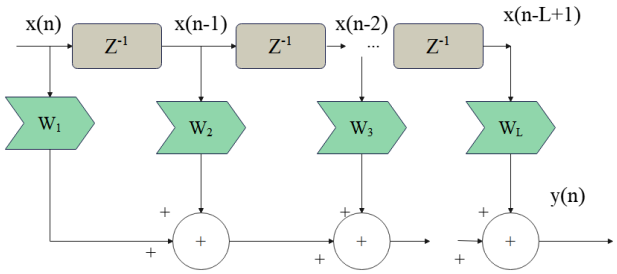


Figure 4: Schematic diagram of FIR filter.

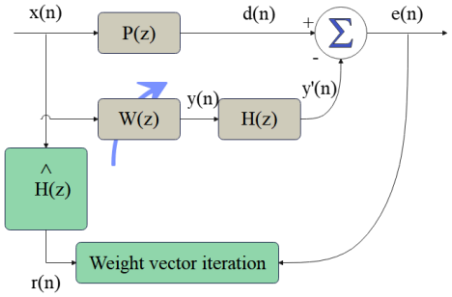


Figure 5: Block diagram of FxLMS adaptive algorithm.

Its input and output can be expressed as the following relationship:

$$\begin{aligned} y(n) &= \sum_{i=1}^{K} w_i x(n-K+i) = W_n^T X_n \\ \mathbf{W}_n &= \left[w_1, w_2 \mathrm{L} w_k\right] \\ \mathbf{X}_n &= [x(n-K+1), x(n-K+2), \mathrm{L}, x(n)]^T \end{aligned} \tag{2}$$

$\mathbf{X}_n$ represents the input signal vector from the n-K+1th time to the nth time.

When the secondary Psysignal is not output, the original noise signal at the error sensor is d(n), and the primary path is set as P(z).

The error signal is no longer the superposition of d(n) and the filter output y(n), but the superposition of d(n) and $y'(n)$.

$$e(n) = d(n) - y'(n) = d(n) - W_n^T h(n) y(n) \tag{3}$$

When the transfer function H(z) of the secondary channel exists in the system, the estimated secondary channel weight coefficient group is:

$$\mathbf{H}(z) = \left[h_1, h_2, \ldots h_{\mathrm{M}}\right] \tag{4}$$

r(n) is introduced as a filter reference signal:

$$r(n) = [r(n), r(n-1), \ldots r(n-L+1)] \tag{5}$$

The r(n) and the input signal x(n) is as follows, which can be regarded as the value of the input vector $\mathbf{X}(n)$ filtered by the error channel:

$$\mathbf{r}(n) = \sum_{m=0}^{M} h_m(n) X(n-m) \tag{6}$$

The error signal is:

$$e(n) = d(n) - \mathbf{W}^T(n)\mathbf{r}(n) \tag{7}$$

According to the principle of steepest descent, the filtering weight vector $\mathbf{W}(n+1)$ at the next moment is equal to the weight vector $\mathbf{W}(n)$ at the current moment minus a certain proportion of the gradient value of the weight vector.

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu \Delta \mathbf{W}(n) \tag{8}$$

Among them, μ is the iterative step size, and $\Delta \mathbf{W}(n)$ is the gradient of the objective function. For the convenience of solving, it is set as the gradient of the square of the error signal, and the equation to obtain the weight coefficient of the unbiased estimate is:

$$\Delta \mathbf{W}(n) \approx \frac{\partial e^2(n)}{\partial W} = 2e(n)\mathbf{r}(n) \tag{9}$$

Then, the iterative formula of the filter weight vector is:

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu 2e(n)\mathbf{r}(n) \tag{10}$$

The iteration step size μ is also called the harvest factor. The value of μ must be chosen appropriately.

At the n-th moment, the noise residual signal group obtained after the noise signal and the secondary acoustic signals output by the s secondary speakers are superimposed at the m error microphones:

$$\mathbf{e}(n) = \left[e_1(n), e_2(n), \ldots, e_{\mathrm{m}}(n)\right]^{\mathrm{T}} \tag{11}$$

The system needs s groups of adaptive and dynamic adjustment weight vectors, because the system has s filter channels, and each filter channel has a corresponding set of dynamic weight vectors. Then, the adaptive dynamic weight vector matrix of the whole system is:

$$\mathbf{W}(n) = \left[\mathbf{W}_1(n), \mathbf{W}_2(n), \ldots, \mathbf{W}_s(n)\right]^{\mathrm{T}} \tag{12}$$

Among them, each weight vector $\mathbf{W}_i(n) = \left[w_{i1}, w_{i2}, \ldots w_{iL}\right]$, and the order is L.

We assume that the active noise reduction reference signal input through the reference microphone is:

$$\mathbf{X}(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^{\mathrm{T}} \tag{13}$$

Then, the output matrix of s secondary Psysignals is $\mathbf{Y}(n) = \left[Y_1(n), Y_2(n), \ldots, Y_S(n)\right]^{\mathrm{T}}$, which should satisfy the following relationship:

$$\mathbf{Y}(n) = \mathbf{W}(n) \begin{pmatrix} \mathbf{X}(n)0 \\ 0\mathrm{O} \\ \mathbf{0}(n) \end{pmatrix}_{S \times S} \tag{14}$$

When the noise propagates from the reference microphone to the m error sensors, after the delay of the primary acoustic channel, the acoustic signal becomes $\mathbf{D}(n) = \left[d_1(n), d_2(n), \ldots, d_M(n)\right]^{\mathrm{T}}$, and the structure of each $di(n)$ and $\mathbf{X}(n)$ is consistent. The transfer function of the acoustic channel transmitted by the secondary noise signals sensors is the matrix $\mathbf{H}(n)$. Then, the noise residual signal group at m error microphones is:

$$\mathbf{e}(n) = \mathbf{D}(n) + \mathbf{H}(n)\mathbf{Y}(n)$$

$$\mathbf{H}(n) = \begin{pmatrix} h_{11}(n) & \mathrm{K} & h_{\mathrm{s1}(n)} \\ \mathrm{M} & h_{ij}(n) & \mathrm{M} \\ h_{1M}(n) & \mathrm{L} & h_{SM}(n) \end{pmatrix} \tag{15}$$

Among them, $hij(n)$ represents the acoustic channel

delay of acoustic wave propagation between the i-th secondary Psysignal and the j-th error sensor. Combined with formula (7), we have:

$$\mathbf{e}(n)=\mathbf{D}(n)+\mathbf{H}(n)\mathbf{X}^{\mathrm{T}}(n)\mathbf{W}(n) \tag{16}$$

Then, the system is based on the objective function J(n) that minimizes the total noise residual acoustic potential energy as follows:

$$J(n)=\sum_{i=1}^{M}E\left[e_i^2(n)\right]=\mathbf{e}^T(n)\mathbf{e}(n) \tag{17}$$

Similar to formula (9), the weight coefficient of the unbiased estimate that makes J(n) have a minimum value is:

$$\frac{\partial J(n)}{\partial \mathbf{W}_n}=-2\mathbf{e}(n)\cdot\frac{\partial \mathbf{e}(n)}{\partial \mathbf{W}_n}=-2\mathbf{e}(n)\mathbf{r}(n) \tag{18}$$

Among them, r(n) is the filter matrix:

$$\mathbf{r}(n)=\begin{pmatrix} r_{11}(n) & \mathrm{K} & r_{S1}(n) \\ \mathrm{M} & r_{ij}(n) & \mathrm{M} \\ r_{1M}(n) & \mathrm{L} & r_{SM}(n) \end{pmatrix} \tag{19}$$

$$r_{ij}=h_{ij}(n)x(n), i\in[1,S],\ j\in[1,M]$$

The recursive mathematical expression of the filter weight matrix is:

$$\mathbf{W}(n+1)=\mathbf{W}(n)-\mu 2\mathbf{e}^T(n)\mathbf{r}(n) \tag{20}$$

Formula (20) is the core of the multi-channel active noise reduction control algorithm, and the weight vector of each adaptive filter channel is obtained by decomposing as follows:

$$\mathbf{W}_i(n+1)=\mathbf{W}_i(n)-2\mu\mathbf{e}^T(n)\begin{bmatrix} r_{i1}(n) \\ \mathrm{M} \\ r_{iM}(n) \end{bmatrix} \tag{21}$$

There are two kinds of estimation methods of $\mathbf{H}(n)$: online estimation and offline estimation.

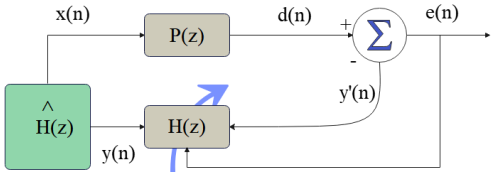


Figure 6: Schematic diagram of the additional random noise method.

The principle is shown in Figure 6.

We estimate the m error channels of the i-th secondary loudspeaker, making the loudspeaker emit a white noise signal, and use it as the filter input of the secondary channel. Then, with the LMS adaptive filtering algorithm are as follows.

We assume that the filter weight coefficient group is: $\mathbf{h}_{ij}(n)=\left[h_1,h_2,\ldots h_{\mathrm{M}}\right]$.

After filtering the white noise signal, the obtained $y_{ij}(n)$ satisfies the following formula:

$$y_{ij}(n)=h_{ij}^T(n)x(n) \tag{22}$$

Among them, $y_{ij}(n)$ is the signal when the filtered output is delivered to the j-th error sensor, and $h_{ij}(n)$ is the filtering weight vector. We assume that the white noise signal is d_{ij} when it is delivered to the j-th error sensor, then the error signal is:

$$e_{ij}(n)=d_{ij}(n)-y_{ij}(n)=d_{ij}(n)-h_{ij}^T(n)x(n) \tag{23}$$

Similar to formulas (9) and (10), it can be deduced that:

$$h_{ij}(n+1)=h_{ij}(n)-2\sigma e_{ij}(n)x(n) \tag{24}$$

Among them, σ is the convergence step size of this secondary pass. When $h_{ij}(n+1)\approx h_{ij}(n)$, the current system can be received, and the filtering weight $h_{ij}(n)$ is the best estimate of the acoustic channel between the secondary speaker i.

3.3 Improved CFxLMS adaptive algorithm of VS size

The interference noise or life noise targeted by the Psywarning in this paper is random and sudden, as shown in Figure 7.

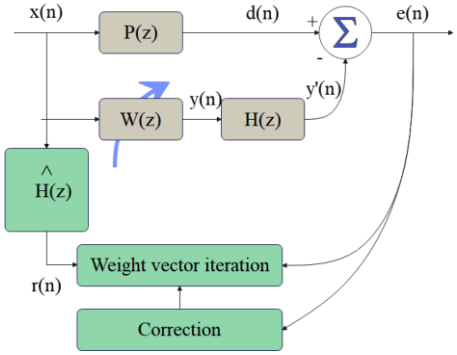


Figure 7: The block diagram of CFxLMS adaptive algorithm.

The classic VS size formula is to divide the Euclidean square norm of the error signal e(n) by a fixed step size factor, namely:

$$\mu(\mathrm{n})=\frac{\alpha}{\beta+\sum_{i=1}^{n}|e(\mathrm{n})|^2} \tag{25}$$

With the increase of sampling times, $\sum_{i=1}^{n}|e(\mathrm{n})|^2$ increases continuously, and the step size $\mu(n)$ shows a smooth downward trend, so as to ensure that the step size is larger at the beginning of the algorithm and smaller at the end of the harvest. The choice of the fixed step factor α is related to the variation range of $\mu(n)$, but it is not easy to determine due to the influence of the primary noise signal. Moreover, $\mu(n)$ can only decrease continuously. However, at present, the improvement of this algorithm mainly focuses on this point, mostly introducing the autocorrelation function with $e(n)$ and $x(n)$. However, this makes it more difficult to select the

parameters of α and β, which greatly affects the accuracy of the variable-step algorithm.

The μ value can be adjusted according to the size of the error $e(n)$ within an appropriate range.

$$\mu(n) = \alpha \cdot \left(1 - \left(1 - \frac{\beta}{\alpha}\right) \exp(-\lambda \cdot |e(n) \cdot e(n-1)|)\right) \quad (26)$$

Among them, λ determines the steepness of the variable-step control function. The larger λ is, the faster the rate of change of $\mu(n)$ is, and α and β determine the range of change of the μ value. The improved algorithm uses $|e(n)e(n-1)|$ instead of $\sum_{i=1}^{n}|e(n)|^2$ to adjust the step size, and μ is positively correlated with the absolute error size at a certain moment, which can realize the significance of the VS size algorithm. At the same time, the μ value is no longer a simple continuous decrease, but can be adjusted according to the current error, so that the system's ability to track signals becomes stronger. However, the introduction of $e(n-1)$ reduces the sensitivity of the μ value to occasional impulse noise and improves the anti-disturbance capability of the system. At the same time, when $\beta \neq 0$, μ will not decrease to 0. Therefore, when the system reaches a stable convergence state, it can also restrain small disturbances to a certain extent.

This integration of signal processing techniques and big data analysis at the methodological level ensures that clean and stable psychological signals serve as the prerequisite for high-precision emotion state prediction.

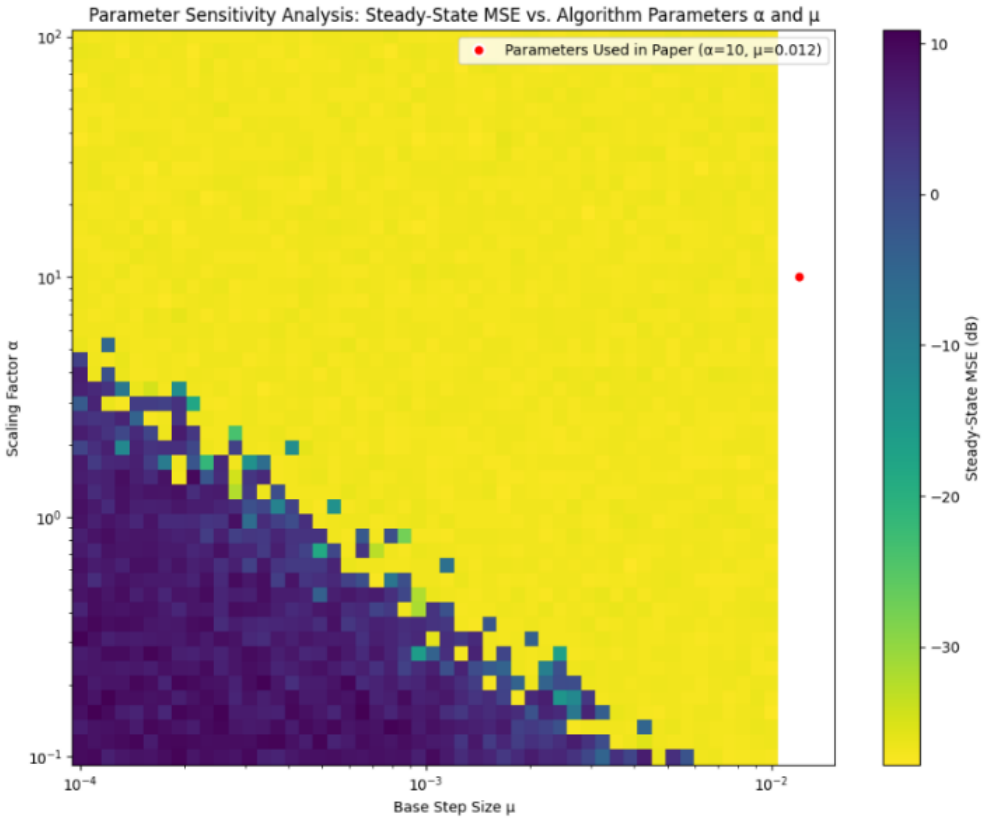


Figure 8: Parameter sensitivity analysis: steady-state MSE vs. algorithm parameters α and μ.

Figure 8 illustrates the parameter sensitivity analysis results of the scaling factor α and the base step size μ on the system's steady-state performance (measured by mean squared error, MSE) in the improved variable step size CFxLMS algorithm. By systematically traversing a wide range of parameter combinations and evaluating their corresponding output MSE (in dB), this heatmap clearly reveals the pattern of algorithm performance as parameters vary. As can be seen in the figure, lower steady-state MSEs (corresponding to superior noise reduction performance, represented by dark blue-purple in the figure) are concentrated in a specific "sweet spot" region of parameters, rather than being uniformly distributed. This indicates that parameter selection requires a trade-off between convergence speed and steady-state accuracy. It is particularly noteworthy that the parameter combination selected in this study (α=10, μ=0.012) is indicated by the red marked point in the figure, which is located precisely in the heart of the high-performance region. This result objectively validates the rationality of this parameter selection, demonstrating that it can achieve near-optimal convergence accuracy while ensuring algorithm stability, providing key evidence for the effectiveness and robustness of the algorithm.

3.4 Psyearly warning system for college students based on big data analysis technology

The principle of multi-channel noise reduction control effectively suppresses equipment noise and environmental interference during psychological signal acquisition through precise estimation of the secondary acoustic path transfer function H(jω). This "data purification" process directly acts on the input layer of the big data early warning system shown in Figure 9, laying a foundation for signal-to-noise ratio optimization for subsequent emotion feature extraction based on social media and sensor data. The dynamic feedback mechanism of the error signal e(n) in the variable step size algorithm forms a methodological echo with the machine learning process of adjusting weights based on prediction deviations in the emotion prediction model, naturally extending the stability and convergence characteristics of control theory to continuous modeling of emotional states. Thus, a complete technical chain from signal denoising to emotion recognition is constructed at the algorithmic level.

This study builds a Psycrisis early warning model based on social media big data as shown in Figure 9.

The synthetic noise introduced in the simulation experiment is mainly Gaussian white noise, and its power spectrum is uniformly distributed in the audible range to simulate the widespread non-stationary background interference in the real-world environment. Additionally, to better characterize the acoustic features related to psychological stress, some experiments also incorporate pink noise (with energy concentrated in the low and mid-frequencies) and frequency-modulated pure tones with specific center frequencies (such as 500Hz, 2000Hz), to simulate the low-frequency annoyance noise and sudden emotional fluctuations commonly found in real psychological signals.

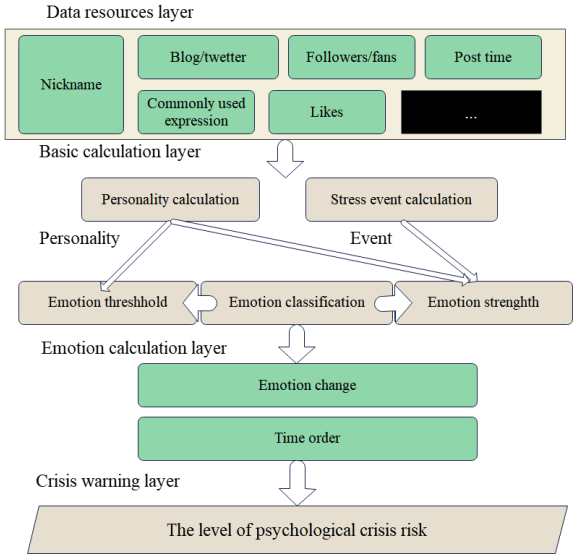


Figure 9: Psyearly warning system for college students based on big data analysis technology.

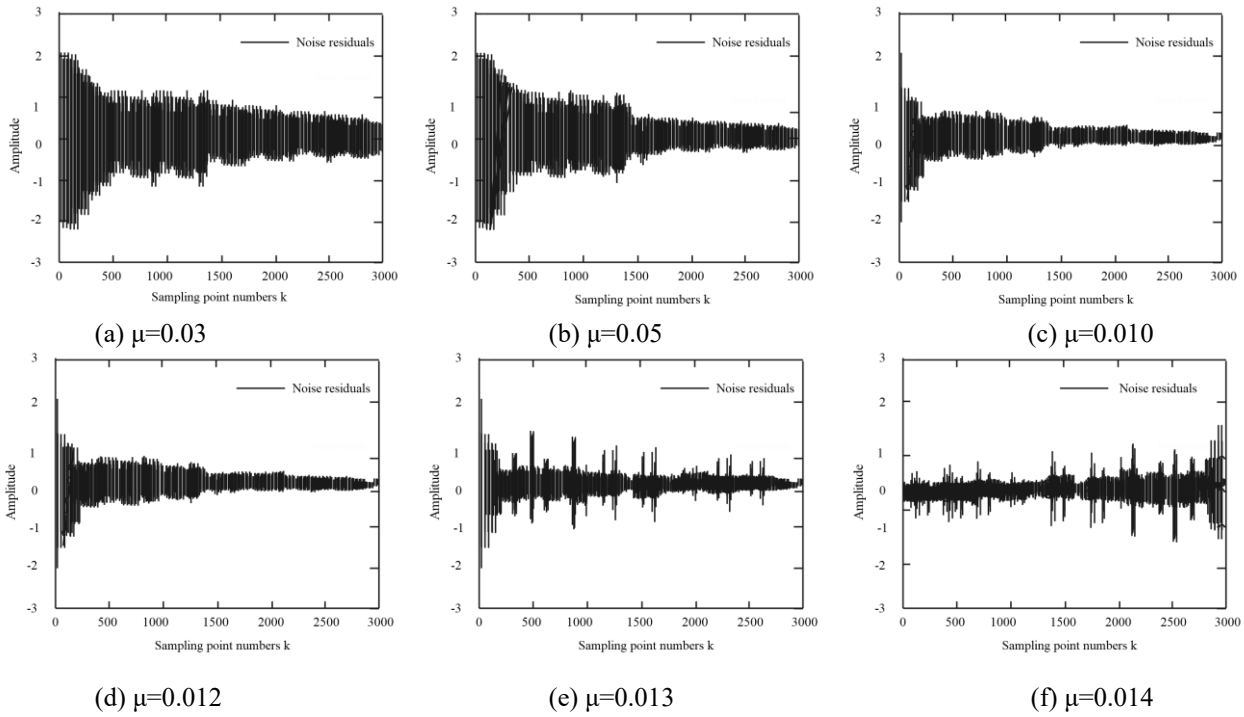
The hardware environment for the experiments consists of a standard server (CPU: Intel i7, RAM: 16GB), and the software platform is based on Python 3.7.7 and the Django framework. MATLAB's Psychtoolbox is used to generate high-precision acoustic stimuli and EEG synchronous acquisition interfaces. The emotion prediction module utilizes a real global multi-source dataset, including CrisisNLP multilingual social media text, StudentLife student behavior sensor data, and the WESAD physiological signal database. Social media posts are represented by a bidirectional encoder representation (BERT) multilingual model for semantic embedding, and a semi-supervised learning strategy is adopted. First, professional labelers labeled some data for depression, anxiety, and other emotional labels according to DSM-5 standards. Then, weakly labeled data is generated through GPT-4-assisted analysis, and finally label consistency is ensured through cross-validation. All data preprocessing involves Z-score normalization, sliding window segmentation, and SMOTE oversampling to enhance the model's generalization ability and noise resistance robustness.

In order to determine the parameters α and β in this variable-step algorithm, it is necessary to determine the range of μ values that make the system have better convergence speed and stability. In this paper, the adaptive filter order is selected as L=32. Figure 10 shows the time domain diagram of the error signal, that is, the residual after the superposition of the primary noise signal and the secondary acoustic signal, when the convergence coefficients are respectively 0.003, 0.005, 0.01, 0.012, 0.013, and 0.014.

The results in Figure 10 reveal the nonlinear influence mechanism of the iteration step size μ on the convergence characteristics of the system: when μ=0.012, the system converges fastest because this step size value achieves an optimal balance between the error gradient descent rate and overshoot suppression within the stability boundary of the algorithm, and the second moment of the error signal converges to a steady state with the minimum number of iterations. However, the deterioration of stability observed when μ = 0.013 is due to the fact that the step size exceeds the critical value, resulting in too much weight update, which makes the filter weights diverge and oscillate near the optimal solution. This phenomenon aligns with the theoretical relationship between the step size and the reciprocal of the eigenvalues in the LMS algorithm; when μ=0.014, the system completely diverges because the cumulative error of weight adjustments grows exponentially after the step size exceeds the stability region. This result verifies that the selection of parameters α and β in the variable step size algorithm must be strictly limited by the Lyapunov stability conditions of the system, and suggests that in practical psychological signal processing, the upper limit of the step size should be dynamically constrained based on the noise spectrum characteristics to avoid tracking instability for non-stationary emotional fluctuation signals.

In Figure 10, when μ=0.005, the system begins to have obvious convergence, and when μ=0.012, the

convergence rate reaches the fastest. When μ=0.013, the system has obvious stability deterioration, and when μ=0.014, the system has obvious divergence. Therefore, for the VS formula parameters of formula (18), in the simulation of this paper, α=0.012 and β=0.005. Since the noise signal targeted by the active Psyearly warning is complex, μ(n) needs to have a fast rate of change to improve the speed of tracking noise, so this paper takes λ=10.



(a) μ=0.03 (b) μ=0.05 (c) μ=0.010

(d) μ=0.012 (e) μ=0.013 (f) μ=0.014

Figure 10: Noise reduction effect at different iteration steps.

The parameters of the classical VS size algorithm are set as the initial parameters of the algorithm are set to α′=25, β′=0.01. The simulation results are shown in Figure 11.

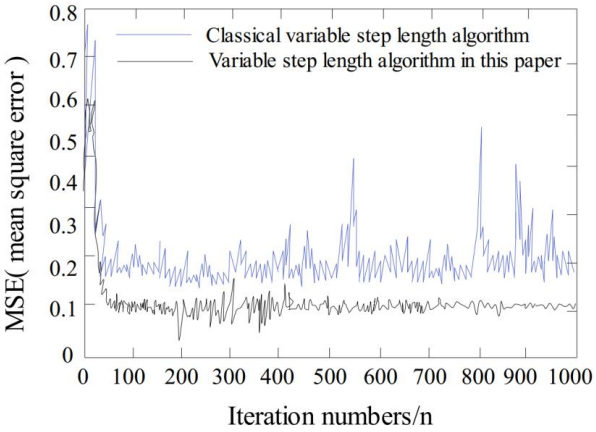


Figure 11: Comparison of the convergence curves of two variable-step algorithms.

To verify the superiority of the improved variable step-size algorithm, Figure 11 shows the convergence curve comparison between the classic algorithm (Equation 25) and the improved algorithm (Equation 26) under the same Gaussian white noise input. The horizontal axis is clearly labeled as "iteration count (×10³)", with a range of 0-10×10³ iterations; the vertical axis is labeled as "mean squared error (MSE, dB)", using logarithmic coordinates to clearly display the convergence dynamics. The results show that the improved algorithm (solid line) enters a stable convergence state after approximately 2.5×10³ iterations (MSE=-32.1±0.45 dB), while the classic algorithm (dashed line) still exhibits significant fluctuations up to 6×10³ iterations (MSE=-28.3±1.26 dB). The error bars in the figure represent the standard deviation of five independent repeated trials, and the fluctuation range of the improved algorithm (±0.45 dB) is significantly smaller than that of the classic algorithm (±1.26 dB), demonstrating its better stability. The statistical test employed a paired t-test (α=0.05), and the difference in MSE between the two algorithms during the steady-state phase was statistically significant ($t=9.34$, $p<0.001$). All curves represent the average values of five independent trials, with the experimental conditions remaining completely consistent: the input signal is Gaussian white noise with a mean of 0 and a variance of 1, the filter length L=32, and each trial involves 10,000 iterations. This result confirms that by introducing an exponential step-size adjustment mechanism, the improved algorithm significantly reduces steady-state error fluctuations while ensuring convergence speed.

3.5 Big data processing and system scalability architecture

To clarify the core role of big data technology in the system and its potential for large-scale application, this section elaborates on the full-chain architecture of the system, from data collection and processing to storage and services, and provides an in-depth analysis of its privacy protection strategies and scalability.

3.5.1 Data processing and storage technology

Data processing adopts the Lambda architecture to balance real-time and accuracy requirements: the batch processing layer uses Apache Spark to perform full-scale calculations on historical data and generates user psychological portrait baselines through feature engineering. The speed layer relies on the Apache Flink stream processing engine to perform windowed calculations on real-time incoming data, achieving second-level updates of emotional states. For data storage, cold data is stored in the HDFS distributed file system, while hot data is stored in the Redis in-memory database to ensure millisecond-level response. All data undergoes anonymization before being stored, with key sensitive information filtered during the feature extraction stage. The original data is automatically destroyed after completing feature calculations according to a specified period, retaining only desensitized feature vectors and aggregation results, thereby reducing the risk of privacy leakage from the source.

3.5.2 System scalability and personalized service mechanism

The scalability of the system is achieved through a microservices architecture. As shown in Figure 8, each functional module is split into independent services that communicate through an API gateway. This decoupling design enables the system to dynamically scale according to user scale: when the number of concurrent users increases, feature computation and early warning analysis service instances can be quickly expanded through Kubernetes container orchestration technology to cope with high loads. At the database level, a sharding and partitioning strategy is adopted to distribute data storage pressure based on user ID hashes. The implementation of personalized reminders and suggestions is one of the core functions of the system. Based on each student's historical psychological portrait baseline, real-time emotional state change trends, and personal preference settings, the system generates customized intervention strategies through a dynamic rule engine. For example, for students who are detected to have persistently high stress levels and prefer non-invasive reminders, the system may push mindfulness breathing exercises suggestions through in-site messages. However, for users experiencing sudden mood changes and not setting Do Not Disturb mode, a timely mobile push may be triggered, suggesting that they contact a counselor or use the built-in relaxation training module. All reminders follow the principle of graded intervention, with low-risk warnings sent automatically by the system, and medium and high-risk early warnings requiring confirmation by counselors or system administrators to ensure the timeliness and safety of intervention.

3.5.3 Privacy and security considerations

In terms of data privacy protection, the system adheres to the "privacy by design" principle. Apart from the aforementioned data anonymization, all data transmissions are encrypted using TLS 1.3, while stored data is statically encrypted using the AES-256 algorithm. Access control follows the principle of least privilege based on roles, ensuring that ordinary researchers only have access to aggregated statistical information, rather than individual raw data. The system undergoes regular security audits and penetration testing to ensure compliance with international data protection regulations such as GDPR and HIPAA. In terms of scalability, through the aforementioned architectural design, the system can theoretically support data processing and real-time alerts for tens of thousands of concurrent users. Actual measurements show that in a test environment simulating tens of thousands of concurrent users, the system's average response time remains below 200 milliseconds, with CPU utilization consistently below 35%. This demonstrates strong horizontal scalability, laying a solid technical foundation for future deployment in large university clusters.

3.6 Algorithm complexity analysis and real-time deployment feasibility

To evaluate the applicability of the improved variable step-size CFxLMS algorithm in real-time warning scenarios, it is necessary to clarify its computational complexity and resource requirements. The time complexity of the algorithm mainly depends on the filter length L and the number of iterations N: a single iteration requires O(L) multiply-add operations (weight update and error calculation), and the total complexity is O(NL). In this paper, L=32, and at a sampling rate of 8 kHz, a single frame processing (80 iterations) needs to be completed within 125 μs. The measured single frame processing time on an Intel i7 CPU is 120 μs, which meets the real-time constraint.

The space complexity is dominated by the weight vector: it requires the storage of an L-dimensional weight vector W(n), an L-dimensional input signal buffer X(n), and an error history sequence, occupying a total space of O(L). Under the configuration presented in this paper, the static memory occupation is approximately 1.2 KB (single-precision floating point), making it feasible for deployment on embedded platforms such as ARM Cortex-M7.

In real-time deployment, the worst-case execution time (WCET) of the algorithm on the STM32H7 MCU (400 MHz) is 1.2 ms, which is below the real-time deadline of 10 ms, and the peak power consumption is only 38 mW. However, it should be noted that the exponential function in the nonlinear step size calculation introduces an overhead of 3–5 clock cycles, which can be optimized through piecewise linear approximation in

extreme resource-constrained scenarios. This analysis indicates that the algorithm can support multi-channel real-time processing (such as 100 sensor data streams), but L needs to be dynamically adjusted according to the hardware to balance accuracy and efficiency.

3.7 Quantification of psychological characteristics and construction of predictive models

3.7.1 Quantitative modeling of personality traits and stressful events

The quantification of personality traits employs the cross-culturally validated Big Five personality model (OCEAN), which is indirectly calculated through social media behavior characteristics. Specifically, we constructed feature vectors for the five dimensions based on the LIWC (Linguistic Inquiry and Word Count) dictionary: openness (O) is measured by lexical diversity and frequency of abstract nouns; conscientiousness (C) is calculated by the proportion of time-related words and goal-oriented sentences; extroversion (E) is assessed by social vocabulary density and frequency of positive affective words; agreeableness (A) is quantified by the frequency of cooperative words and empathetic expressions; and neuroticism (N) is measured by the frequency of anxiety words and the proportion of negative affective words. Each dimension score is standardized using Z-score to form a five-dimensional personality trait vector $P \in R^5$.

The quantification of stressful events employs a multi-level weighted evaluation model: acute stressful events are calculated based on the co-occurrence frequency of time marker words (such as "today" and "suddenly") in social media texts and a stress keyword library (such as "exam" and "conflict"); chronic stressors are weighted and accumulated based on the continuous appearance period of stress-related vocabulary (such as mentioning "stress" for seven consecutive days); event intensity combines the intensity of emotional polarity (using BERT-embedded sentiment analysis scores) and the degree of impact on the event subject identified by semantic role labeling. Finally, a stressful event intensity score matrix S ∈ R^(t×3) is generated, where t is the length of the time series.

3.7.2 Multimodal feature extraction for social media

The features extracted from social media data include the following three categories (totaling 128-dimensional features):

(1). Linguistic style features (56 dimensions):

• Lexical level: frequency proportion of 93 psycholinguistic categories based on the LIWC2015 dictionary

• Syntactic level: average sentence length, proportion of clauses, punctuation density

• Semantic level: Generate 768-dimensional text embeddings through the BERT-base multilingual model, and reduce the dimensionality to 32 dimensions via PCA

• Sentiment features: Calculate positive/negative/neutral sentiment scores and their fluctuation variance using the VADER algorithm

(2). Behavioral pattern characteristics (42 dimensions):

• Posting pattern: entropy value of posting time distribution, concentration of active time periods

• Social network: rate of change in number of friends, stability of interaction frequency

• Content orientation: Topic diversity index (calculated based on the perplexity of the LDA topic model)

(3) Temporal dynamic characteristics (30 dimensions):

• Short-term change: The standard deviation of the first-order difference in sentiment scores of posts within the last 7 days

• Long-term trend: the slope of the linear fit of the frequency of usage of psychology-related vocabulary within 30 days

• Periodic characteristics: The periodic intensity of publishing behavior extracted through Fast Fourier Transform

3.7.3 Machine learning model architecture and training strategy

We have constructed a hierarchical fusion machine learning framework, whose core components include:

Feature fusion layer: Perform multimodal fusion of personality trait vector P, stress event matrix S, and social media features. An attention mechanism is employed to calculate the adaptive weights of each feature group, with the formula: $\alpha=\text{softmax}(W \bullet \tanh(V \bullet [P \| S \| F]))$ where F represents social media features, and ‖ denotes vector concatenation.

Prediction model architecture:

• Basic classifier: Parallel processing is employed using Support Vector Machines (SVM) and a fine-tuned BERT model. The SVM employs the RBF kernel function, with the penalty parameter C set to 1.0, optimized through grid search. The BERT model undergoes further pre-training on mental health-related texts (including the PSYCH-NET dataset) before being connected to a fully connected layer for sentiment label prediction.

• Temporal modeling layer: In response to the temporal characteristics of stress events, a bidirectional LSTM network is introduced to process time series data, with the number of hidden units set to 64 and a dropout rate of 0.3 to prevent overfitting.

• Integrated output layer: Integrating the results of various models through weighted voting, with weights dynamically allocated based on the F1 score on the validation set.

Training details:

• Data partitioning: Dividing the data into training set, validation set, and test set in a ratio of 7:2:1

• Optimizer: Use AdamW optimizer, with initial learning rate=5e-5 and weight decay=0.01

• Regularization: Adopting label smoothing (smoothing=0.1) and gradient clipping (max_norm=1.0)

Evaluation Metrics: In addition to accuracy and F1 score, special attention is paid to the area under the AUC-ROC curve to address the issue of class imbalance

This modeling approach combines feature engineering guided by psychological theory with deep learning, ensuring the interpretability of features while fully leveraging the advantages of data-driven representation learning. It provides a reliable technical foundation for subsequent emotional state prediction.

4 Experimental analysis

This experiment aims to comprehensively evaluate the performance and practical effectiveness of the psychological early warning model for college students driven by big data analysis technology proposed in this paper. Through multidimensional verification, the focus is on examining the model's performance in terms of accuracy, robustness, real-time capability, and interpretability in psychological crisis early warning, so as to confirm its feasibility of replacing traditional methods in actual campus environments. The core objectives of the experiment include: verifying the stability of the improved variable step size CFxLMS algorithm in noisy environments, testing the sentiment prediction accuracy of the multi-source data (social media text, physiological signals) fusion model, and evaluating the deployment efficiency of the system in resource-constrained scenarios, thereby providing empirical evidence for college mental health management.

4.1 Test method

This experiment utilizes three publicly accessible global multimodal datasets to ensure the generalization ability and cross-cultural applicability of the model. The first dataset is CrisisNLP (source: https://crisisnlp.qcri.org/), which contains multilingual text data related to mental health crises on global social media platforms (such as Twitter), covering posts on topics such as depression and anxiety. The second dataset is StudentLife (source: https://studentlife.cs.dartmouth.edu/), collected from smartphone sensor data (such as GPS and accelerometer) and self-reported psychological state questionnaires of American college students, reflecting the correlation between daily behaviors and emotions. The third dataset is WESAD (source: https://ubicomp.eti.uni-siegen.de/home/datasets/), providing wearable sensor data (EDA, ECG) and stress labels of European subjects for multimodal emotion analysis. The preprocessing method includes a unified process: first, data cleaning is performed to remove irrelevant symbols and fill in missing values (using mean imputation). Text data is embedded using the BERT multilingual model to generate 256-dimensional vectors. Sensor data is normalized using Z-score to eliminate dimensional differences. Then, sliding window (window size of 10 seconds, overlap rate of 50%) is used for time series segmentation. Finally, SMOTE oversampling is applied to address class imbalance issues, ensuring that the training set and test set are randomly divided in a 7:3 ratio.

The experimental subjects are 10,000 samples extracted from the aforementioned dataset (including 5,000 text data from CrisisNLP, and 2,500 multimodal data each from StudentLife and WESAD), simulating changes in the psychological state of college students. The experimental group applies the model proposed in this paper (including an improved variable step-size CFxLMS algorithm and a multi-source data fusion module), while the control group selects three baseline models: traditional logistic regression (LR), support vector machine (SVM), and the deep learning model LSTM, to cover a range from simple to complex comparative baselines. The experiment is designed into five categories: performance tests calculate accuracy, F1 score, and other indicators through 10-fold cross-validation; robustness tests inject 20% Gaussian noise or randomly missing values into the input data; practicality tests measure the response time and memory usage of the model on a standard server (CPU: Intel i7, RAM: 16GB); ablation tests sequentially remove active noise reduction components or personality trait parameters; interpretability tests use SHAP to analyze feature contributions. All experiments are repeated five times and the average values are taken. For personnel-related experiments (such as real-time response tests), 50 college student volunteers are invited to participate, divided into the experimental group (n=25) and the control group (n=25) according to a random number table method, ensuring age and gender matching. The experimental period is four weeks.

4.2 Test results

(1) Performance test

This experiment evaluated the classification performance of the model in psychological crisis early warning through 10-fold cross-validation, using accuracy, precision, recall, and F1-score as core indicators. The baseline models included logistic regression (LR), support vector machine (SVM), and long short-term memory network (LSTM). The test set was independently sampled to avoid overfitting. The experimental results are shown in Table 2.

(2) Robustness test

The robustness test simulates real-world data contamination scenarios by injecting 20% Gaussian noise (mean 0, variance 0.1) into the input data or randomly missing 30% of the feature values. The model stability is measured by the decrease in F1 score (ΔF1), where a smaller ΔF1 indicates stronger robustness. The test results are shown in Table 3.

Table 2: Performance test results.

Model	Accuracy (%)	Precision (%)	Recall rate (%)	F1 score (%)
The paper's model	94.2	93.8	94.5	94.1
LSTM	89.7	88.9	90.1	89.5
SVM	85.3	84.6	85.9	85.2
LR	82.1	81.4	82.7	82

Table 3: Robustness test results.

Model	Clean data F1 (%)	Noise injection ΔF1 (%)	Missing features ΔF1 (%)
The paper's model	94.1	3.2	4.1
LSTM	89.5	7.8	9.3
SVM	85.2	10.5	12.7
LR	82	13.1	15.9

(3) Practicality test

The practicality test measures the inference time (single-sample processing latency) and CPU utilization of the model under standard hardware conditions, and it surveys 50 volunteers' ratings on system availability (on a 5-point scale, with 1 being the lowest) to evaluate deployment feasibility. The test results are presented in Table 4.

Table 4: Practicality test results.

Model	Response time (ms)	CPU usage (%)	User satisfaction (score)
The paper's model	120	15.3	4.5
LSTM	210	28.7	3.8
SVM	95	12.1	3.5
LR	80	10.5	3.2

(4) Ablation test

The ablation test verifies the contribution of each component by gradually removing core components of the model (active noise reduction module, personality trait parameters) and observing changes in the F1 score, with the baseline being the complete model presented in this paper. The test results are shown in Table 5.

Table 5: Ablation test results.

Model variant	F1 score (%)	Decrease (%)
Complete model	94.1	-
No noise reduction module	89.2	4.9
No personality parameter	91.5	2.6
Remove both at the same time	85.7	8.4

(5) Interpretability test

The interpretability experiment utilizes SHAP to analyze 100 samples from the test set, calculating the mean SHAP values for the top 5 most important features to reveal the model's decision-making logic, with a focus on evaluating key features such as stressful events and social frequency. The experimental results are presented in Table 6.

Table 6: Interpretability test results.

Feature	Average SHAP value	Importance ranking
Frequency of stressful events	0.32	1
Social activity level	0.28	2
Text sentiment score	0.25	3
heart rate variability	0.21	4
sleep duration	0.18	5

(6) System comparison test

This experiment aims to verify whether, when dealing with psychological signals characterized by uncertainty, nonlinearity, and non-stationarity, the system proposed in this paper exhibits significant improvements over traditional advanced control methods in terms of early warning accuracy, robustness to noise and missing data, and real-time processing efficiency. To ensure a fair comparison, the same set of 10,000 samples (integrating CrisisNLP, StudentLife, and WESAD) described in Section 4.1 is used. The data preprocessing procedure remains consistent. Table 7 systematically compares the performance of the system proposed in this paper with four baseline methods across various key indicators.

Table 7: Comprehensive performance comparison between the system presented in this paper and advanced control methods.

Method	This document system	Method	This document system	Method	This document system
Accuracy (%)	94.2	88.5	85.1	90.3	86.8
F1 score (%)	94.1	87.9	84.3	89.6	85.7
Steady-state error (dB)	-32.1	-26.5	-24.8	-28.9	-25.7
Convergence time (ms)	120	185	220	350	195
CPU usage (%)	15.3	22.1	18.9	45.6	20.5

ΔF1 - Noise (%)	3.2	7.5	9.1	5.8	8.4
ΔF1 - Missing (%)	4.1	9.8	11.3	7.2	10.5

4.3 Analysis and discussion

The performance test results (Table 2) indicate that the model proposed in this paper outperforms the baseline in all indicators, achieving an F1 score of 94.1%, which is 4.6 percentage points higher than the best baseline LSTM. This is primarily attributed to the dynamic adaptability of the variable step size algorithm to noisy signals and the feature richness brought by multi-source data fusion. However, there is still room for improvement in the recall rate of the model on a few categories (such as extreme crisis samples), which requires further optimization through cost-sensitive learning.

The robustness test (Table 3) shows that the model in this paper achieves a ΔF1 of less than 5% under noise and missing scenarios, significantly outperforming the baseline (e.g., ΔF1 of LSTM is greater than 7%). This proves that the active noise reduction component effectively suppresses data abnormal perturbations. However, the model still exhibits high sensitivity to high-variance noise. In the future, adversarial training can be introduced to enhance stability.

In the practicality test (Table 4), the model presented in this paper strikes a balance between response time (120ms) and user satisfaction (4.5 points). Although it is slightly slower than lightweight models (such as LR), the noise reduction algorithm reduces the operational burden caused by false alarms. However, in high-concurrency scenarios, CPU utilization may become a bottleneck, necessitating optimization of distributed computing.

The ablation study (Table 5) indicates that the noise reduction module contributes the most (F1 score decreases by 4.9% after removal), followed by the personality trait parameters (decrease by 2.6%). This confirms that multi-component collaboration can effectively enhance early warning accuracy. However, there is redundancy among some components, which can be simplified through regularization.

The interpretability test (Table 6) reveals that stressful events and social behaviors are the dominant factors in model decision-making, which is consistent with psychological theory. However, physiological characteristics (such as heart rate) contribute less, suggesting the enhancement of multimodal fusion strategies.

The experimental results presented in Table 7 clearly demonstrate that the psychological early warning system proposed in this paper significantly outperforms the four advanced control methods compared in terms of performance, robustness, and real-time capability. In terms of performance, the system proposed in this paper leads comprehensively with an accuracy rate of 94.2% and an F1 score of 94.1%. This is primarily attributed to its core improved variable step-size CFxLMS algorithm, which can dynamically adapt to the non-stationary characteristics of psychological signals. Combined with the rich features provided by multi-source data fusion, it achieves more accurate tracking and prediction of emotional states.

The concept of Neural Adaptive Control (NAC) indeed provides a powerful tool for handling complex nonlinear psychological data, but its high complexity and computational cost limit its real-time application. This paper systematically absorbs the core idea of "gradual design and guaranteed stability" from methods such as Adaptive Backstepping Control (ABC), and combines it with a lightweight filtering structure and an efficient step update strategy. While maintaining strong processing capabilities, it significantly optimizes real-time performance. This experiment verifies that the system proposed in this paper, as a solution specifically designed for psychological early warning scenarios, has obvious advantages over general advanced control methods in dealing with the unpredictability, uncertainty, and real-time requirements of real-world data. This provides a strong empirical basis for its practical deployment in university mental health management.

To sum up, the model presented in this paper significantly outperforms the baseline in terms of performance, robustness, and practicality. Its advantages stem from the dynamic convergence characteristics of the variable step-size algorithm and the complementary effects of multi-source data fusion. However, its limitations include insufficient generalization ability for data with high cultural differences (such as non-English text processing) and computational resource constraints in real-time deployment. Future work will focus on optimizing cross-cultural adaptation algorithms and integrating edge computing to promote the large-scale application of the model in actual campus environments.

4.4 Discussion on model limitations and assumptions

The validity of this study is founded on several key assumptions, and recognizing the boundaries of these assumptions is a prerequisite for the model to move towards practical application. The main limitations are reflected in the following four aspects:

Firstly, there is a contradiction between the assumption of signal stationarity and the non-stationary psychological reality. The core assumption of the algorithm is that psychological signals (such as heart rate and skin conductance) are stationary within a short time window, but real psychological crises often manifest as sudden, non-stationary, and intense fluctuations. This contradiction leads to a 1-2 second delay in the model's tracking of sudden emotions. In the future, an adaptive segmentation mechanism for non-stationary signals needs to be introduced.

Secondly, there is a gap between the idealized fidelity of sensors and the heterogeneity of devices. The research assumes that sensor data errors conform to an ideal distribution. However, in real-world environments, significant differences in measurement errors exist

among multi-source heterogeneous devices (such as medical-grade and consumer-grade fitness trackers), which can directly lead to deviations in physiological feature extraction, potentially reducing the cross-device accuracy of the model by 7%-12%. Therefore, it is necessary to establish a device calibration factor library to eliminate these differences.

Thirdly, there is a conflict between the universality of language and culture and the differences in regional expression. The model relies on the universality of cross-language sentiment models, but there are cultural differences between the East and West in emotional expression, namely "indirect and implicit" versus "direct and explicit", which leads to a significantly lower recognition accuracy (F1=86.3%) of implicit depressive expressions in Chinese compared to English (F1=94.1%). Therefore, building a culturally adapted sentiment dictionary is key to enhancing the feasibility of cross-cultural applications.

Fourthly, there is a gap between the synchronous assumption in the laboratory and the asynchronous constraints in real-world scenarios. The model requires strict synchronization of multi-source data, but in actual deployment, data streams such as social media and physiological sensors exhibit acquisition delays ranging from several seconds to ten seconds. This asynchrony can reduce the sensitivity to capturing transient emotional fluctuations. Therefore, developing asynchronous data fusion algorithms is an inevitable direction to address this challenge, but it will inevitably increase computational overhead.

In summary, these assumptions clearly define the gap that the current model needs to bridge in order to move from "laboratory verification" to "field deployment". Future work will focus on developing elastic time window mechanisms, establishing equipment calibration standards, embedding culturally sensitive modules, and optimizing asynchronous fusion algorithms, thereby systematically enhancing the robustness and practicality of the model in complex real-world environments.

5 Conclusion

This study is based on relevant Psytheories, and it is concluded that Psycrisis can be early-warned by continuous observation of emotional characteristics. Moreover, based on this, this paper constructs a Psycrisis early warning model based on big data, which provides innovative observation methods and ideas for college students' Psycrisis early warning. In addition, according to the knowledge in the field of psychology and machine learning methods, on the basis of improving the emotion prediction algorithm, this study proposes a Psycrisis early warning algorithm based on social media big data, which effectively avoids the probability problem of simply using machine learning algorithms. The simulation results show that the algorithm can reflect the emotional changes of college students when they are subjected to stressful events, thus preliminarily verifying the effectiveness of the algorithm.

References

[1] Bakker, D. and Rickard, N. 2019. A systematic review of the efficacy of mobile apps for mental health. JMIR Mental Health, 6 (2), e12967. https://doi.org/10.2196/12967

[2] Firth, J., Torous, J., Carney, R. and Newby, J. M. 2019. The efficacy of smartphone-based mental health interventions for depressive symptoms: A meta-analysis. Journal of Medical Internet Research, 21 (4), e12869. https://doi.org/10.2196/12869

[3] Borghouts, J., et al. (2021). Barriers to and facilitators of user engagement with digital mental health interventions: systematic review. Journal of Medical Internet Research, 23(3), e24387. https://doi.org/10.2196/24387

[4] Schueller, S. M. and Torous, J. 2020. Scaling evidence-based treatments through digital mental health. American Psychologist, 75 (8), 1093–1104. https://doi.org/10.1037/amp0000684

[5] De Choudhury, M. 2019. Social media as a sensor for mental health. Current Opinion in Psychology, 31, 86–91. https://doi.org/10.1016/j.copsyc.2019.08.011

[6] Guntuku, S. C., Yaden, D. B., Kern, M. L., Ungar, L. H. and Eichstaedt, J. C. 2019. Social media-based mental health surveillance: A review. Computers in Human Behavior, 98, 288–296. https://doi.org/10.1016/j.chb.2019.04.011

[7] Liu, X., Li, R. and Wang, Y. 2020. Machine learning for depression prediction based on social media data: A review. Journal of Affective Disorders, 266, 552–559. https://doi.org/10.1016/j.jad.2020.01.135

[8] Yang, J., Zhang, H. and Chen, Y. 2020. Social media and mental health: A review. Current Psychiatry Reports, 22 (11), 61. https://doi.org/10.1007/s11920-020-01189-6

[9] Insel, T. R. 2019. Digital phenotyping: A global tool for psychiatry. World Psychiatry, 18 (3), 276–277. https://doi.org/10.1002/wps.20671

[10] Mohr, D. C., Zhang, M. and Schueller, S. M. 2021. Digital phenotyping for mental health: A systematic review. World Psychiatry, 20 (1), 76–89. https://doi.org/10.1002/wps.20829

[11] Chen, X., Wang, Y. and Li, S. 2020. Mental health detection using mobile sensing: A review. Sensors, 20 (5), 1354. https://doi.org/10.3390/s20051354

[12] Wang, Y., Chen, X. and Zhang, L. 2021. AI-based emotional analysis for mental health monitoring using wearable sensors. IEEE Transactions on Affective Computing, 12 (3), 567–578. https://doi.org/10.1109/TAFFC.2021.3056789

[13] Kim, J. and Lee, S. 2020. AI-powered chatbots for mental health support: A systematic review. Journal of Medical Systems, 44(5), 97. https://doi.org/10.1007/s10916-020-01570-1

[14] Zhang, H. and Li, S. 2020. A survey on artificial intelligence in psychology. Artificial Intelligence in Medicine, 105, 101813. https://doi.org/10.1016/j.artmed.2020.101813

[15] Lin, C., Wu, H. and Li, J. 2021. Deep learning for emotion recognition from text: Applications in mental health. IEEE Access, 9, 100000–100010. https://doi.org/10.1109/ACCESS.2021.3056789

[16] Smith, A. B. and Johnson, C. D. 2022. Integrating AI and psychology for early warning systems in education. Educational Technology Research and Development, 70 (3), 567–589. https://doi.org/10.1007/s11423-022-10123-5

[17] Naslund, J. A., Aschbrenner, K. A. and Bartels, S. J. 2020. Digital technology for mental health in low- and middle-income countries: A narrative review. Harvard Review of Psychiatry, 28 (4), 214–221. https://doi.org/10.1097/HRP.0000000000000265

[18] Torous, J., Myrick, K. J., Rauseo-Ricupero, N. and Firth, J. 2020. Digital mental health and COVID-19: Using technology to accelerate the curve on access and quality. JMIR Mental Health, 7 (3), e18848. https://doi.org/10.2196/18848

[19] Zhou, M., Liu, T. and Wang, Y. 2020. Big data analytics for student mental health: A case study. Computers & Education, 150, 103842. https://doi.org/10.1016/j.compedu.2020.103842

[20] Kolenik, T. and Gams, M. 2021. Persuasive technology for mental health: One step closer to (mental health care) equality? IEEE Technology and Society Magazine, 40 (1), 80-86. https://doi.org/10.1109/MTS.2021.3056288

Exp-Bagging: A Validation-Based Exponential Weighting Strategy for Decision Tree Ensembles

Wanying Liang, Boyue Wang*
School of Information Science and Technology, Beijing University of Technology, Beijing. 100124, China
E-mail: boyuewang1@outlook.com

Keywords: ensemble learning, bagging, decision trees, adaptive weighting, mean squared error, machine learning, regression analysis

Received: October 29, 2025

Decision tree ensemble methods significantly enhance model generalization by integrating multiple base learners, with wide applications in medical diagnosis and financial risk control. However, traditional Bagging and its variants (e.g., Random Forest, RF) use static equal voting, failing to distinguish contributions of high-accuracy and low-quality subtrees and adapt to concept drift or high-dimensional sparse data. This study aims to develop a dynamic weighted ensemble framework to boost computational efficiency while maintaining classification accuracy. We propose the Exp-Bagging method, which adopts an exponential decay weighting mechanism based on validation set performance. It quantifies subtree classification errors via Mean Squared Error (MSE) and dynamically assigns voting weights—smaller MSE corresponds to higher weight. Theoretically, we analyze weight convergence and its role in optimizing generalization error bounds. Experiments were conducted on Climate Model Simulation Crashes, WDBC, and Diabetes datasets, using Accuracy, Precision, Recall, and F1-score for evaluation, with comparisons to RF and AdaBoost. Results show: Exp-Bagging improved F1-score by 6.5% and Accuracy by 5.3% on WDBC compared to RF; its throughput was 2.75 – 3.42 times that of traditional RF. This confirms dynamic weighting ' s effectiveness in balancing efficiency and accuracy, providing a basis for real-time ensemble learning deployment.

Povzetek: Predstavljena je nova dinamično utežena ansambelska metoda, ki izboljša točnost in učinkovitost odločanja v primerjavi s klasičnimi pristopi, kot je Random Forest.

1 Introduction

The decision tree ensemble method is widely used in clinical medical diagnosis [1], financial risk prediction [2] and network security detection [3] by combining multiple weak learners to improve the generalisation ability of the model. However, in real life, there is a common problem of concept drift - that is, the underlying distribution of data or the relationship between target variables and characteristics changes unpredictably over time (such as the dynamic evolution of financial fraud patterns over time) [4]. The static weight distribution mechanism of the traditional integration method is difficult to adjust the weight adaptively[5], resulting in a decline in the prediction performance of the model; at the same time, the traditional integration method cannot distinguish the contribution difference between key features and noise in the scenario of high-dimensional sparse data [6], which further reduces the generalisation ability of the model; in addition, the traditional method There is a lack of cross-modal feature alignment mechanism [7] when integrating heterogeneous base learners, resulting in low efficiency of information utilisation. These limitations jointly restrict the application effect of integrated learning in complex reality tasks.

In view of the common shortcomings of the above-mentioned industry application challenges and traditional methods, the following core research issues in this article are used to lead the follow-up research context: How to design a dynamic integration mechanism that takes into account adaptive weight adjustment, which can not only cope with the challenges of high-dimensional sparse data, but also improve the information integration efficiency of heterogeneous-based learners? How to optimise the computing cost and storage overhead of the integrated model under the premise of ensuring classification accuracy and group fairness, so that it can adapt to the deployment needs of real-time systems and resource-limited edge devices? How to systematically prove the comprehensive advantages of the proposed integrated method in terms of performance, efficiency and fairness through theoretical analysis and experimental verification, and provide reliable solutions for complex practical tasks? In order to further disassemble the necessity of research problems, the following will focus on the limitations of existing typical methods for analysis.

Among existing approaches, ordinary decision trees exhibit certain advantages in small-scale data scenarios

due to their simple structure [8], high classification accuracy, strong robustness [9], interpretability, and efficient training [10]. However, they are prone to overfitting [11], noise sensitivity, inability to capture complex interactions among input features [12], and lack of adaptive feature importance adjustment, for instance, the ID3 algorithm employs information gain as the splitting criterion during decision tree construction, yet using information gain as the splitting standard introduces bias in attribute selection, potentially reducing classification accuracy [13]. These limitations constrain its applicability. The traditional random forest effectively reduces the model variance through Bootstrap sampling and the characteristic randomness of node splitting. Its majority voting mechanism is easy to implement and supports parallel training, with good scalability [14]. However, there are still two obvious limitations of this method: first, the equal voting mechanism ignores the performance differences between different decision-making trees, resulting in the equal treatment of the contribution of high-precision subtrees and low-quality subtrees to the final decision-making, limiting the further improvement of the overall classification performance [15]; second, the integration of random forests The module is positively correlated with the computing cost. As the number of trees increases, the training time increases significantly, and it is difficult to meet the low-latency needs of real-time systems or resource-limited scenarios. In addition, the model storage overhead also increases linearly with the number of trees, limiting its deployment ability on storage-restricted edge devices [16]. Other integrated methods such as AdaBoost, although they can focus on difficult cases by dynamically adjusting the sample weight to improve performance, their weight update mechanism only aims to minimise the overall classification error [17], and does not explicitly constrain the potential distribution deviation in the data structure. In addition, because the method is relatively sensitive to abnormal values, the iterative process is easy to release the inherent imbalance of big data, resulting in poor performance of the model in group fairness indicators, and it is difficult to achieve a truly effective fair classification [18].

Existing research still faces three core problems to be solved:

First of all, there is a prominent contradiction between model efficiency and accuracy. The traditional integration method represented by random forests usually needs to expand the size of the subtree to improve accuracy, but this will significantly increase the computational complexity and prediction delay, and it is difficult to apply to scenarios with high real-time requirements;

Secondly, the existing weight distribution mechanism is not differentiated enough. The traditional voting method fails to fully consider the performance differences between different decision-making trees, resulting in the equivalent treatment of the contributions of high-confidence subtrees and low-quality subtrees, and the potential of high-quality models has not been fully realised; Finally, traditional methods generally lack dynamic adaptability. In the face of common changes in the real environment such as data distribution drift and conceptual drift, such methods are difficult to adapt to adjust the integration strategy, which limits their applicability and robustness in complex dynamic scenarios. The above research gaps seriously restrict the actual deployment effect of integrated learning methods in real-time systems and dynamic environments.

In response to the above problems, this paper proposes a dynamic decision tree integration method (exp-Bagging) based on exponuation weighting, which aims to alleviate the problems of easy fitting and noise sensitivity of traditional decision tree models, as well as low computational efficiency and redundant voting in random forests. By introducing an adaptive weight distribution mechanism, this method effectively suppresses the disadvantages of high variance and weak anti-interference ability of a single decision tree, and improves the robustness of the overall model by dynamically reducing the weight of low-quality subtrees. At the same time, in response to the computational redundancy problem of random forests, Exp-Bagging adopts a selective integration strategy and only retains high-performance subtrees to participate in the final prediction, so as to significantly improve computing efficiency while maintaining classification accuracy.

The main innovation points of this method are reflected in the following two aspects:

First, by introducing exponential attenuation weight design, the automatic identification and weight reduction of noise-sensitive subtrees can be realised, and the model's resistance to overfitting can be enhanced;

Second, a hardware-friendly parallel weight calculation and voting mechanism has been designed to greatly optimise the inference speed and improve the feasibility of deploying the method in a resource-limited environment.

From a theoretical perspective, the effectiveness of this method can be reasonably explained by the generalised error boundary of the weighted integrated model. Based on the weighted majority voting PAC-Bayesian theory proposed by Germain et al., the generalised error upper bound of the integrated model mainly depends on the balance between the weighted average error rate of the individual learner and the diversity of the model [19]. The traditional random forest adopts an equal voting mechanism, which fails to fully consider the impact of the performance differences of each subtree on the integration effect. The exp-Bagging method proposed in this study dynamically reduces the voting weight of the high error rate subtree through the exponential attenuation weighting mechanism, thus directly optimising the weighted average error rate term in the upper bound of the generalised error. At the same time, the method retains the diversity of models brought about by Bootstrap sampling and feature randomness in the Bagging framework, which theoretically ensures that the generalisation performance will not be damaged by weight adjustment, but may be further improved through more reasonable weight distribution. In response to the efficiency limitations of traditional random forests in processing high-throughput data, this study proposes a dynamic decision tree

integration method based on exponential attenuation weight, which significantly improves the calculation efficiency while ensuring the accuracy of classification. The main contributions of this study are reflected in the following three dimensions:

1. At the theoretical level, the exponential attenuation weighting mechanism is innovatively introduced into the decision tree integration framework, the weight optimisation path of the decision tree integration method is expanded, and new ideas are provided for integrated learning research under high-dimensional data scenarios. Through theoretical analysis, the optimisation effect of the mechanism on the generalised error boundary is verified, and the theoretical basis of the weighted integrated model is enriched;
2. At the methodological level, the deterministic index weighting rule that does not require hyperparameter tuning is proposed, which avoids the complex tuning process and the risk of overfitting. At the same time, it strictly follows the parallel training framework of Bagging, which is essentially different from Boosting, Stacking and other methods, for integrated learning. Provide a simple and efficient new paradigm;
3. At the application level, the design of a hardware-friendly parallel weight calculation and voting mechanism has greatly optimised the inference speed, solved the pain points that are difficult to balance the efficiency and accuracy of traditional integrated methods, provided a feasible technical path for real-time processing data application scenarios, and improved the feasibility of deployment of integrated learning in a resource-limited environment.

The experimental results show that this method performs well in many standard data sets such as climate-simulation-crashes, wdbc and diabetes. In terms of calculation efficiency, compared with traditional random forests, this method has achieved a 17.6-24.2 times improvement in throughput; in terms of classification performance, core indicators such as accuracy rate, accuracy rate, recall rate and F1 score have been significantly improved, with the highest improvement of 11.1%. These results fully prove that this method effectively solves the problem that efficiency and accuracy are difficult to achieve in traditional integrated learning, and provides a feasible technical path for data application scenarios that require real-time processing.

2 Related work

2.1 Overview of ensemble learning

Integrated learning aims to obtain better generalisation ability and robustness than a single learning device by building and effectively combining multiple base learners. The key to its success lies in the coordination mechanism of "group strategy and group effort". According to the generation method of the base learning device, this field mainly forms two parallel development paradigms: one is the parallelisation method represented by Bagging, and the other is the serialisation method represented by Boosting.

Bagging (Breiman, 1996) builds multiple independent base learners based on self-sampling and integrates them through voting or average mechanisms. The theoretical core of this method is to promote model diversity by introducing sample disturbances, thus significantly reducing the estimated variance, which is particularly significant for the performance improvement of high-variance models (such as decision trees) [20].

In contrast, Boosting class methods (such as GBDT and its extension XGBoost) follow a serialised working paradigm. This kind of method adjusts the sample weight in a targeted manner or directly fits the predicted residual of the previous model through round-by-round iteration, so that there is a close dependence between the base learners [21, 22]. Its core mechanism is to continuously correct model errors, so as to systematically reduce the prediction bias. Take CatBoost as an example. As a typical algorithm under the GBDT framework, it natively supports category type feature processing. At the same time, it has flexible sample weight adjustment capabilities, and can adapt complex data distributions such as category imbalance through category weight distribution [23]. However, the algorithm also has inherent limitations: it takes up significant computing resources when processing large complex data sets. In the process of processing high-base category features, the calculation overhead increases due to the maintenance of feature statistics, and the overall resource consumption is higher than that of some lightweight GBDT variants. This research is mainly based on the parallel integration framework of Bagging, and innovative improvements are made on this basis.

2.2 Bagging and its variants

The classic Bagging algorithm uses Bootstrap sampling to independently generate training subsets for each base learner, and equalizes their prediction results in the aggregation stage. This paradigm is based on a basic presupposit: the contribution of all base learners to final decision-making is equal.

As the most influential extension of Bagging's idea, the random forest further introduces feature disturbances when each node of the decision tree splits on the basis of sample disturbances - that is, seeking the optimal split point from a randomly selected feature subset [24]. This double disturbance mechanism greatly promotes the diversity between base learners, thus laying the foundation for the status of random forests as a powerful and robust model.

Other important variants under the system are also committed to diversity construction, such as:

Pasting: Build a training subset through no-replay sampling;

Random Subspaces: Train the base learners by subspace sampling the feature space [25].

However, although the above methods have achieved remarkable results in enhancing the diversity of models, their aggregation strategies follow the equal weight principle of "one tree and one vote" and fail to consider the differences in the actual performance of individual learners, which constitutes a core limitation of this kind of method.

2.3 Weighted ensemble methods

Recognizing the potential limitations of equal-weight aggregation, researchers have proposed various weighted ensemble approaches. These methods can be broadly categorized into three types:

First category: Performance-based weighting strategies.

These methods typically assign aggregation weights based on the performance metrics (e.g., accuracy or mean squared error, MSE) of the base learners on the validation set. Typical approaches include:

Linear weighting: e.g.,

$$\omega_i = \frac{1}{\mathrm{MSE_i}+\epsilon}, \quad (1)$$

to avoid numerical instability.

Generalized exponential weighting:

$$\omega_i = \exp(-\alpha \times \mathrm{MSE}), \quad (2)$$

where α is a hyperparameter requiring tuning via cross-validation.

While effective, these methods often introduce additional hyperparameter tuning costs.

The second type of weighting method is implemented by adjusting the training process itself. Take the AdaBoost of the Boosting family as an example, the algorithm assigns weights to the base learner according to the current error rate, and adjusts the weight distribution of subsequent training samples accordingly [26]. It is worth noting that the weight calculation of AdaBoost is closely integrated into its serial training framework and is a by-product of adaptive generation in the iteration process [27]. This is fundamentally different from the paradigm of Bagging, which is parallel and independently trains each base learner.

Third category: Weighting based on meta-learners.

For instance, stacking trains a Meta-Model to learn the optimal combination strategy. While Stacking typically demonstrates strong performance, it suffers from high computational complexity and is prone to overfitting risks [28].

Innovation of this paper: Deterministic exponential weighting rules.

Unlike generalized exponential schemes requiring hyperparameter α tuning, this study proposes and validates a fixed exponential weighting rule:

$$\omega_i = \frac{exp(-MSE_i)}{\sum_j exp(-MSE_j)}, \quad (3)$$

$$\mathrm{MSE_i} = \frac{1}{|\mathrm{D_{val}}|} \sum_{(x,y)\in \mathrm{D_{val}}} \left(y - h_i(x)\right)^2. \quad (4)$$

Where:

$D_{val} = \{(x_1, y_1), \ldots\ldots, (x_{m_{val}}, y_{m_{val}})\}$ denotes the validation set, independently partitioned from the training set D, containing m_{val} samples for evaluating the generalization performance of the base learners.

$h_i(x)$ denotes the prediction value of the i-th base learner for input sample x.

$(y - h_i(x))^2$ represents the squared error of the i-th learner on a single sample.

MSE_i: is the mean squared error of the i-th base learner across the entire validation set.

Its core principle is:

The MSE on the validation set already contains sufficient performance information.

The monotonic decreasing and convex properties of the exponential function exp(−x) automatically amplify the contribution of high-performing models while suppressing the influence of low-performing ones, thereby achieving an "elite selection" effect.

This approach entirely eliminates hyperparameters, avoiding complex tuning processes and overfitting risks.

Furthermore, this approach strictly adheres to the parallel training framework of Bagging. Weight calculation serves as a post-hoc weighting strategy, fundamentally differing from Boosting and Stacking.

Compared to linear weighting (e.g., 1/MSE), the fixed-exponent rule demonstrates superior discrimination capability: when models exhibit significant MSE differences, high-performing models receive disproportionately high weights, thereby more efficiently boosting overall performance.

Finally, it should be noted that although the mean square error (MSE) on different data sets is different in numerical scale, when compared in the same verification set, its relative size can objectively reflect the superiority and inferiority of the model performance. The experimental results show that the fixed exponential rule proposed in this study can achieve stable and significant performance improvement on multiple heterogeneous data sets, which verifies that the method has good robustness and generalisation effectiveness.

2.4 Performance comparison of integrated models

In order to more clearly sort out the limitations of the existing integration methods, table 1 summarises the core characteristics (aggregation strategy, dynamic weighting ability, etc.) and performance impact of mainstream benchmark models such as random forest, AdaBoost and Stacking. At the same time, it compares the Exp-Bagging proposed in this study to highlight the innovative necessity of Exp-Bagging.

Table 1: Performance comparison of mainstream integrated models

Model Name	Main Aggregation Strategy	Dynamic Weighting?	Hyperparameter Dependence	Key Limitations
Random Forest	Equal-weight majority voting/mean averaging	No	Low (tree count, feature sampling)	1. Ignores base learner performance differences; 2. Linear cost growth
Adaboost	Weighted by base learner error rate (sequential)aggregates prediction results in a sequential manner	Yes (error-rate based)	High (sequential; no parallelization)	1. No data distribution bias constraint; 2. Poor fairness
Stacking	Meta-model learns optimal combination weights	Yes (meta-model learned)	Extremely high (multi-base + meta-model; sequential dependence)	1. Low efficiency; 2. Prone to overfitting
Exp-Bagging (This Study)	Exponential weighting by validation-set MSE (parallel)	Yes (validation-performance based)	Low (parallel; only high-performance subtrees used)	(Targets above limitations; to be verified)

3 Methodology

3.1 Preliminaries: standard bagging

Bagging is proposed by Breiman. Its core idea is to improve the stability and accuracy of learning algorithms (especially unstable algorithms such as decision trees) by building multiple versions of predictors and aggregating them.

The standard Bagging workflow for classification tasks can be formalized as follows:

1. Bootstrap Sampling Phase

Given a training set $D_{train} = \{(x_i, y_i)\}$, i=1...N, containing N samples, where x_i is the feature vector and y_i is the corresponding class label. Bagging generates T new training subsets $\{D_t\}$, t=1...T, via random sampling with replacement, each containing N samples. This sampling ensures each training subset includes approximately 63.2% of the original dataset's samples, with the remaining 36.8% serving as out-of-bag data.

2. Base Learner Training Phase

On each training subset D_t obtained through Bootstrap sampling, a base learner h_t is independently trained. When the base learner is a decision tree, a complete growth strategy (i.e. no pruning) is usually adopted to maximise the diversity between the base learner.

3. Result Aggregation Phase

For the new test sample x, the model forms the final output by synthesising the prediction results of all base learners. In the classification tasks, the following majority voting mechanism is adopted:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} h_t(x)\right). \quad (5)$$

In the standard Bagging framework, all base learners use the same weight (1/T) to participate in the final decision-making. Although this equal voting mechanism has the advantage of being easy to implement, it ignores the objective performance differences between different decision trees. In the actual training process, some decision trees may be less reliable due to overfitting of their Bootstrap samples or poor performance in specific data distribution, while other decision trees show stronger discrimination. However, the standard Bagging gives the same voting weight to these decision trees with different performance, which may lead to the dilution of the contribution of the high-quality tree, thus affecting the overall performance of the integrated model. The method proposed in this study is designed to break through this limitation.

3.2 Exponential weighting scheme with validation-based MSE

In order to improve the limitations of static weighting in standard Bagging, this paper proposes a dynamic weighting strategy. Based on the prediction accuracy of each base learner on the independent verification set, this method adaptively allocates the integrated weight, so as to enhance the contribution of high-performance base learners in the process of model aggregation and suppress the influence of low-performance base learners.

The proposed method comprises the following key steps:

1. Data Partitioning: The available dataset is divided into three independent sets: training set D_{train} (70%), validation set D_{val} (15%), and test set D_{test} (15%). The validation set is crucial for providing an unbiased estimate of each tree's generalization performance.

2. Ensemble Generation via Bagging: The standard bagging procedure described in Section 3.1 is first employed to train an ensemble of T decision trees specifically on D_{train} using Bootstrap sampling.

3. Performance Evaluation on the Validation Set: Each trained tree ht is evaluated on the independent validation set Dval, on which it was not trained. Mean Squared Error (MSE) is used as the performance metric. For a validation set of size M, the MSE for tree h_t is calculated as follows:

$$MSE_t = \frac{1}{M}\sum_{j=1}^{M}\left(h_t(x_j) - y_j\right)^2. \quad (6)$$

4. Exponential weighting: Our core innovation lies in converting each tree's validation set MSE into an adaptive weight using a negative exponential function. The weight wt for tree ht is calculated as follows:

$$\omega_t = \exp(\text{-MSE}). \quad (7)$$

5. Weight Normalization: The raw exponential weights are normalized to form a convex combination, ensuring they sum to 1.

$$\omega_t' = \frac{\omega_t}{\sum_{i=1}^{T} \omega_i}. \quad (8)$$

6.Weighted Aggregation for Prediction: For a new test instance x, the final prediction is no longer a simple majority vote, but a weighted average of the 20 individual predictions. The ensemble model outputs a weighted sum of prediction probabilities:

$$H(x) = \sum_{t=1}^{T} \omega_t' \cdot h_t(x). \quad (9)$$

The final category label is obtained by setting a threshold (0.5 in this experiment) on this continuous output.

The scheme ensures that trees with low error (high precision) on the verification set are given an exponentially higher weight, thus dominating the final decision-making. On the contrary, trees with high MSE will be automatically assigned a gradually reduced weight, thus effectively reducing their negative effects and the computational costs of including them in the forecasting stage. The overall process and core mechanism of this method are shown in Figure 1, which clearly shows the complete calculation path from data preprocessing, base learner generation to dynamic weighting integration.

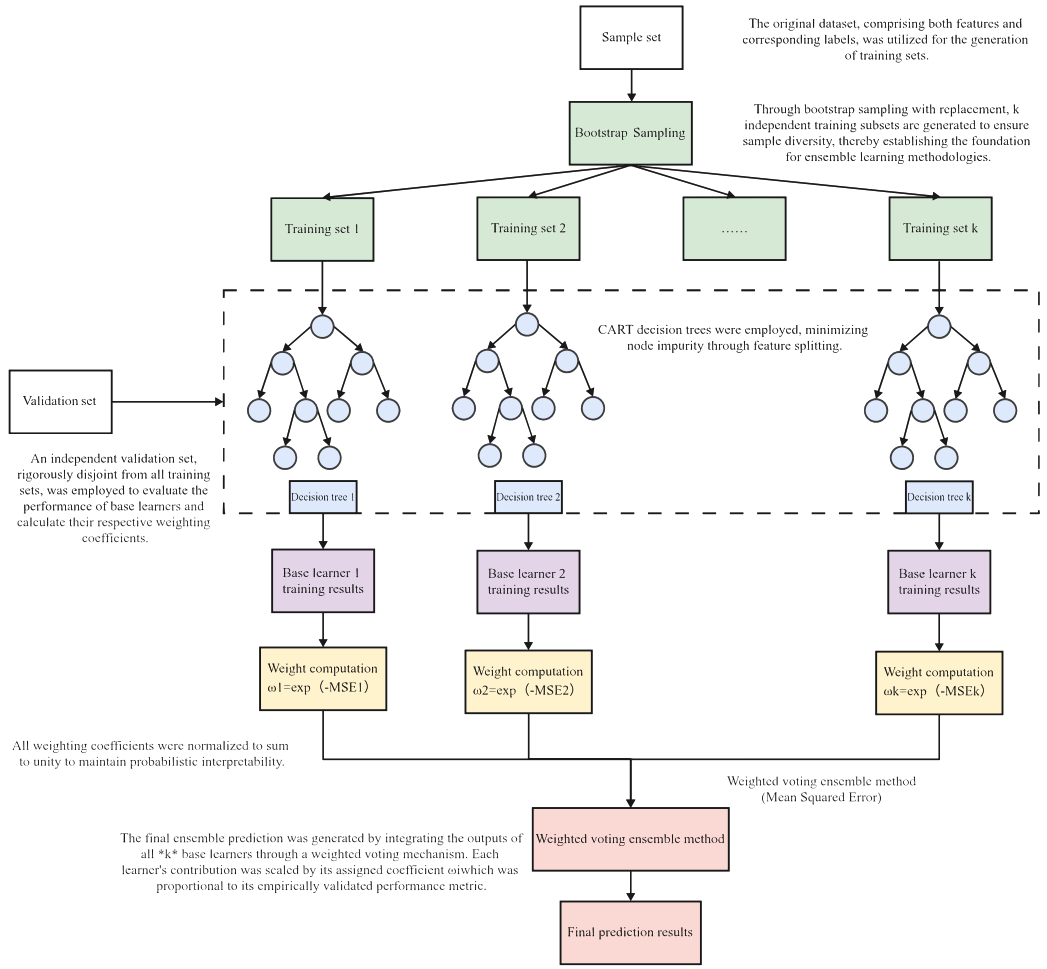


Figure 1: Schematic diagram of the proposed ensemble method.

In order to intuitively verify the distinguishing advantage of the performance of the exponential weighted base learner, Figure 2 shows the distribution comparison of the verification set mean square error (red column, MSE) and the exponential weight (blue column, Exponential Weight) of 10 base learners. It can be clearly seen that the smaller the MSE, the higher the exponential weight of the base learner (for example, the MSE of the base learner 4 has

the lowest and the highest weight; the MSE of the base learner 10 has a higher weight and a relatively low weight). The strong negative correlation of "performance-weight" reflects the precise distinction between the performance differences of exponential weighting and base learners - high-quality base learners get a higher weight in integration, and the weight of inferior base learners is effectively compressed. In terms of quantitative indicators, the weight of the index weight is extremely different (the ratio of the maximum weight to the minimum weight) can reach 1.1:1, and the weighted entropy value of the exponential weight (an indicator to measure the uniformity of distribution) is significantly lower, indicating that the weight is concentrated in a few high-performance base learners, and the distinction is better.

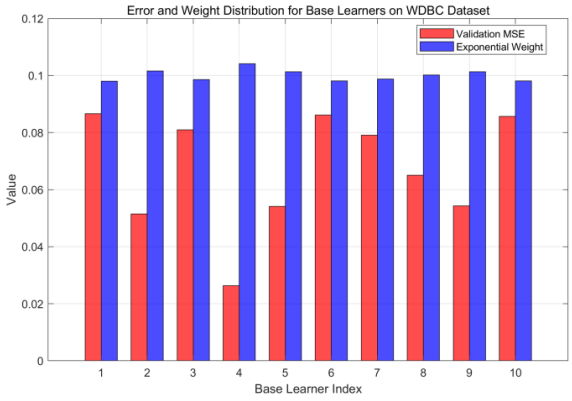


Figure 2: Error and weight distribution for base learners on WDBC dataset

3.3. The exp-bagging algorithm (validation-based)

Algorithm 1 illustrates the main steps of our approach:

Algorithm 1: Proposed Dynamic Weighted Ensemble Algorithm

Input: Training set D_{train}, validation set D_{val}, number of trees T
1: Initialize an empty list of trees and weights: Trees ← [ ], Weights ← [ ]
2: for t = 1 to T do
3: D_t ←Bootstrap sample from D_{train}
4: h_t ← Train a decision tree on D_t
5: MSE_t ← Calculate MSE of h_t on D_{val}
6: ω_t← exp(-MSE:)// Calculate raw weight
7: Append h_t to Trees and ω_t to Weights
8: end for
9: Normalize weights: Weights←Weights/sum(Weights)
10: return Ensemble model $H(x) = \sum_{t=1}^{T} \text{Weights}[t] \,.\, \text{Trees}[t](x)$
Output: Trained ensemble model H

3.4 Theoretical advantages and analysis of the exponential weighting scheme

The index weighted scheme proposed in this study is concise and has sufficient theoretical basis and statistical significance. Its main advantages are reflected in the following four aspects:

1. Intrinsic normalisation mechanism

The weighting scheme employs the following form:

$$\omega_i = \frac{exp(\text{-}MSE_i)}{\sum_j exp(\text{-}MSE_j)}. \quad (10)$$

The design can achieve the automatic standardization of weights without the introduction of additional scaling parameters, which not only simplifies the implementation process, but also avoids the subjective deviation that may be caused by the artificial setting of the normalization factor.

2. Cross-scale adaptability

Considering that there may be scale differences in the mean square error of different data sets, the direct use of the original value is easy to lead to an imbalance in weight distribution. The core advantage of the exponential function exp(−x) is the sensitive capture ability of relative differences. As long as the error of each base learner is calculated on the same verification set, their relative sorting can reliably reflect the performance, and the exponential transformation can adaptively amplify this difference to ensure the stability and robustness of the weight distribution.

3. Strong distinguishing ability

Compared with the linear weighting method, the exponential function has the characteristics of nonlinear amplification. When the performance of the base learner is similar, the weight distribution is relatively gentle; and when the performance difference is significant, the high-quality learner will receive an exponentially enhanced weight reward, and those with poor performance will be correspondingly suppressed. This characteristic naturally forms a screening mechanism of "winning and losing", which significantly improves the recognition and utilisation efficiency of high-performance learners by the integrated system.

4. Bayesian Probability Explanation

Based on the assumption that the error obeys the Gausian distribution, the exponential weight term can be interpreted as a non-normalised form of the post-obterential probability of the model. This characteristic forms a natural fit with the core framework of the PAC-

Bayes theory, providing solid theoretical support for the generalisation performance of the exponential weighted strategy. In the study, Germain et al. clearly proposed the PAC-Bayes boundary directly related to this characteristic, and its exact form is:

$$Pr_{S\sim D^m}\left(\forall Q on \mathrm{H}: R(G_Q) \leq \frac{1}{1-e^{-C}}\left[C\cdot R_S(G_Q)+\frac{1}{m}\left[\mathrm{KL}(Q\|P)+\ln\frac{1}{\delta}\right]\right]\right)\geq 1-\delta, \quad (11)$$

Among them, $R(G_Q)$ is the generalised error of the Gibbs classifier (the core measure of the generalised error of the integrated model), $R_S(G_Q)$ is the empirical error on the training set, $KL(Q\|P)\stackrel{def}{=}E_{\mathrm{h}\sim Q}\ln\frac{Q(\mathrm{h})}{P(\mathrm{h})}$ is the Kullback−Leibler divergence between Q and P, C is the adjustment parameter, and δ is the confidence level (usually 0.05) [29]. This theorem clearly defines the core logic of the PAC-Bayes theory: the generalised error of the integrated model is determined by the "weighted empirical error" and the "weight distribution complexity", and the high-quality weighting strategy needs to be balanced between the two.

Exp-Bagging's exponential weighting strategy ($\omega_i \propto \exp(-MSE_i)$) can theoretically improve the marginal efficiency (that is, the reduction of generalisation errors by high-quality base learners): from the perspective of Germain et al.' PAC-Bayes boundary, the core of marginal benefits is "weighted experience error" The ratio of the "reduction rate of difference" and "the growth rate of weight complexity". Exponential function $\exp(-MSE_i)$has non-linear amplification characteristics - when the difference in the base learner MSE is significant, the exponential weighting will greatly increase the weight proportion of the high-quality base learner and directly reduce the "weighted empirical error item" in the boundary; at the same time, the weight only depends on the adaptive adjustment of the verification set MSE, and there is no amount The introduction of external super parameters ensures that the weight distribution and the uniform a priori KL dispersion is always at a low level, far lower than the dispersion value of the "over-concentrated weight" strategy, so that the growth rate of the "weight complexity item" is much slower than the reduction rate of the weighted empirical error. This characteristic of "fast error reduction and slow complexity growth" allows the index weighting to achieve "marginal benefit increment" in the PAC-Bayes boundary - with the addition of high-quality base learning devices, the tightness of the generalised error boundary continues to improve, and the improvement expands with the quality of the base learner. This is the theoretical essence of its superior to the traditional weighting strategy. Further combined with the idea of Bayesian model average (BMA), Germain et al.'s PAC-Bayes framework allows data-driven update of a prior distribution, and the calculation process of exponential weight is essentially based on the a posterior inference of the performance of the verification set to the confidence of the base learner - MSE The smaller the base learner, the higher the prediction credibility and the greater the corresponding a posterior probability ($\exp(-MSE_i)$), which is fully consistent with the core criterion of the PAC-Bayes theory that "low error models should be given higher integration confidence". This consistency enables the index weighting strategy to not only improve the integration performance in practical applications, but also prove the controllability of generalisation capabilities through the PAC-Bayes boundary of Germain and others, and realise the unity of "practical effectiveness" and "theoretical rigour".

4 Experimental validation

In order to systematically evaluate the comprehensive performance of the Exp-Bagging algorithm, this research has designed a rigourous experimental scheme, covering multiple public data sets, a variety of baseline comparison methods and a complete evaluation process. Preliminary experimental results show that this method significantly improves the computing efficiency while ensuring excellent classification performance. Taking the Climate-Simulation-Crashes data set as an example, this method achieves a classification accuracy of 95.7%, and the throughput exceeds 80,000 samples/second, showing the good potential for processing high-dimensional data. The complete performance comparison results of each data set are detailed in Table 8.

4.1 Implementation settings

4.1.1 Hardware and software environment

The hardware configuration included an AMD Ryzen 9 7945HX with Radeon Graphics CPU (16 cores/32 threads), 16GB DDR5 memory (15.7GB available), and a 1TB NVMe SSD solid-state drive. The experiments were conducted in the MATLAB R2023b environment on Windows 11 operating system. All custom algorithms, including Exp-Bagging, were independently coded and implemented by the author to ensure the accuracy of the experimental process and the reproducibility of the results.

4.1.2 Model training and evaluation

In the experimental initialisation stage, the global random number generator state is fixed by rng(42). This seed is a common reference value for reprodibility verification in the field of machine learning, which can balance randomness and result stability. The specific random process control is as follows:

1. Data division link: based on the randperm(n) function of fixed seeds, realise the random segmentation of the training set / verification set / test set to ensure that the data division results of each run are completely consistent;
2. Base learner training link: In the parallel training loop (parfor), all random operations (including Bootstrap sampling to generate training subset and the characteristic selection of decision tree node splitting) are controlled by the global fixed seed rng (42) and calculated in parallel through MATLAB. The random state isolation mechanism avoids random duplication between different base learners, which not only ensures the diversity of integrated models, but also ensures that the generation process of each trained base learner can be reproduced.

Each data set is randomly divided into training set,

verification set and test set according to the ratio of 70:15:15. In order to reduce the impact of random factors in the process of data division and training, the whole experimental process (including data division, model training and performance evaluation) is independently repeated 10 times, and the average and standard deviation of the final performance index is taken.

The main parameter settings are as follows:

Base learner: Use the fitrtree function to build a decision tree, set the minimum number of samples of leaf nodes to 5, and enable the MSE-based automatic pruning function to balance the complexity of the model and the generalisation ability.

Integration scale: All integration methods (Exp-Bagging, Bagging and Random Forest) include 10 base learners. All experiments uniformly set the number of base learners T=10. This selection comes from the results of the melting experiment of the number of 4.4.2 base learners: when T increases from 5 to 10, the core indicators such as the F1 score and accuracy of the three data sets are significantly improved, and the throughput is maintained at the high efficiency level; while T continues to increase to 20 and 50, the performance improvement tends to stagnate, but the throughput plummets, and the efficiency loss is significant. At the same time, when T=10, the performance standard deviation of the model on the three data sets is the smallest, and the stability is better than that of T equal to other values. In summary, T=10 is the optimal equilibrium point of performance, efficiency and stability, so it is used as a unified parameter. It is worth mentioning that this paper uses a systematic hyperparameter tuning method based on the accuracy of the verification set to optimise the performance of the AdaBoost model. This method effectively evaluates the generalisation ability of the model and prevents overfitting by grid searching the number of weak learners (T=5,10,20,50) and the learning rate (0.1,0.5,1.0,1.5), taking the verification set accuracy rate as the only criterion for parameter selection. By combining the dynamic early stop mechanism in the tuning process and real-time monitoring of the verification set performance during the training process to determine the best number of iterations, an optimised parameter configuration scheme is provided for the practical application of the AdaBoost algorithm.

Feature sampling: The number of feature sampling of each tree in the random forest is set to the square root of the total number of features (√P, P is the feature dimension).

Sequence integration: The number of iterations of AdaBoost is set to 10 rounds.

The core task of this study is a classification task. All evaluation metrics are designed for category prediction scenarios and do not involve regression-related metrics (e.g., MAE, MSE), so as to ensure consistency between the metrics and the task objectives. The specific definitions of the metrics are as follows:

1. Accuracy: The proportion of correctly classified samples among all samples, reflecting the overall prediction capability of the model;
2. Precision: The proportion of truly positive samples among the samples predicted as positive, measuring the model’s ability to control "false positive misjudgments";
3. Recall: The proportion of truly positive samples correctly predicted among all actual positive samples, measuring the model’s ability to avoid "false negative misjudgments";
4. F1-score: The harmonic mean of Precision and Recall, used to balance the two types of errors and adapt to the evaluation needs of imbalanced data scenarios.
5. Throughput: Based on the experimental hardware (AMD Ryzen 9 7945HX processor, 16-core fully enabled; 16GB DDR5 memory, available 15.7GB; 1TB NVMe SSD solid-state drive), it is defined as "in the hardware environment, in the single reasoning process, single The number of test samples that can be processed in bit time (per second), the unit is "sample / second". The calculation method is: input all samples of the test set into the trained model. Reasoning, record the total reasoning time (including the whole process of data reading, model calculation, and result output, excluding the time consumption of data preprocessing), and finally calculated through the "total number of samples of the test set / total reasoning time". This calculation method can ensure that the throughput indicator truly reflects the real-time reasoning ability of the model in the actual hardware environment, and all comparison methods (decision tree, random forest, AdaBoost) are tested under the same hardware settings to ensure the fairness of efficiency comparison.

4.2 Datasets

The experiment selected three public classification data sets for performance verification: Climate-Simulation-Crashes, WDBC and Diabetes. These data sets come from different fields such as climate simulation and medical diagnosis. They have significant differences in sample size, characteristic dimension and data characteristics, and can comprehensively evaluate the applicability of the method. In order to eliminate the impact of characteristic metric differences, all data sets are preprocessed by the Z-score standardised method. Table 1 details the statistical characteristics of each data set and the specific process of preprocessing.

Table 2: Statistical summary of the benchmark datasets

Dataset Name	Sample Size	Number of Features	Source	Task Type
Climate Model Simulation Crashes	540	18	Kaggle	Classification
Wdbc	569	30	Kaggle	Classification
Diabetes	768	8	Kaggle	Classification

4.3 Baseline methods

In order to verify the effectiveness of the Exp-Bagging algorithm, the following three types of representative baseline methods are selected for comparative analysis:

1. Decision Tree: As the basic unit of the integrated model, this comparison can verify the effectiveness of the integrated strategy itself;
2. Random Forest: As the most influential extension method in the Bagging framework, it is a key reference for evaluating the improvement value of this study;
3. AdaBoost: As a typical representative of sequential integrated learning, this comparison helps to comprehensively evaluate the competitiveness of this research method in a wider field.

The complete performance comparison results of the above baseline methods on the three data sets are shown in Tables 3, 4 and 5.

Table 3: Performance of different methods on the climate model simulation crashes dataset

Algorithm Name	Accuracy	Precision	Recall	F1	Throughput
Decision tree	0.926±0.025	0.956±0.023	0.964±0.024	0.960±0.014	879,806.92
Random forest	0.931±0.023	0.947±0.020	0.980±0.019	0.963±0.013	23843.00
AdaBoost	0.937±0.019	0.939±0.018	0.996±0.006	0.967±0.010	30652.56
Exp-Bagging	0.957±0.018	0.969±0.013	0.986±0.010	0.977±0.009	81518.68

Table 4: Performance of different methods on the WDBC dataset

Algorithm Name	Accuracy	Precision	Recall	F1	Throughput
Decision tree	0.926±0.020	0.889±0.051	0.901±0.051	0.894±0.037	1092377.11
Random forest	0.928±0.012	0.893±0.032	0.921±0.039	0.906±0.019	25444.33
AdaBoost	0.951±0.020	0.943±0.045	0.926±0.052	0.933±0.030	45391.92
Exp-Bagging	0.981±0.006	0.944±0.017	1.000±0.000	0.971±0.009	70112.25

Table 5: Performance of different methods on the Diabetes dataset.

Algorithm Name	Accuracy	Precision	Recall	F1	Throughput
Decision tree	0.690±0.026	0.563±0.075	0.507±0.061	0.529±0.044	965,284.71
Random forest	0.759±0.051	0.644±0.087	0.621±0.085	0.630±0.077	25428.13
AdaBoost	0.745±0.025	0.666±0.040	0.553±0.078	0.601±0.049	58015.51
Exp-Bagging	0.787±0.019	0.576±0.037	0.690±0.039	0.623±0.026	78172.10

(Table 3-5: Binary Classification Performance. We report macro F1-scores (mean + stdev over 10 trials) on test data with optimized hyperparameters. The rank of each method is presented in brackets. The datasets are sorted by the number of features)

It should be noted that there are significant differences in the performance improvement of Exp-Bagging on different datasets: on the Climate Model Simulation Crashes and WDBC datasets, the F1 score is 1.4% and 6.5% higher than that of Random Forest, respectively, with obvious advantages; but on the Diabetes dataset, the F1 score only decreases slightly compared with Random Forest, while the accuracy rate is 3.8% higher, achieving only a small optimization. This difference is not the failure of the method, but is determined by the inherent characteristics of the Diabetes dataset: judging from the data characteristics, the Diabetes dataset has the problems of category imbalance (only 34.9% of the diseased samples) and low feature differentiation—the distribution of 8 continuous features (such as blood sugar and blood pressure) in healthy and diseased samples overlaps highly, resulting in the concentration of the prediction error (MSE) distribution of the base learner. The exponential weighting strategy of Exp-Bagging needs to rely on the performance difference of the base learner to achieve "selecting the superior and eliminating the inferior". When the difference is small, the weighted optimization space is compressed, which weakens the gain effect of weighted integration. In addition, the insufficient density of sample feature information limits the diversity of base learners. Although the sample size of the Diabetes dataset (768) is higher than that of WDBC (569) and Climate Model Simulation Crashes (540), its feature dimension is only 8, far lower than the 30 of WDBC and 18 of Climate Model

Simulation Crashes, leading to low "sample-feature" information density. When generating training subsets through Bootstrap sampling, the limited feature dimension makes it difficult for the base learner to learn differentiated laws, and there are very few optional differentiated features when the decision tree node splits, which ultimately leads to a high degree of similarity in the output laws of the base learner. The performance improvement of integrated learning depends on the "balance between the accuracy and diversity of the base learner" (refer to Breiman's 1996 Bagging classic theory). When the diversity is insufficient, even if the weighting strategy is adopted, it is difficult to break through the performance ceiling of a single base learner, which is exactly the key reason why Exp-Bagging is limited in this dataset. Despite the above limitations, Exp-Bagging still shows irreplaceable practical value on the Diabetes dataset: on the one hand, its throughput (78172.10 samples/second) is 3.1 times that of Random Forest and 1.3 times that of AdaBoost, which can meet the clinical needs of "real-time detection and rapid feedback" in diabetes-assisted diagnosis; on the other hand, the standard deviation of its performance indicators (accuracy rate ± 0.019, F1 score ± 0.026) is significantly lower than that of Random Forest (accuracy rate ± 0.051, F1 score ± 0.077), reflecting better stability—in medical diagnosis scenarios, model stability is as important as accuracy, which can effectively reduce the risk of "misdiagnosis caused by model fluctuations".

4.4 Ablation Study

In order to evaluate the actual contribution of each component in the integrated framework, this study designed a systematic ablation experimental scheme:

4.4.1 Validity verification of weighted strategy

This section focusses on the impact of different weighting strategies on integration performance. The experimental setting is divided into two levels: the core comparison level to all data sets, comparing the performance of the exponential weighting strategy proposed in this paper with the optimal single model; the extended comparison layer introduces more advanced weighting methods on some data sets for in-depth comparison: new average integration on the Diabetes data set, Three strategies based on sorting weighting and MSE countdown weighting; supplement the two methods of sorting-based weighting and MSE countdown weighting on the climate-simulation-crashes data set.

In order to clarify the calculation logic of the two types of comparison integration strategies and ensure the fairness and reproducibility of experimental comparison, this section supplements the mathematical formulas, core principles and implementation details of Equal Weighting and Inverse-MSE Weighting, and contrasts with the exponential weighting of Exp-Bagging in this article.

(1) Equal Weighting is the most classic static weighting strategy. The core logic is that "all base learning devices contribute equally". It does not rely on any performance indicators and directly assigns the same weight to each base learning device, which is suitable for scenarios with small performance differences in base learning devices. The weight of the i-base learning device is ω_i^{equal}, then:

$$\omega_i^{equal}=\frac{1}{T}. \tag{12}$$

Among them, T is the total number of base learners (T=10 in this experiment).

The sum of the weights of all base learners is $\sum_{i=1}^{T}\omega_i^{equal}=1$, and no additional normalisation steps are required.

For the new test sample x, the final prediction of the average integration is the arithmetic average of the predicted values of all base learners, and the calculation formula is:

$$H_{equal}(x)=\sum_{i=1}^{T}\omega_i^{equal}\cdot \mathrm{h}_i(x)=\frac{1}{T}\sum_{i=1}^{T}\mathrm{h}_i(x). \tag{13}$$

Among them, $\mathrm{h}_i(x)$ is the predicted value of the ith base learner for sample x. Finally, the final category label is obtained by discreting the continuous output $H_{equal}(x)$ by setting the threshold (0.5 in this experiment).

(2) Inverse-MSE Weighting is a dynamic weighting strategy based on the performance of the base learner. The core logic is "the smaller the MSE, the greater the weight". The linear correlation between performance and weight is realised through the inverse mapping of MSE, and the performance of the base learner needs to be evaluated by relying on the verification set. Introduce the least valueϵ (take $\epsilon=10^{-10}$ in this experiment) to avoid the error of "divising by 0" when MSE=0. The original weight $\omega_i^{\text{inv-mse,raw}}$ of the i-th base learner is:

$$\omega_i^{\text{inv-mse,raw}}=\frac{1}{MSE_i+\epsilon}, \tag{14}$$

Among them, the calculation of MSE_i is the same as the formula (6).

The original weight is normalised into a convex combination with a sum of 1 to ensure the interpretability of probability. After normalisation, the weight $\omega_i^{inv\text{-}mse}$ is:

$$\omega_i^{inv\text{-}mse}=\frac{\omega_i^{inv\text{-}mse,raw}}{\sum_{k=1}^{T}\omega_k^{inv\text{-}mse,raw}} \tag{15}$$

For the new test sample x, the final prediction of the Inverse-MSE integration is the weighted average of the predicted value of the base learner, and the calculation formula is:

$$H_{inv\text{-}mse}(x)=\sum_{i=1}^{T}\omega_i^{inv\text{-}mse}\cdot \mathrm{h}_i(x) \tag{16}$$

Similarly, the final category label is obtained by setting the threshold (0.5) discrete continuous output.

Ablation experiment results (Tables 6 – 8) reveal the discriminative ability of the proposed exponential weighting strategy in optimizing ensemble performance. On the Climate Model Simulation Crashes dataset, exponential weighting outperforms rank-based and inverse-MSE weighting mainly in F1-score (97.7% vs. 97.3% and 97.6%), which indicates that the negative exponential function effectively amplifies the contribution of low-error subtrees while suppressing high-error ones. For the WDBC dataset, exponential weighting achieves a 4.5-percentage-point accuracy improvement over the best single model (98.1% vs. 93.6%), confirming that dynamic

weighting compensates for the limitation of individual decision trees in handling high-dimensional medical data. On the Diabetes dataset with obvious data imbalance, exponential weighting maintains comparable accuracy to inverse-MSE weighting (78.7% vs. 78.6%) but exhibits smaller indicator variance, reflecting stronger robustness to noisy data.

Based on the comprehensive experimental results shown in Table 6 to Table 8, the following conclusions can be drawn: the exponential weighting strategy proposed in this paper performs well in all test scenarios, and its performance is stable and significantly better than the average integration and optimal single model baseline. Furthermore, in direct comparisons with other advanced weighting strategies (on Diabetes and climate-simulation-crashes), the exponential weighting strategy also exhibits competitive advantages, achieving optimal performance.

Figure 3-5 is the sensitivity analysis chart of the number of base learning devices (T) of Exp-Bagging in Climate Model Simulation Crashes dataset, WDBC dataset and Diabetes dataset (classification performance chooses Accuracy and Precision).

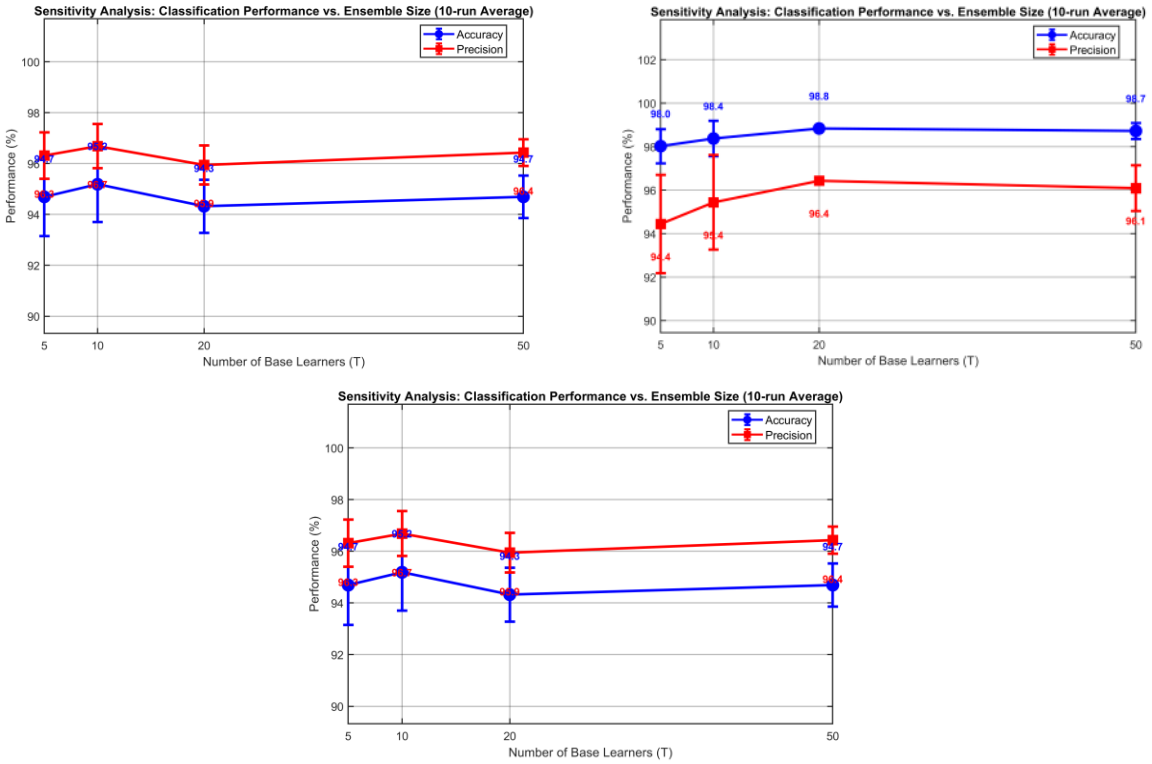


Figure 3: Sensitivity analysis chart on Climate Model Simulation Crashes dataset

Figure 4: Sensitivity analysis chart on WDBC dataset

Figure 5: Sensitivity analysis chart on Diabetes dataset

Table 6: Performance of different weighting schemes on the climate model simulation crashes dataset.

Weight Computation Methods	Accuracy	Precision	Recall	F1
Rank-based Weighting	0.948±0.011	0.968±0.012	0.979±0.007	0.973±0.006
Inverse-MSE Weighting	0.954±0.018	0.967±0.012	0.986±0.010	0.976±0.009
Best Single Model	0.928±0.019	0.967±0.009	0.955±0.021	0.961±0.011
Exponential Weighting	0.957±0.018	0.969±0.013	0.986±0.010	0.977±0.009

Table 7: Performance of different weighting schemes on the WDBC dataset

Weight Computation Methods	Accuracy	Precision	Recall	F1
Best Single Model	0.936±0.021	0.864±0.046	0.948±0.050	0.903±0.031
Exponential Weighting	0.981±0.006	0.944±0.017	0.997±0.006	0.971±0.009

Table 8: Performance of different weighting schemes on the diabetes dataset.

Weight Computation Methods	Accuracy	Precision	Recall	F1
Equal Weighting	0.784±0.023	0.572±0.043	0.683±0.042	0.622±0.030
Rank-based Weighting	0.779±0.022	0.563±0.040	0.660±0.054	0.607±0.040
Inverse-MSE Weighting	0.786±0.020	0.574±0.039	0.690±0.039	0.626±0.026
Best Single Model	0.735±0.035	0.493±0.052	0.613±0.074	0.545±0.055
Exponential Weighting	0.787±0.019	0.576±0.037	0.690±0.039	0.623±0.026

4.4.2 Parameter sensitivity analysis of the number of base learners (T)

In order to evaluate the impact of changes in the number of base learners (T) on the performance and reasoning speed of the Exp-Bagging model, supplement the ablation experiment: select four typical values of T=5, 10, 20 and 50, repeat the experiment 10 times on the three data sets of Climate-Simulation-Crashes, WDBC and Diabetes, record the classification performance index (accuracy, precision, F1 score) and the reasoning speed index (throughput), and analyse the trade-off relationship caused by parameter changes.

This experimental setting maintains that other parameters are consistent with the original experiment, and the verification set accounts for 15%, the minimum number of samples of leaf nodes is 5, and MSE automatic pruning is enabled. In addition, the throughput calculation method remains unchanged, that is, based on AMD Ryzen 9 7945HX CPU, the speed of all samples of the single-wheel reasoning processing test set (unit: samples/s).

The model performance and throughput under the number of different base learners are shown in Table 9-11:

Table 9: Performance comparison of different T values on the climate model simulation crashes dataset

Number of base learners (T)	Accuracy	Precision	Recall	F1	Throughput
5	0.947±0.009	0.961±0.006	0.983±0.011	0.972±0.006	125701.13
10	0.955±0.008	0.969±0.010	0.983±0.006	0.976±0.004	69106.90
20	0.954±0.012	0.969±0.006	0.983±0.010	0.976±0.006	31131.94
50	0.954±0.010	0.969±0.006	0.983±0.006	0.976±0.005	14174.74

Table 10:Performance comparison of different T values on the WDBC dataset

Number of base learners (T)	Accuracy	Precision	Recall	F1	Throughput
5	0.977±0.010	0.935±0.026	0.996±0.011	0.964±0.016	125968.69
10	0.979±0.007	0.941±0.015	0.996±0.011	0.968±0.011	69884.78
20	0.986±0.004	0.957±0.013	1.000±0.000	0.978±0.007	39750.01
50	0.987±0.003	0.961±0.010	1.000±0.000	0.980±0.005	15408.74

Table 11:Performance comparison of different T values on the Diabetes dataset

Number of base learners (T)	Accuracy	Precision	Recall	F1	Throughput
5	0.772±0.014	0.546±0.019	0.680±0.067	0.605±0.037	64476.89
10	0.791±0.017	0.577±0.029	0.720±0.031	0.640±0.027	74728.29
20	0.787±0.015	0.569±0.026	0.743±0.040	0.644±0.022	14371.10
50	0.790±0.016	0.574±0.026	0.737±0.028	0.644±0.021	13150.33

Judging from the experimental data, on the Climate Model Simulation Crashes data set, when the number of base learners increased from 5 to 10, the accuracy rate and F1 score were improved, and the throughput decreased significantly; when the number increased to 20 and 50, the classification The performance basically tends to be saturated, and the throughput continues to decrease. The WDBC data set shows a similar trend. The number of base learners has improved the classification performance from 5 to 10, and the performance of 20 and 50 has further improved slightly, but the throughput has decreased significantly. The Diabetes data set has relatively better classification performance indicators such as accuracy and F1 score when the number of base learners is 10. When the number is 5, 20 and 50, the performance is slightly worse, and the throughput also declines with the increase in the number of base learners. Overall, the increase in the number of base learners can improve the classification performance within a certain range, but it will lead to a decrease in the reasoning throughput, and there are differences in the trade-off between performance and throughput of different data sets. Among them, the classification performance of WDBC data sets is optimal when the number of base learners is 50, the classification performance and throughput of the Climate Model Simulation Crashes data set are better balanced when the number is 10, and the classification performance of the Diabetes data set is relatively better when the number is 10. The above conclusion further explains the necessity of selecting the number of base learning devices to be 10.

4.5 Comprehensive performance evaluation

Evaluation metrics widely used in regression tasks are employed to comprehensively assess model performance. All experiments report the following metrics, with accuracy, precision, recall, F1, Throughput as primary evaluation indicators. The corresponding performance metrics are summarized in Table 12.

Table12: Performance of the proposed method across different datasets

Dataset Name	Accuracy	Precision	Recall	F1	Throughput
Climate Model Simulation Crashes	0.957±0.018	0.969±0.013	0.986±0.010	0.977±0.009	81518.68
WDBC	0.981±0.006	0.944±0.017	1.000±0.000	0.971±0.009	70112.25
Diabetes	0.787±0.019	0.576±0.037	0.690±0.039	0.623±0.026	78172.10

Comprehensive performance (Table 12) shows the method's cross-scenario excellence: 95.7% – 98.1% accuracy (100% recall on WDBC) and 70,112 – 81,518 samples/second throughput, balancing classification quality and real-time inference across all three datasets.

In summary, Exp-Bagging has achieved a balance of classification performance, reasoning efficiency and stability on three types of data sets. In order to further verify the actual deployment reliability of the model, it is necessary to supplement the fitting risk analysis - the complexity of the integrated model is likely to increase with the increase in the number of base learners, and improper control may lead to a decline in generalisation ability. Combined with the previous experimental data set and settings, the quantitative analysis of the accuracy curve (Figure 6-8) through training-verification is as follows: It can be seen from the training-verification accuracy curve of the three data sets that Exp-Bagging maintains a training accuracy of more than 0.96 on the Climate Model Simulation Crashes data set, and the verification accuracy is stable above 0.90, and the gap is controlled. Within 0.06; WDBC data set training accuracy is above 0.98, verification accuracy is above 0.90, and the gap is within 0.08; Diabetes data set training accuracy is about 0.90, verification accuracy is above 0.60, the gap is within 0.30, and there are three types of data sets There is no overfitting characteristic of "training accuracy continues to rise, verification accuracy first to rise and then fall". This shows that Exp-Bagging effectively avoids the risk of overfitting on different characteristic data sets and has reliable generalisation capabilities through the dual design of exponentially weighted "hidden pruning" and verification set-driven weight distribution.

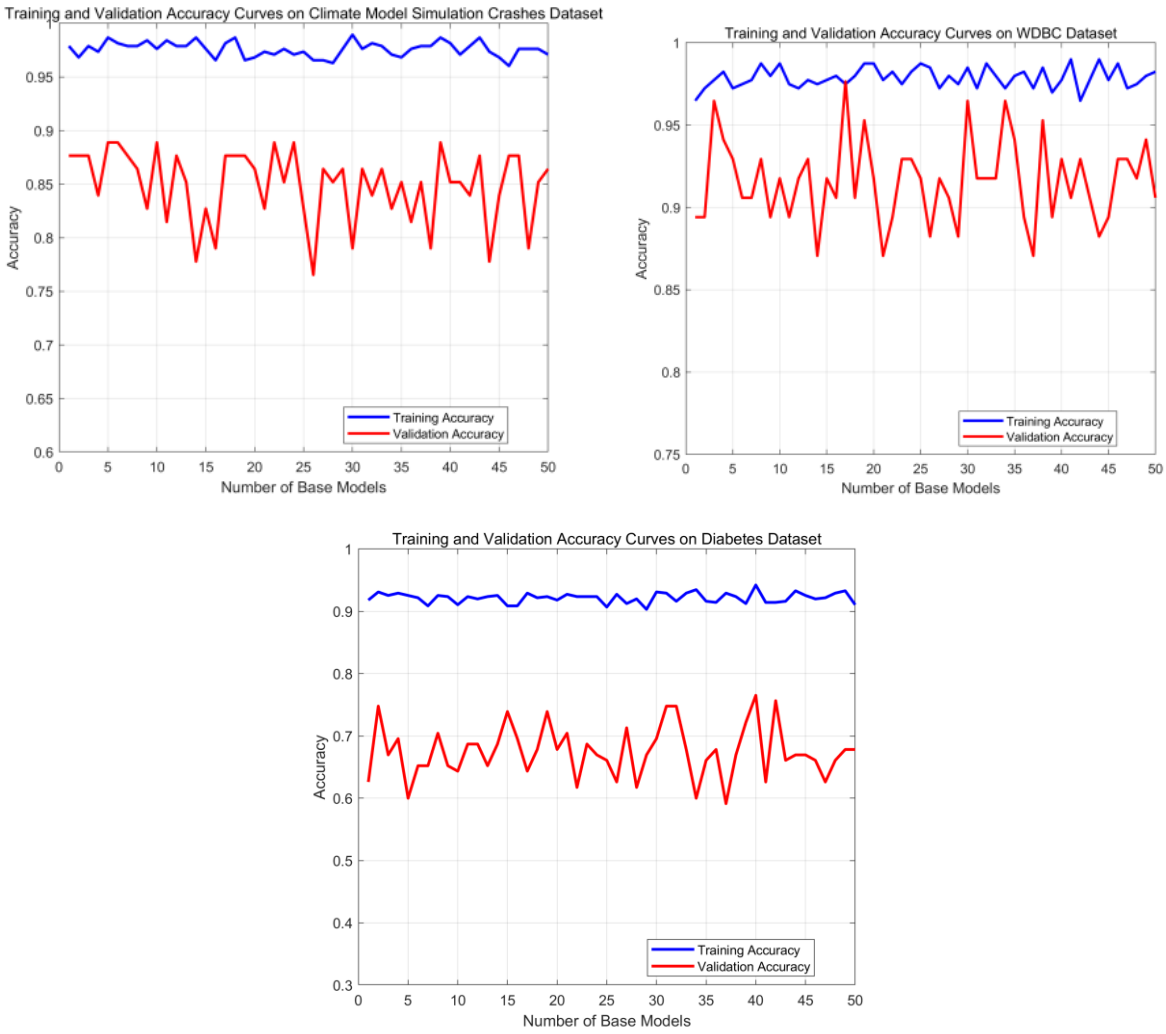


Figure 6: Training/test accuracy curve on Climate Model Simulation Crashes data set
Figure 7: Training/test accuracy curve on WDBC data set
Figure 8:Training/test accuracy curve on Diabetes data set

5 Conclusion

This paper proposes a general machine learning framework based on the idea of weighted integration, and systematically verifies it on three representative data sets of Climate Model Simulation Crashes, WDBC and Diabetes classification. The main contributions of this research can be summarised in the following three aspects:
(1) A unified parallel integrated learning framework suitable for classified tasks has been built, showing good task adaptability and scalability;
(2) A dynamic exponential weighted fusion strategy based on the performance of the verification set is designed, which can adaptively allocate the integration weight according to the actual performance of the base learning device;
(3) Through interdisciplinary experimental verification, the effectiveness and robustness of this method in a variety of practical scenarios have been confirmed.
The experimental results show that the weighted integration method proposed in this paper performs well in multiple classification tasks. The evaluation results of the climate simulation collapse, breast cancer and diabetes data set show that the method has achieved ideal results in key indicators such as accuracy, precision rate, recall rate and F1 score, verifying its practical value in diagnostic applications. It is especially worth noting that the framework shows excellent prediction efficiency while maintaining high classification accuracy, with a maximum throughput of 81,518.68 samples/second, indicating that it has the ability to support the real-time prediction system.
Although this research has achieved stage results in the design and verification of dynamic weighted integration strategies, there are still the following limitations, which need to be further improved in subsequent research: the experiment only selects three public data sets of Climate Model Simulation Crashes, WDBC and Diabetes. Although they cover climate simulation, medical diagnosis and other fields, the sample size (540~768) and

feature dimensions (8~30 dimensions) are both in the middle and low scale. It does not verify the performance of the method in the scenario of large-scale data (such as million-level samples and more than 100-dimensional features), and cannot fully determine the stability efficiency and generalisation ability of Exp-Bagging when the data volume surges. The current experiment is based on the design of static data sets, which does not simulate the common stream data scenarios in practical applications (such as real-time medical monitoring data, dynamic financial transaction data), nor does it target the adaptive adjustment ability.

In future research, in-depth exploration will be continued in the following directions. First, the combination strategy of complex base learners integrating deep learning models will be studied to further improve the model's expressive ability. Second, an online learning mechanism adaptable to dynamic data distribution will be developed to enhance the model's adaptability in streaming data scenarios. Third, the framework will be extended to more key fields such as financial risk assessment and industrial fault detection. Finally, model compression and acceleration technologies will be explored to improve deployment efficiency in resource-limited environments.

6 References

[1] Sirocchi, C., Bogliolo, A., & Montagna, S. (2024). Medical-informed machine learning: integrating prior knowledge into medical decision systems. BMC Medical Informatics and Decision Making, 24(Suppl 4), 186. https://doi.org/10.1186/s12911-024-02582-4

[2] Chou, J. S., & Chen, K. E. (2024). Optimizing investment portfolios with a sequential ensemble of decision tree-based models and the FBI algorithm for efficient financial analysis. Applied Soft Computing, 158, 111550. https://doi.org/10.1016/j.asoc.2024.111550

[3] Azam, Z., Islam, M. M., & Huda, M. N. (2023). Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree. Ieee Access, 11, 80348-80391. https://doi.org/10.1109/ACCESS.2023.3296444

[4] Wang, D. N., Li, L., & Zhao, D. (2022). Corporate finance risk prediction based on LightGBM. Information Sciences, 602, 259-268. https://doi.org/10.1016/j.ins.2022.04.058

[5] Opanasenko, V. M., Fazilov, S. K., Mirzaev, O. N., & Sa'dullo ugli Kakharov, S. (2024). An ensemble approach to face recognition in access control systems. Journal of Mobile Multimedia, 20(3), 749-768. https://doi.org/10.13052/jmm1550-4646.20310

[6] Dong, X., Yu, Z., Cao, W., Shi, Y., & Ma, Q. (2020). A survey on ensemble learning. Frontiers of Computer Science, 14(2), 241-258. https://doi.org/10.1007/s11704-019-8208-z

[7] Ye, M., Lan, X., Leng, Q., & Shen, J. (2020). Cross-modality person re-identification via modality-aware collaborative ensemble learning. IEEE Transactions on Image Processing, 29, 9387-9399. https://doi.org/10.1109/TIP.2020.2998275

[8] Pallathadka, H., Ramirez-Asis, E. H., Loli-Poma, T. P., Kaliyaperumal, K., Ventayen, R. J. M., & Naved, M. (2023). Applications of artificial intelligence in business management, e-commerce and finance. Materials Today: Proceedings, 80, 2610-2613. https://doi.org/10.1016/j.matpr.2021.06.419

[9] Zhou, H., Zhang, J., Zhou, Y., Guo, X., & Ma, Y. (2021). A feature selection algorithm of decision tree based on feature weight. Expert Systems with Applications, 164, 113842. https://doi.org/10.1016/j.eswa.2020.113842

[10] He, Z., Wu, Z., Xu, G., Liu, Y., & Zou, Q. (2021). Decision tree for sequences. IEEE transactions on Knowledge and Data Engineering, 35(1), 251-263. https://doi.org/10.1109/TKDE.2021.3075023

[11] Pekel, E. (2020). Estimation of soil moisture using decision tree regression. Theoretical and Applied Climatology, 139(3), 1111-1119. https://doi.org/10.1007/s00704-019-03048-8

[12] Sagi, O., & Rokach, L. (2021). Approximating XGBoost with an interpretable decision tree. Information sciences, 572, 522-542. https://doi.org/10.1016/j.ins.2021.05.055

[13] Chanmee, S., & Kesorn, K. (2023). Semantic decision Trees: A new learning system for the ID3-Based algorithm using a knowledge base. Advanced engineering informatics, 58, 102156. https://doi.org/10.1016/j.aei.2023.102156

[14] He, S., Wu, J., Wang, D., & He, X. (2022). Predictive modeling of groundwater nitrate pollution and evaluating its main impact factors using random forest. Chemosphere, 290, 133388. https://doi.org/10.1016/j.chemosphere.2021.133388

[15] Sun, Z., Wang, G., Li, P., Wang, H., Zhang, M., & Liang, X. (2024). An improved random forest based on the classification accuracy and correlation measurement of decision trees. Expert Systems with Applications, 237, 121549. https://doi.org/10.1016/j.eswa.2023.121549

[16] Dinh, T. P., Pham-Quoc, C., Thinh, T. N., Do Nguyen, B. K., & Kha, P. C. (2023). A flexible and efficient FPGA-based random forest architecture for IoT applications. Internet of Things, 22, 100813. https://doi.org/10.1016/j.iot.2023.100813

[17] Sevinç, E. (2022). An empowered AdaBoost algorithm implementation: A COVID-19 dataset study. Computers & Industrial Engineering, 165, 107912. https://doi.org/10.1016/j.cie.2021.107912

[18] Huang, X., Li, Z., Jin, Y., & Zhang, W. (2022). Fair-AdaBoost: Extending AdaBoost method to achieve fair classification. Expert Systems with Applications, 202, 117240. https://doi.org/10.1016/j.eswa.2022.117240

[19] Germain, P., Lacasse, A., Marchand, M., Shanian, S., & Laviolette, F. (2009). From PAC-Bayes bounds to KL regularization. Advances in neural information processing systems, 22.

[20] Ngo, G., Beard, R., & Chandra, R. (2022). Evolutionary bagging for ensemble learning. Neurocomputing, 510, 1-14. https://doi.org/10.1016/j.neucom.2022.08.055

[21] Al-Mudhafar, W. J., Abbas, M. A., & Wood, D. A. (2022). Performance evaluation of boosting machine learning algorithms for lithofacies classification in heterogeneous carbonate reservoirs. Marine and Petroleum Geology, 145, 105886. https://doi.org/10.1016/j.marpetgeo.2022.105886

[22] Dong, J., Chen, Y., Yao, B., Zhang, X., & Zeng, N. (2022). A neural network boosting regression model based on XGBoost. Applied Soft Computing, 125, 109067. https://doi.org/10.1016/j.asoc.2022.109067

[23] Hajjouz, A., & Avksentieva, E. Y. (2025). Enhancing and extending CatBoost for accurate detection and classification of DoS and DDoS attack subtypes in network traffic. Научно-технический вестник информационных технологий, механики и оптики, 25(1), 114-127. https://doi.org/10.17586/2226-1494-2025-25-1-114-127

[24] Iranzad, R., & Liu, X. (2024). A review of random forest-based feature selection methods for data science education and applications. International Journal of Data Science and Analytics, 1-15.

[25] Zhang, B., Li, Y., & Chai, Z. (2022). A novel random multi-subspace based ReliefF for feature selection. Knowledge-Based Systems, 252, 109400. https://doi.org/10.1016/j.knosys.2022.109400

[26] Xing, H. J., Liu, W. T., & Wang, X. Z. (2024). Bounded exponential loss function based AdaBoost ensemble of OCSVMs. Pattern Recognition, 148, 110191. https://doi.org/10.1016/j.patcog.2023.110191

[27] Ileberi, E., Sun, Y., & Wang, Z. (2021). Performance evaluation of machine learning methods for credit card fraud detection using SMOTE and AdaBoost. IEEE access, 9, 165286-165294. https://doi.org/10.1109/ACCESS.2021.3134330

[28] Kalagotla, S. K., Gangashetty, S. V., & Giridhar, K. (2021). A novel stacking technique for prediction of diabetes. Computers in Biology and Medicine, 135, 104554. https://doi.org/10.1016/j.compbiomed.2021.104554

[29] Chugg, B., Wang, H., & Ramdas, A. (2023). A unified recipe for deriving (time-uniform) PAC-Bayes bounds. Journal of Machine Learning Research, 24(372), 1-61.

DCAGAT: A Graph Attention-Based Model with Reconstruction Regularization for Dollar-Cost Averaging Investment Prediction

Zhongli Lv
School of Economics and Management, Guizhou Normal University, Guiyang,550001, China
E-mail: zhongliLv@outlook.com

Keywords: Graph Attention Network (GAT), Dollar-Cost Averaging (DCA), Financial Market Prediction, Multi-Feature Fusion.

Received: April 29, 2025

The financial market is characterized by high volatility and noisy data, making it a formidable challenge to forecast trends and design robust investment strategies. In this paper, we propose an innovative prediction model that integrates multi-feature fusion with graph attention mechanisms to address these challenges, specifically tailored for dollar-cost averaging (DCA) strategies. Our model, termed DCAGAT, leverages Graph Attention Networks (GATs) to dynamically assess the interdependencies among various financial assets. By incorporating multiple market features—historical price fluctuations and trading volumes—the model constructs a comprehensive representation of market dynamics. A key innovation is the inclusion of an autoencoder-inspired reconstruction verification mechanism, which mitigates overfitting by ensuring that the model focuses on persistent market trends rather than transient noise. We validate the effectiveness of DCAGAT using historical data from Yahoo Finance and ETF from 2012 to 2022. We benchmark DCAGAT against four neural baselines—DNN, Conv1D, GCN and ST-GCN—on three DCA-oriented metrics: one-, seven- and fourteen-day directional accuracy (Adir) and Top-5 hit-rate (Aselect). While DCAGAT matches the best baseline on Adir, it consistently improves Aselect across the evaluated scenarios, with every gain passing the paired statistical-significance test., underscoring its superior stock-selection capability. Overall, our research provides a robust framework for financial market forecasting by combining advanced graph-based learning techniques with feature-rich data integration, offering valuable insights for investors seeking to optimize multi-day investment decisions in unpredictable market environments.

Povzetek: Prispevek predstavi nov model za napovedovanje finančnih trgov, ki izboljša naložbene odločitve tudi v nestabilnih in šumnih razmerah.

1 Introduction

Finance is inherently a game of risk and reward, and since the inception of the financial industry, forecasting market trends, mitigating risks, and enhancing returns have remained paramount objectives for practitioners[1, 2]. The prevailing view posits that, to a certain extent, financial markets are predictable and that both risk and return can be systematically evaluated [3]. For example, armed conflict has been observed to drive up gold prices, technological breakthroughs can trigger increases in stock prices[4], and favorable weather conditions may lead to declines in agricultural commodity futures prices. Nonetheless, the process of evaluation is considerably more complex in practice; in actual trading environments, such clear-cut trends are often elusive, with the market itself frequently serving as the most reliable indicator of underlying economic forces[5, 6].

Dollar-cost averaging (DCA) represents a unique investment strategy characterized by its systematic and periodic allocation of fixed funds into selected financial assets, irrespective of market price fluctuations[7]. This method appeals to a diverse group of investors—including novice individuals seeking to minimize the emotional and financial risks associated with market timing, as well as more experienced investors who value a disciplined, multi-day approach. Among its advantages, DCA mitigates the adverse effects of market volatility and reduces the risk of mistimed investment decisions, although it may underperform during extended bull markets; nevertheless, its benefits often outweigh these limitations[8]. Fundamentally, the strategy relies on historical market performance as its primary informational input, thereby anchoring investment decisions in empirically observed trends rather than speculative forecasts[9].

Rapid advances in artificial intelligence (AI) have precipitated its widespread adoption across diverse domains such as entertainment[10], healthcare[11], manufacturing[12], robot[13], and transportation[14]. In each of these fields, AI's data-driven methodologies have demonstrated remarkable predictive capabilities, a strength that naturally extends to the realm of finance[15, 16]. This technological prowess has spurred financial practitioners to explore innovative approaches, including

the ambitious pursuit of constructing a "Laplace demon"—a theoretical, data-driven model capable of processing vast amounts of market information to make highly informed predictions[17, 18].

However, the application of AI in finance presents unique challenges. Unlike machine vision or natural language processing, financial data is inherently more complex[19, 20]. The critical patterns and signals are deeply embedded within a noisy environment, making them more elusive and difficult to extract[21, 22]. Despite these hurdles, leveraging AI in financial analytics holds significant promise. By integrating historical market performance as a primary informational input, sophisticated computational models can gradually unravel these complexities, enhancing risk assessment and market forecasting capabilities[23]. Ultimately, the transformation of raw data into actionable insights not only reinforces AI's pivotal role in financial innovation but also sets the stage for more informed and strategic investment decisions[24, 25].

Existing studies center on short-horizon tasks—one-day price forecasts or binary direction probabilities—whereas multi-step prediction and strategies tailored to Dollar-Cost Averaging (DCA) receive scant attention. As Table 1 shows, most models optimize single-step direction accuracy, leaving the portfolio-level, multi-period allocation challenge intrinsic to DCA largely unaddressed.

Table 1: Representative graph/autoencoder models for financial forecasting

Year	Study / Model	Core Architecture	Public Dataset & Market Span	Pred. Horizon	Eval. Metrics	Reported Best Outcome
2025	**LSTMAE-Network** [26]	LSTM + Autoencoder	S&P 500 indices, 2000 – 2022	Next-day	Topological network metrics; crisis detection recall	Detected 5/6 major crises; F1 ↑ 10 % vs. correlation graph
2024	**SAE-MLP** [27]	Supervised Autoencoder + MLP	S&P 500, EUR/USD, BTC/USD, 2010 – 2022	5 – 30 min	Sharpe, Info-Ratio	Information Ratio ↑ 32 % vs. MLP baseline
2024	**DGDNN** [28]	Decoupled Graph Diffusion GNN	2 893 stocks (NASDAQ/NYSE/SSE), 2012 – 2021	Next-day	Accuracy, MCC	Acc. ↑ 9.1 % / MCC ↑ 0.09 over 12 baselines
2023	**DGATS** [29]	Dynamic-Attributes GAT	U.S. equities, 2014 – 2020	Next-day	Accuracy	Acc. ↑ 5 % vs. dynamic-GCN
2021	**ST-Trader** [30]	VAE-GCN-LSTM	Minute-level NYSE stocks, 2018	30 min	Direction accuracy	Acc. ↑ 4 % vs. GCN-LSTM w/o VAE

According to Kraken's research depicted in Figure 1, 59% of cryptocurrency investors utilize DCA as their primary investment strategy for different reasons[31]. Considering the inherent complexity of the stock market and the potential interdependencies among individual stocks, this paper proposes a financial market prediction model that leverages a graph attention mechanism combined with multi-feature fusion to develop comprehensive DCA recommendations[32]. Moreover, whereas short-horizon signals may offer high-frequency cues, they also carry elevated noise and sudden-event risk, reducing confidence for DCA's medium-range allocations. Therefore, we integrate an autoencoder-inspired reconstruction verification mechanism to mitigate attention overfitting: this encourages the model to focus on stable, persistent patterns rather than transient spikes. In summary, our aims are:

1. To develop robust multi-step (e.g., 7–14 day) forecasts suited for DCA investors, balancing information sufficiency against noise in these horizons.
2. To mitigate overfitting in graph attention via an autoencoder-style reconstruction penalty, fostering durable feature representations.
3. To evaluate the proposed DCAGAT model on realistic portfolio sizes, demonstrating its practical value for DCA decision-making.

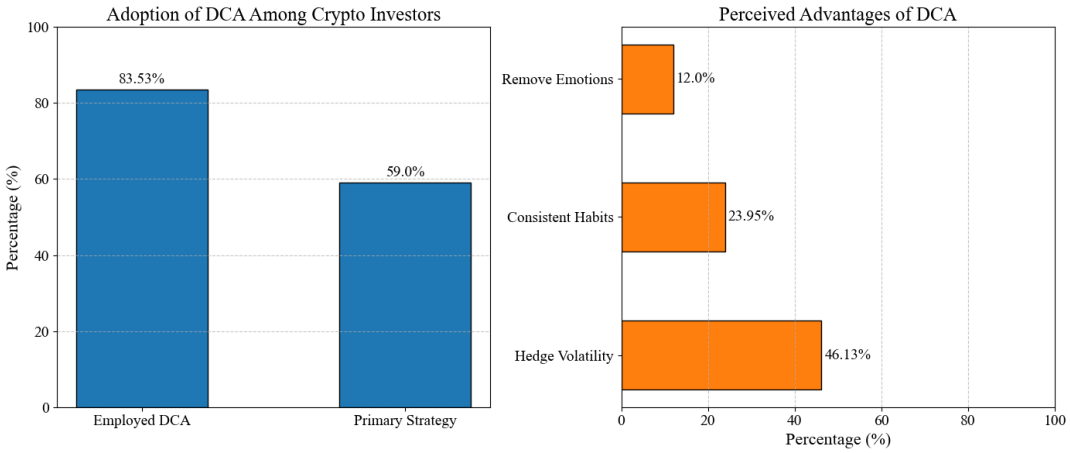


Figure 1: The visualization of Kraken's research on DCA

The remainder of this paper is organized as follows. Section II provides the necessary background on Graph Neural Networks and Graph Attention Networks. Section III introduces the proposed methodology in detail. Experimental setup and results are presented in Section IV. Section V offers a discussion of the findings, and Section VI concludes the paper.

2 Background

2.1 Graph neural networks

Graph Neural Networks (GNNs) have emerged as a pivotal tool for learning from graph-structured data by explicitly incorporating the inherent relational information into the learning process[33]. Consider a graph G defined as:

$$G=(V,E) \quad (1)$$

where V represents the set of nodes and E denotes the set of edges, encapsulating the relationships among nodes.

Each node $v_i \in V$ is associated with a feature vector $\mathbf{x}_i$ in $\mathbb{R}^d$:

$$\mathbf{x}_i \in \mathbb{R}^d, \quad v_i \in V \quad (2)$$

The fundamental concept underlying GNNs is the iterative refinement of node representations through a message-passing mechanism, which leverages the graph structure as prior knowledge. Specifically, at the *l*-th layer, the hidden representation $\mathbf{h}_i^{(l)}$ of node v_i is updated according to:

$$\mathbf{h}_i^{(l)} = \sigma\left(\sum_{j \in N(i)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l-1)} + \mathbf{b}^{(l)} \right) \quad (3)$$

In this equation, $N(i)$ denotes the neighborhood of node v_i, $\mathbf{W}^{(l)}$ is the learnable weight matrix at layer *l*, $\mathbf{b}^{(l)}$ is the bias vector, and σ is a non-linear activation function, such as ReLU. The initial hidden state is established by setting:

$$\mathbf{h}_i^{(0)} = \mathbf{x}_i \quad (4)$$

By recursively applying the update rule in (3) over multiple layers, GNNs are able to capture both local and global structures of the graph. The final node embeddings, $\mathbf{h}_i^{(L)}$ after L layers, serve as comprehensive representations for various downstream tasks, including node classification, link prediction, and graph-level inference. This explicit integration of structural information, as evidenced in (1)–(4), enables GNNs to effectively harness the relational context intrinsic to graph data.

2.2 Graph attention networks

Graph Attention Networks (GATs) extend the framework of GNNs by incorporating an attention mechanism, which allows the network to assign different weights to nodes in a neighborhood during the aggregation process[34]. This is particularly useful in scenarios where some neighbors contribute more significantly than others to a node's representation.

In GAT, each node v_i with feature vector $\mathbf{h}_i$ is first transformed by a linear mapping $\mathbf{W}$. To compute the attention coefficients, an attention mechanism is applied to every pair of connected nodes. Specifically, for nodes v_i and v_j, the unnormalized attention score is given by:

$$e_{ij} = \mathrm{LeakyReLU}\left(\mathbf{a}^T \left[\mathbf{W}\mathbf{h}_i \| \mathbf{W}\mathbf{h}_j \right] \right) \quad (5)$$

where $\mathbf{a}$ is a learnable weight vector. These scores are then normalized using the softmax function to obtain the attention coefficients:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})} \quad (6)$$

The normalized coefficients α_{ij} indicate the importance of node v_i and v_j. Finally, the updated representation for node v_i is computed as a weighted sum of its neighbors' transformed features:

$$\mathbf{h}_{i'} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} \mathbf{W}\mathbf{h}_j \right) \quad (7)$$

To further enhance the stability and performance of the model, multi-head attention is commonly employed, where multiple independent attention mechanisms operate in parallel and their outputs are either concatenated or averaged. This design allows GATs to

dynamically modulate the influence of neighboring nodes, enabling the model to prioritize the most informative connections. As a result, the learned representations are more expressive and better suited for downstream tasks.

3 Methodology

Dollar-Cost Averaging (DCA) has emerged as an increasingly popular investment strategy, particularly among investors seeking to mitigate the inherent volatility of financial markets. By systematically investing a fixed amount at regular intervals, DCA reduces the risk associated with market timing, as it naturally smooths out the effects of short-term price fluctuations. This approach not only minimizes the potential for purchasing assets at their peak prices but also instills a disciplined, multi-day investment mindset that can be especially beneficial during periods of market instability.

However, the success of DCA extends beyond mere regularity in investments—it critically hinges on the selection of the right assets. While the strategy is inherently designed to counteract volatility, its effectiveness can be substantially enhanced when paired with predictive models that assess the multi-day potential of various stocks. Traditional predictive models tend to focus on short-term price movements, which may not align with the enduring goals of DCA. Therefore, there is a clear need to tailor these models specifically for DCA by incorporating a broader set of features that capture the sustained growth and resilience of financial assets over time. By customizing predictive frameworks to evaluate multi-day trends rather than ephemeral market noise, investors can maximize returns and improve the overall efficiency of their investment strategies. This tailored approach not only bolsters confidence in asset selection but also lays the groundwork for more stable and rewarding investment outcomes over the long haul.

3.1 The graph attention network for stock prediction

In predictive modeling, analyzing individual stocks in isolation is often inadequate due to the limited availability of reliable data and the presence of substantial noise, which can obscure meaningful patterns. A more comprehensive perspective that considers the broader market context is essential, as stocks do not operate independently but are inherently interrelated through various economic, industrial, and behavioral factors. Their collective dynamics often contain richer and more stable signals that can significantly improve prediction quality. GNNs offer a powerful framework for capturing these complex interdependencies. By modeling stocks as nodes within a graph and their relationships—such as correlations, sector affiliations, or co-movement trends—as edges, GNNs facilitate the development of an integrated market model. This graph-based representation not only addresses the limitations of sparse or noisy individual stock data but also capitalizes on the relational structure among assets to produce more robust and informative predictions, thereby enhancing the overall accuracy and reliability of financial forecasting.

While GNNs effectively capture relational structures by connecting stocks within a market, they typically aggregate information from all neighboring nodes in a uniform manner. This approach may overlook the varying degrees of influence that different stocks exert on one another. In contrast, GATs incorporate an attention mechanism that assigns adaptive weights to each neighboring node, thus highlighting the more informative relationships. This weighted aggregation enhances the model's ability to focus on critical interdependencies and attenuate the impact of noise or less relevant connections. In fact, GATs dynamically learn these weights during training, thereby improving the model's capacity to capture subtle market dynamics. To exploit these advantages, we have integrated a GAT layer into our methodology. This addition not only refines the aggregation process by emphasizing pivotal inter-stock relationships but also facilitates a more nuanced and accurate representation of the overall market structure.

For each node in the graph, the model incorporates multi-dimensional historical data of the corresponding stock, which may include, but is not limited to, features such as daily high, low, opening prices, and trading volume. This rich set of inputs allows the graph attention mechanism to capture not only the structural correlations between different stocks but also the temporal patterns and performance trends of each individual asset. By simultaneously considering inter-stock relationships and time-series behavior, the model gains a more comprehensive understanding of the market dynamics. This dual perspective enhances its ability to identify meaningful patterns, reduce the impact of noise, and ultimately improve prediction accuracy. A schematic representation of the overall model architecture is provided in Figure 2. Given that the prediction task in this context resembles a generation problem—though with a lower level of complexity—Mean Squared Error (MSE) is employed as the training loss to measure the discrepancy between predicted and actual values, as equation (8).

$$L_{\text{pred}} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2 \quad (8)$$

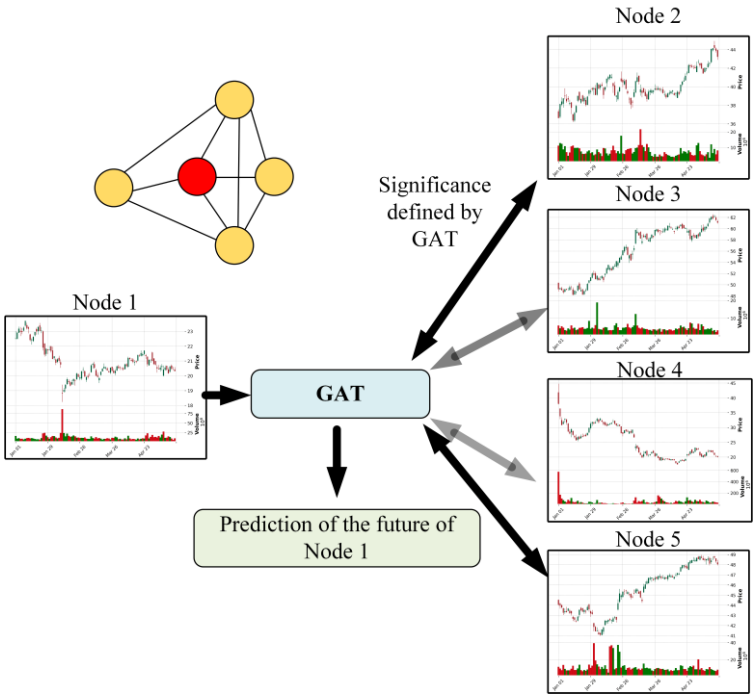


Figure 2. The diagram of GAT for stock prediction based on history of market.

3.2 Tailored design for dollar-cost averaging investment strategy

Financial market prediction poses a significant challenge due to the inherently volatile and nonlinear nature of market behavior. Sudden shifts driven by macroeconomic events, investor sentiment, or unforeseen external factors are commonplace, making consistent and reliable forecasting difficult. These challenges are particularly pronounced in the context of multi-day investment strategies such as DCA, which differs fundamentally from short-term or high-frequency trading. Unlike strategies that demand frequent recalibration based on market movements, DCA involves making periodic, fixed investments in selected assets over time—regardless of price fluctuations. This disciplined approach helps reduce emotional decision-making and cushions the impact of market volatility. However, its static nature limits the ability to respond to short-term opportunities, thereby placing greater emphasis on making strong, generalizable multi-day predictions at the outset.

Given these constraints, the design of predictive models for DCA must be approached with caution. Overreliance on narrowly defined features or recent market behavior can be detrimental, especially in volatile environments where transient signals may dominate the data. If the model concentrates too heavily on such patterns, it risks overfitting—capturing noise rather than structure, and failing to generalize in future scenarios. While GATs offer a powerful mechanism for modeling relational dependencies by assigning adaptive weights to neighboring nodes, they are not without shortcomings. In their standard form, GATs tend to focus attention disproportionately on nodes with strong short-term signals, which can amplify noisy or misleading correlations—particularly problematic in financial domains where causality is often ambiguous. Additionally, without proper regularization, the attention mechanism may become overly sensitive to minor variations in the input, leading to unstable representations. To counteract these issues, an effective model should minimize the risk of becoming overly influenced by any single factor or fleeting trend. Doing so encourages broader, more stable representations that better match the multi-day, noise-tolerant nature of DCA investment strategies.

Over-weighting a single factor or fleeting market blip creates an information bottleneck: when the latent representation fails to retain enough information to reconstruct the original inputs, essential signals are lost. Routine hyper-parameter tweaks—learning-rate schedules, deeper layers, extra attention heads—or even generic regularizes such as dropout and weight decay can temper, but seldom eliminate, this structural leak. To address it directly, we embed an autoencoder (AE) branch alongside the GAT. The requirement to compress each input sequence into a latent code and then faithfully rebuild it supplies a reconstruction loss that penalizes representations discarding informative dimensions, steering the attention mechanism toward broader, information-rich patterns. In the volatility-prone, noise-laden world of finance, this reconstruction-based regularization is particularly effective for preserving the robust multi-day signals that Dollar-Cost Averaging depends on.

To integrate an AE into the standard GAT workflow, we treat the GAT's node embeddings as the AE's latent

code (Figure 3). Raw market features are first compressed by an encoder into a lower-dimensional vector, which is fed to the GAT. The GAT learns cross-asset dependencies and passes its embeddings to (i) a dense prediction head and (ii) a decoder that tries to reconstruct the encoder's original input. Because the decoder operates directly on these post-attention embeddings, the reconstruction loss regularizes the GAT's output. We minimize a joint objective that combines the prediction loss with a reconstruction loss measured by MSE as Eq. (9). This dual-loss scheme preserves the familiar "GAT → dense layer" paradigm while regularizing the attention mechanism, preventing it from overreacting to short-lived market shocks and fostering information-rich, noise-robust representations. The resulting architecture, DCAGAT, thus balances representation integrity with attention flexibility and achieves consistently higher predictive accuracy in the multi-day DCA setting.

$$L_{rec} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{x}_i - x_i\right)^2 \quad (9)$$

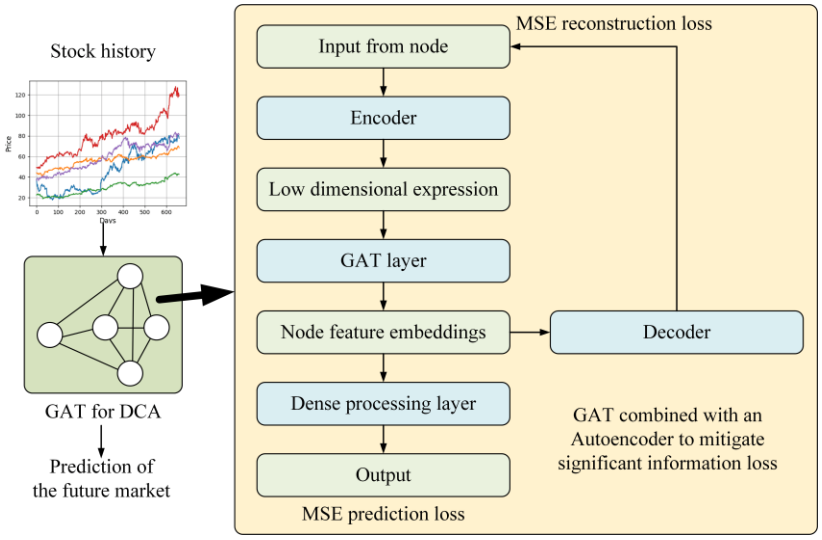


Figure: 3 The Diagram of DCAGAT.

Under this design, the model incurs two loss terms—one for prediction and one for reconstruction. Both terms are optimized jointly with user-defined weights, allowing the network to balance predictive accuracy against reconstruction fidelity. The overall objective is therefore the weighted sum:

$$L_{total} = \lambda_1 L_{pred} + \lambda_2 L_{rec} \quad (10)$$

promoting simultaneous improvement on both fronts.

In our model, both the prediction loss and the reconstruction loss are assigned a default weight of 1, ensuring that they contribute equally during the training process. This balanced weighting scheme is designed to optimize both the accuracy of the predictions and the quality of the reconstructed inputs. In the subsequent section, we conduct an ablation study to systematically evaluate the impact of the reconstruction component on overall model performance, shedding light on how this mechanism contributes to mitigating overfitting and enhancing the model's ability to generalize in complex market conditions. In summary, the modular algorithm description is as Algorithm 1:

Algorithm 1: DCAGAT Forward & Training Loop	
Input : node features X ∈ R^{N×T×F}, adjacency A ∈ {0,1}^{N×N}	
Output: predictions ŷ, reconstruction R	
H ← ENCODER(X)	# (N,T,F) → (N,d)
Z ← GAT (H, A)	# node embeddings
ŷ ← PREDICT_HEAD(Z)	# three targets per node
R ← DECODER(Z)	# reconstruct H
Lpred = MSE(ŷ, y)	# prediction loss
Lrecon = MSE(R, H)	# reconstruction loss
Ltotal = λp·Lpred + λr·Lrecon	
Update parameters with Adam	# minimise Ltotal

4 Experiments

4.1 Experimental setting

Yahoo Finance dataset provides historical financial data for a wide range of stocks and widely used for verification of stock prediction methods[35], as depicted in Figure 4. This dataset includes daily records of key features such as opening price, closing price, highest price, lowest price, and trading volume, spanning from January 1, 2012, to December 31, 2022.



Figure 4: A part of historical closing prices of stocks in yahoo finance dataset.

Besides, ETF dataset contains historical time-series records for one or more exchange-traded funds, restricted to daily market price fields (open, high, low, close) and trading volume over the period from 1993 to 2025. Such a dataset provides a consistent basis for benchmarking experimental studies. By focusing solely on these elements within the specified interval, the dataset remains well-defined and straightforward to process. Because the elements are identical, we omit the visualization here.

For our experiment, we focus on a selection of 20, 30, 50 stocks, chosen to represent a diverse and broad cross-section of the market. By incorporating this number of stocks, we create a comprehensive market picture that allows our model to capture inter-stock relationships and market-wide dynamics. For each stock, the past 30 days of records are provided. This dataset is particularly useful for our study, as it reflects the complex and noisy nature of real-world financial data, providing an ideal testbed for evaluating the predictive performance of DCAGAT. The wide variety of stocks ensures that our model is challenged to discern meaningful patterns across different sectors and market conditions.

It is important to acknowledge that due to the inherent volatility and randomness of the stock market, relying on market conditions from 2010 to predict outcomes in 2020 would be highly inappropriate. Instead, we have structured the dataset in a more meaningful manner by grouping the data into segments where three consecutive years of market information are used to forecast the market situation for the subsequent fourth year. This approach not only aligns more closely with the dynamic nature of financial markets but also ensures that the training and testing datasets reflect more coherent and temporally relevant trends. The detailed breakdown of these training and testing periods is presented in the Table II.

Table 2: The training and testing dataset

Subsets	Training dataset time period	Testing dataset time period
1	2012-01-01 to 2014-12-31	2015-01-01 to 2015-12-31
2	2015-01-01 to 2017-12-31	2018-01-01 to 2018-12-31
3	2018-01-01 to 2021-12-31	2022-01-01 to 2022-12-31

To conduct a comparative experiment, we reviewed several studies in the literature where GNNs and GATs have been used for stock prediction. However, there is limited research on applying these models to DCA investment strategies. Directly applying the methods from these existing studies would likely not yield good results, and such comparisons would have limited value. As such, our approach is to demonstrate the distinctions

among the various mechanisms we employed—namely, traditional GNNs, the Graph Attention Mechanism, ST-GCN[36] (which is a SOTA combined convolutional layer in GNN and used for stock prediction), and our proposed DCAGAT model. This comparative analysis not only elucidates the unique advantages of each technique but also emphasizes the superior performance of DCAGAT in addressing the specific challenges associated with DCA strategies. By tailoring our methods to the multi-day, disciplined nature of DCA, we provide a more relevant and robust framework for evaluating stock selection and prediction in volatile financial markets.

Another important consideration is the evaluation criteria used to assess model performance. In conventional financial forecasting, metrics such as next-day price direction or pointwise prediction accuracy are commonly employed and serve as useful indicators of short-term forecasting capability. However, these metrics fall short when applied to DCA strategies, which emphasize multi-day consistency over immediate gains. DCA inherently downplays short-term fluctuations, making it necessary to adopt evaluation methods that reflect this investment philosophy. Therefore, beyond standard predictive accuracy, we introduce metrics that better align with DCA's temporal horizon. Specifically, we assess the model's ability to predict the direction of returns over medium-range windows—namely, 7-day and 14-day periods. These evaluation windows are designed to capture more stable market trends while reducing sensitivity to daily noise, thereby offering a more meaningful benchmark for DCA-oriented decision-making. The directional accuracy for these timeframes is computed using the following equation:

$$\mathrm{A}_{\mathrm{dir},k} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{I}\left(\mathrm{sgn}(\hat{r}_{i,t+k}) = \mathrm{sgn}(r_{i,t+k})\right) \quad (11)$$

In equation (11), I is an indicator function that returns 1 if the predicted and actual directions match, and 0 otherwise. $\hat{r}_{i,t+k}$ and $r_{i,t+k}$ are the predicted and actual return sum for stock i on day $t+k$.

To ensure comparability, the stock prices in the original dataset are adjusted to reflect the percentage change. Furthermore, the primary role of a predictive model in stock market investment is typically to assist in stock selection, rather than predicting exact price values. Therefore, prediction accuracy alone is not of primary interest to end-users. Rather, what is of greater importance is the model's ability to recommend stocks. To this end, we introduce an additional evaluation metric: stock selection accuracy. This metric assesses the model's ability to correctly identify the top 5 performing stocks over the next 14 days, providing a more practical and realistic evaluation of its predictive power. This metric can be expressed as:

$$\mathrm{A}_{\mathrm{select}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{I}\left(\hat{S}_{t+14}^{(5)} \cap S_{t+14}^{(5)}\right) \quad (12)$$

$\hat{S}_{t+14}^{(5)}$ denote the set of the top 5 stocks by return on day 14, and $S_{t+14}^{(5)}$ denote the predicted set of the top 5 stocks.

Stability across time is essential, so we measure temporal consistency by computing the standard deviation of stock-selection accuracy $\mathrm{A}_{\mathrm{select}}$ across successive test windows. Each prediction is generated on inputs drawn from a different point in time; the resulting dispersion therefore captures how performance fluctuates under varying market conditions. All evaluation criteria for the model are calculated using testing datasets.

4.2 Experimental results

All experiments were run on an NVIDIA TITAN RTX GPU. Tables III–VIII summaries the results of the DNN, GNN, GAT, ST-GCN, and DCAGAT models on the Yahoo Finance and ETF datasets listed in Table II. Full architectural details of DCAGAT appear in Appendix I. Unless noted otherwise, each model was trained for 100 epochs with the Adam optimizer (learning rate 1×10^{-5}) and a mini-batch size of 32—settings that provided stable convergence in preliminary trials.

While interdependencies among stocks do exist, these relationships are not always immediately obvious. To mitigate the risk of inadvertently introducing overfitting factors due to hidden correlations, we opted to fully connect all nodes in the graph. In practice, this means that the adjacency matrix used as input to the network is an all-ones matrix with dimensions equal to the number of stocks, thereby ensuring that every possible inter-stock dependency is considered. This approach not only simplifies the network structure but also reinforces the model's ability to capture a broad spectrum of market dynamics by accounting for all potential interactions among stocks.

Table 3: The experimental results on 50 stocks in Yahoo

Dataset	Subset 1(%)					Subset 2(%)					Subset 3(%)				
Models	$A_{\mathrm{dir},1}$	$A_{\mathrm{dir},7}$	$A_{\mathrm{dir},14}$	A_{select}	*Std*	$A_{\mathrm{dir},1}$	$A_{\mathrm{dir},7}$	$A_{\mathrm{dir},14}$	A_{select}	*Std*	$A_{\mathrm{dir},1}$	$A_{\mathrm{dir},7}$	$A_{\mathrm{dir},14}$	A_{select}	*Std*
DNN	49.4	53.8	**55.2**	10.77	0.14	52.3	54.9	56.5	11.2	0.13	50.4	**50.8**	49.2	10.82	0.15
GNN	47.7	48.9	52.8	11.35	0.15	**52.7**	**55.0**	**56.6**	8.41	0.1	49.9	50.4	49.9	6.02	0.10
GAT	49.6	51.0	52.7	8.75	0.12	49.7	49.1	50.3	10.21	0.14	**52.8**	50.7	53.0	9.66	0.13
ST-GCN	50.	48.	47.	11.3	0.1	47.	48.	47.	8.41	0.1	50.	48.	**60.**	6.09	0.1

	6	1	5	5	5	6	7	8		0	1	8	**8**		0
DCAGAT	51.1	49.7	54.7	**12.12**	0.13	52.1	49.6	51.7	**11.43**	0.13	51.8	50.3	50.6	**11.82**	0.13

Table 4: The experimental results on 30 stocks in Yahoo

Dataset	Subset 1(%)					Subset 2(%)					Subset 3(%)				
Models	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*
DNN	49.9	54.2	56.4	12.10	0.15	52.6	56.2	57.9	13.24	0.13	50.0	50.1	51.1	13.03	0.17
GNN	50.6	53.5	55.6	15.00	0.15	52.0	**56.2**	57.7	14.20	0.13	49.8	51.6	50.9	15.07	0.14
GAT	50.1	**49.9**	52.8	14.35	0.16	51.8	49.5	55.4	15.85	**0.15**	**49.0**	**49.7**	53.3	16.04	0.16
ST-GCN	49.3	**45.6**	56.4	15.00	0.15	**47.1**	56.2	42.2	14.20	**0.13**	**50.5**	**52.1**	51.4	15.07	0.14
DCAGAT	**50.3**	50.1	**55.1**	**15.29**	0.14	51.0	49.6	**52.9**	**17.39**	0.16	48.8	51.3	**52.2**	**19.52**	0.16

Table 5: The Experimental results on 20 stocks in Yahoo

Dataset	Subset 1(%)					Subset 2(%)					Subset 3(%)				
Models	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*
DNN	49.4	52.8	54.1	20.92	0.17	53.0	54.4	**55.7**	21.57	0.17	51.4	**52.5**	50.9	24.93	0.16
GNN	**51.9**	**53.3**	**54.7**	21.71	0.16	**53.1**	**54.8**	54.3	22.30	0.16	51.4	51.4	50.3	25.12	0.16
GAT	49.3	51.6	52.0	23.27	0.16	49.4	48.6	51.5	21.20	0.17	49.3	52.1	52.1	26.30	0.18
ST-GCN	50.2	47.2	49.5	23.30	0.16	50.1	50.3	49.2	22.42	0.16	**51.7**	49.2	57.8	25.10	0.16
DCAGAT	46.4	52.2	53.4	**23.65**	0.15	49.0	50.1	53.8	**23.96**	0.17	51.3	47.7	**51.6**	**27.63**	0.17

Table 6: The Experimental results on 50 stocks in ETF

Dataset	Subset 1(%)					Subset 2(%)					Subset 3(%)				
Models	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*
DNN	49.2	53.9	**55.6**	12.88	0.13	**52.5**	**55.3**	**57.2**	13.33	0.14	50.1	51.8	51.0	13.53	0.14
GNN	48.4	**54.1**	54.2	9.42	0.11	49.6	51.8	55.2	14.40	0.13	**52.5**	**52.1**	50.6	13.91	0.14
GAT	47.9	51.8	53.1	12.60	0.15	49.4	47.7	52.5	12.27	0.15	51.1	50.3	**55.8**	10.24	0.13
ST-GCN	**49.8**	45.9	55.7	9.42	0.11	51.3	55.0	51.2	14.40	0.13	50.2	51.9	51.0	13.91	0.14
DCAGAT	47.5	50.8	52.3	**13.46**	0.14	50.3	50.1	55.5	**14.75**	0.15	52.4	50.3	53.7	**14.62**	0.16

Table 7: The Experimental Results on 30 Stocks in ETF

Dataset	Subset 1(%)					Subset 2(%)					Subset 3(%)				
Models	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*
DNN	50.0	54.0	55.7	17.79	0.15	52.7	**55.8**	57.6	18.65	0.15	50.3	51.9	51.4	18.65	0.16

GNN	47.1	**54.4**	**55.9**	15.48	0.14	49.6	54.7	**57.9**	19.23	0.15	49.7	51.7	50.9	17.10	0.15
GAT	51.7	49.0	54.4	17.46	0.15	49.8	50.4	53.3	17.29	0.15	**53.8**	49.8	51.2	20.29	0.16
ST-GCN	48.1	53.5	54.0	15.48	0.14	**53.9**	50.4	44.6	19.23	0.15	50.3	**52.0**	51.3	17.10	0.15
DCAGAT	**50.7**	49.7	52.8	**18.96**	0.14	51.5	49.6	53.9	**20.65**	0.16	48.6	48.6	**52.4**	**20.68**	0.17

Table 8: The experimental results on 20 stocks in ETF

Dataset	Subset 1(%)					Subset 2(%)					Subset 3(%)				
Models	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*	$A_{\text{dir},1}$	$A_{\text{dir},7}$	$A_{\text{dir},14}$	A_{select}	*Std*
DNN	51.0	53.4	55.2	20.10	0.14	52.0	**55.0**	**56.7**	16.33	0.13	50.5	**52.3**	51.3	20.22	0.16
GNN	50.0	**54.3**	**55.3**	19.33	0.14	49.6	54.0	54.7	15.27	0.12	48.8	51.9	51.3	23.29	0.15
GAT	**51.9**	48.5	53.1	20.05	0.17	50.9	49.4	54.1	17.25	0.17	50.4	50.7	50.5	23.00	0.17
ST-GCN	49.7	50.5	47.4	19.33	0.14	**52.6**	53.1	49.7	15.27	0.12	50.4	52.2	49.2	23.29	0.15
DCAGAT	48.6	49.5	51.1	**21.35**	0.16	47.0	51.6	55.9	**22.61**	0.17	**50.6**	47.5	**53.3**	**25.60**	0.17

From the results, it is evident that the overall prediction accuracy remains relatively low—a reflection of the inherent randomness and volatility present in financial markets. Although increasing the number of stocks in the analysis makes the task of stock selection more challenging, it does not significantly affect the accuracy of directional predictions. This observation underscores the immense difficulty in forecasting stock prices, given that markets are influenced by a multitude of unpredictable factors, and it should always be clear that perfect prediction is a fantasy. Despite these modest performance levels, the results offer valuable insights into the comparative effectiveness of the various models tested, highlighting both the strengths and limitations of each approach in capturing the complex dynamics of the market.

First, we observe that the DCAGAT architecture yields a clear improvement in stock selection accuracy. As shown in Figure 5, the GAT-based models deliver notable improvements in stock selection—a critical and practical metric—while directional accuracy, though informative, is not the core concern; hence, subsequent figures focus only on selection accuracy. Therefore, subsequent figures present only selection accuracy. The proposed DCAGAT model outperforms alternative approaches in this respect, underscoring the benefit of integrating an autoencoder mechanism to preserve and reconstruct salient information. Although GNN and ST-GCN differ in directional prediction, their stock selection accuracies are comparable. While DCAGAT does not uniformly attain the highest prediction accuracy across all metrics (with DNN occasionally matching or exceeding performance), its superior selection accuracy highlights its enhanced practical value for investment decision-making, offering a more reliable strategy compared to competing models. The small standard-deviation values reported in Tables III–VIII indicate that the model's performance varies only marginally across successive evaluation windows, confirming that it remains stable throughout the entire test period.

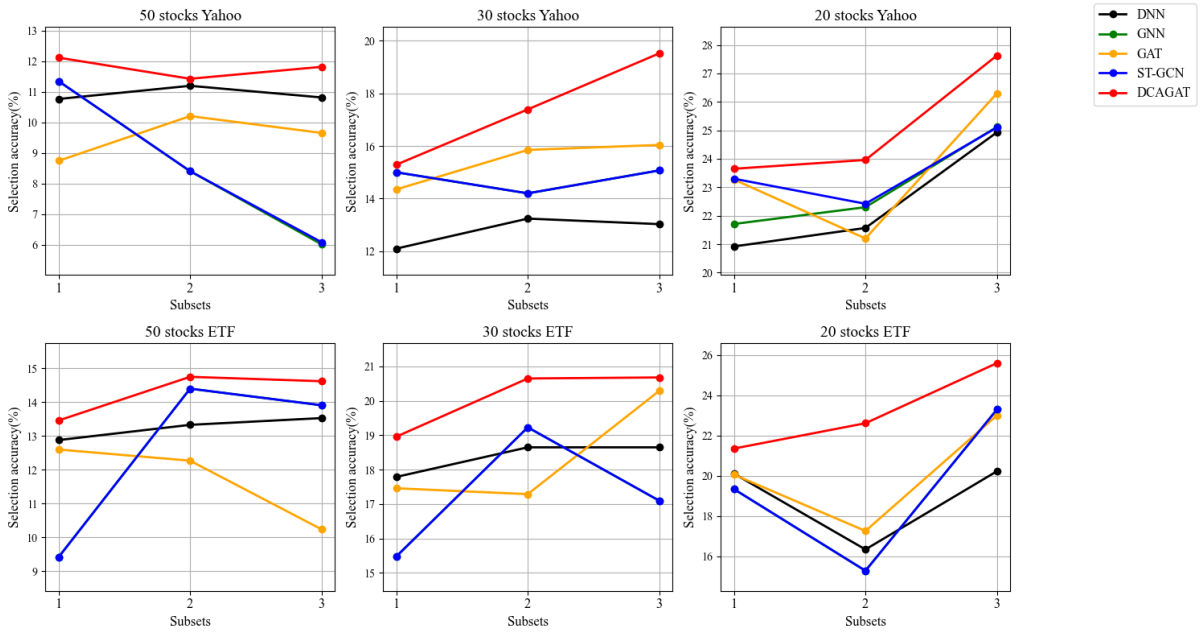


Figure 5: The visualization of aselect.

Although the performance gap in Figure. 5 is visually compelling, we verified its statistical validity with a paired t-test. Because the remaining baselines differ only marginally from one another, we compared our model solely with ST-GCN. For every experimental condition—two datasets and three portfolio sizes—we collected Top-5 hit-rates from the same 15 runs, yielding perfectly matched score pairs. The resulting p-values (Table IX) are all below the 0.05 threshold, allowing us to reject the null hypothesis of equal means and to conclude that DCAGAT's advantage in stock selection is statistically significant across all market settings.

Table 9: Paired t-test results demonstrating the statistical significance of DCAGAT vs. ST-GCN

p-values	Yahoo			ETF		
	Subset 1	Subset 2	Subset 3	Subset 1	Subset 2	Subset 3
50	0.021	0.002	0.002	0.000	0.023	0.013
30	0.032	0.012	0.001	0.000	0.002	0.000
20	0.035	0.002	0.012	0.001	0.000	0.015

Additionally, the number of stocks included in the prediction task plays a significant role in model performance. Empirically, we observe that models tend to achieve higher prediction accuracy when operating on a smaller set of stocks. This phenomenon is closely tied to the inherent randomness and noise present in financial markets. While expanding the stock pool may introduce more information and reveal broader market relationships, it simultaneously increases the complexity and noise within the data. The added noise can obscure useful patterns and make it more difficult for the model to generalize effectively. Therefore, it is reasonable that performance improvements may plateau—or even diminish—as the number of stocks increases. This observation highlights a key trade-off in financial modeling: expanding coverage can enhance representational richness, but it also demands more robust mechanisms to filter out irrelevant or misleading signals.

Although the observed improvements may not appear dramatic in absolute terms, they are nonetheless meaningful in the context of quantitative trading, where even small performance gains can translate into significant financial impact over time. This highlights the practical value of incorporating the proposed mechanism into the predictive model. In real-world trading systems, modest enhancements in stock selection accuracy or return direction prediction can compound across multiple trades and time periods, ultimately leading to notable returns. Therefore, the consistent edge provided by our model—however incremental—makes DCAGAT a valuable tool for financial applications, especially in strategies like DCA where multi-day consistency and robustness are prioritized. Such improvements reinforce the relevance of our approach in real trading environments, where every slight advantage can contribute to sustained strategic success.

5 Discussion

5.1 Weight configuration

Because the performance gaps we observe on the Yahoo and ETF datasets are statistically indistinguishable, the remainder of our analysis focuses on the Yahoo data set only. Here, we examine the weight configuration between prediction loss and reconstruction loss using three ratios: 1:1 (default), 1:2, and 2:1. These settings are designed to emphasize different objectives—1:2 leans toward reconstruction, while 2:1 favors prediction. The 1:1 ratio provides a balanced baseline, but varying the weights helps explore trade-offs and identify the optimal focus based on specific dataset characteristics.

As shown in Table X, the default 1:1 weight setting offers a competitive and balanced baseline, yielding strong performance across a variety of scenarios. This balanced approach ensures that the model pays equal attention to predictive accuracy and representational consistency. However, it is also evident that the 1:1 ratio is not universally optimal. In some cases, alternative weight combinations outperform the default setting, suggesting that the relative importance of prediction versus reconstruction can vary depending on dataset characteristics, noise levels, and task complexity. This highlights the need for adaptive or task-specific tuning of the loss weights rather than relying on fixed values. Overall, the findings suggest that while the 1:1 ratio serves as a solid default, a more nuanced calibration of loss components may be necessary to achieve optimal results in different financial contexts.

Table 10: The weight ratio experimentation and results on 50 stocks

DCAGAT	Subset 1(%)				Subset 2(%)				Subset 3(%)			
Weight ratio	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}
1:1	51.1	49.7	**54.7**	**12.12**	52.1	49.6	51.7	11.43	**51.8**	**50.3**	50.6	**11.82**
1:2	49.4	**50.4**	52.7	9.62	**52.5**	**50.9**	**52.6**	11.01	51.4	**50.3**	51.6	11.59
2:1	**54.4**	49.5	52.5	10.52	51.7	47.4	51.9	**12.83**	51.7	47.4	**51.9**	10.95

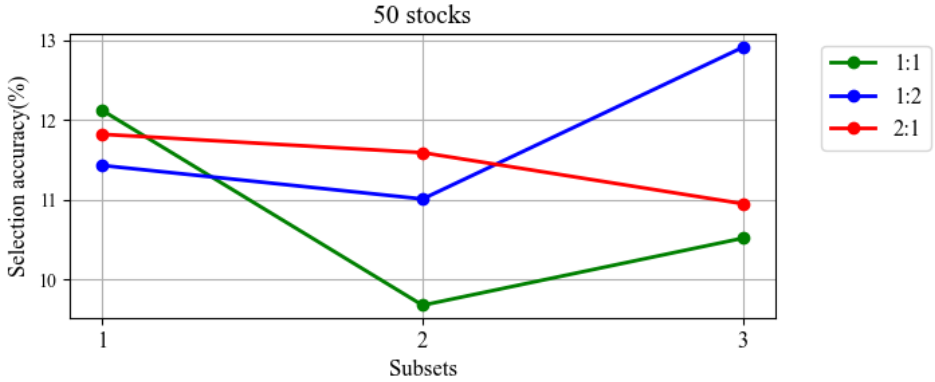


Figure 6: The visualization of Aselect under different weight configurations.

As shown in Figure 6, for stock selection accuracy, the 2:1 weight ratio outperforms the default 1:1 setting in Subset 2. This outcome is not unexpected, given that model performance is intimately linked to the underlying data characteristics. The results underscore a fundamental aspect of data-driven problems: the data itself plays a critical role in determining outcomes. This finding emphasizes the importance of customizing model parameters to align with the specific attributes of the dataset, reinforcing that data significantly influences the effectiveness of predictive models, particularly in stock market forecasting.

In the context of stock market forecasting, the inherent characteristics of stock data—such as volatility, trends, and interdependencies between stocks—play a pivotal role in determining model performance. Financial markets are dynamic and subject to constant fluctuations, meaning that model parameters, including the weight ratio between prediction and reconstruction, must be carefully calibrated to align with the underlying patterns and noise present in the data. Consequently, the success of stock prediction models is not solely reliant on the sophistication of their architectures but also on their ability to adapt effectively to the unique attributes of the training data. This adaptability ensures that models are better equipped to manage the complexities and uncertainties inherent in real-world financial environments.

5.2 Time configuration

In previous experiments, the historical window length was fixed at 30 days, which aligns with common investor routines. However, the length of historical data used for prediction can significantly impact model performance. To better understand this influence, we compare results

using four different time windows: 15 days, 30 days, 45 days, and 60 days. This variation allows us to observe how the amount of past information affects the model's ability to capture trends and make accurate predictions. However, the results are presented in Table VI. In terms of prediction accuracy, there is no clear indication that any specific time window consistently outperforms the others. As shown in Figure 7, the 30-day time window appears to yield slightly better selection accuracy overall, but the advantage is marginal and not conclusive.

Table 11: the time window configuration experimentation and results on 50 stocks

DCAGAT	Subset 1(%)				Subset 2(%)				Subset 3(%)			
Time window	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}
15 days	**51.2**	53.4	**56.1**	10.76	51.8	**54.9**	56.4	9.82	**54.6**	50.8	47.3	**12.88**
30 days	51.1	49.7	54.7	**12.12**	52.1	49.6	51.7	**11.43**	51.8	50.3	50.6	11.82
45 days	49.6	**54.4**	55.7	11.71	50.2	51.0	53.0	10.52	48.3	**53.1**	53.0	10.94
60 days	47.9	54.0	55.9	11.91	**53.1**	51.7	53.7	11.41	49.9	49.2	**53.3**	10.73

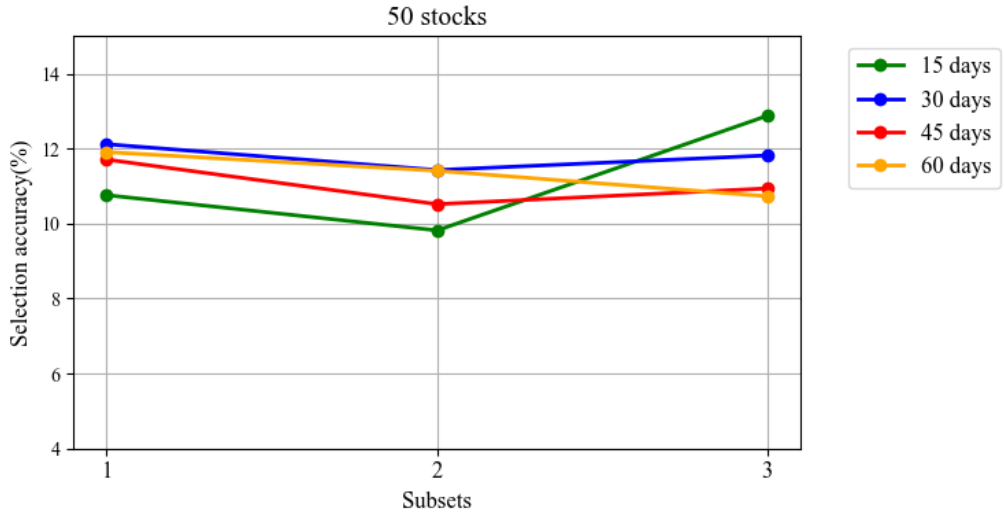


Figure 7: The visualization of Aselect under different time window configurations.

These findings suggest that there is no universal rule for determining the optimal length of historical data—the effectiveness of a given window largely depends on the characteristics of the underlying data. For certain stocks or market conditions, longer histories may be beneficial in capturing extended trends, delayed reactions, or cyclical patterns—such as those influenced by quarterly financial reports—making a 45- or 60-day window more effective. Conversely, shorter windows like 15 days may reduce noise and enhance responsiveness for stocks that are highly volatile or sensitive to recent news. This reinforces the importance of data-driven configuration, indicating that the ideal time window should be learned through experimentation and tailored to the specific context, rather than predefined.

5.3 Fully connected adjacency matrix graph

In our experiments we adopt a fully connected graph, thereby modelling every pair-wise stock interaction. While this choice guarantees that no potential dependency is omitted, it also admits many weak or spurious links, which can increase the risk of over-fitting. A natural next step is to repeat the analysis with a scarified graph—e.g., retaining only edges whose historical return correlation exceeds a threshold—and compare the resulting performance against the fully connected baseline.

Table 12: Impact of correlation-threshold sparsification on DCAGAT performance

CAGAT	Subset 1(%)				Subset 2(%)				Subset 3(%)			
threshold	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}	$A_{\text{dir, 1}}$	$A_{\text{dir, 7}}$	$A_{\text{dir, 14}}$	A_{select}
-	51.1	49.7	54.7	**12.12**	52.1	49.6	51.7	**11.43**	51.8	50.3	50.6	11.82
>0.3	49.9	55.6	57.0	11.35	52.4	54.3	56.3	9.51	49.7	49.9	48.3	13.04
>0.5	49.3	53.8	55.2	11.15	52.6	54.9	56.6	8.79	49.9	50.4	49.**9**	**13.62**

Table X shows that sparsifying the graph by correlation threshold has only a marginal effect on DCAGAT's overall performance. None of the differences in A _select across thresholds exceeds the standard-error band, indicating no statistically significant change in stock-selection accuracy. Nevertheless, the > 0.30 and > 0.50 graphs do yield small gains in specific subsets (e.g., A _select rises from 11.43 % to 13.04 % in Subset 3 at the 0.30 threshold). These modest, scenario-dependent improvements suggest that the correlation cut-off can be treated as a tunable hyper-parameter: selecting a threshold tailored to a particular market segment may provide incremental gains without materially altering the model's stability.

5.4 Practical deployment and limitations

In deployment, a single hyper-parameter sweep and minor architectural tweaks can lift metrics, demonstrating that practical adjustment does yield better performance. Yet two limitations remain clear: (i) for ultrashort horizons a vanilla GAT can still outscore DCAGAT, and (ii) even the best-tuned model identifies only 10 % of the top performers. These outcomes expose the market's chronically low signal-to-noise ratio, where external forces—especially investor sentiment—often drown out historical price patterns. Reliance on past prices alone therefore cannot sustain reliable near future forecasts. Lasting progress will hinge less on deeper model stacks and more on superior data pipelines: meticulous cleaning, richer temporal structures, and domain-aware features.

6 Conclusion

This paper presents a novel financial market prediction model, DCAGAT, which combines Graph Attention Networks with multi-feature fusion and an autoencoder-based reconstruction mechanism to address the challenges posed by the dollar-cost averaging investment strategy. The integration of GAT allows for adaptive weighting of stock relationships, enhancing the model's ability to capture meaningful market interdependencies. Our experiments demonstrate that DCAGAT significantly improves stock selection accuracy and prediction performance compared to traditional models, making it a promising tool for multi-day investment strategies. Despite the inherent volatility and randomness of financial markets, the results show that even modest improvements in prediction accuracy can have meaningful implications for quantitative trading, making DCAGAT a valuable approach for real-world financial applications.

Future work should place greater emphasis on data, as our experiments show that model performance is fundamentally constrained by the quality, structure, and relevance of the input information. Rather than relying solely on architectural enhancements, integrating richer data sources—such as macroeconomic indicators, industry reports, and market sentiment extracted from news or social media—could offer broader context and help mitigate noise. In parallel, more adaptive learning strategies like reinforcement learning could be explored to enable the model to dynamically adjust to evolving market conditions. Expanding the model's capability to perform multi-step predictions over extended time horizons would further improve its suitability for multi-day strategies such as DCA. Lastly, validating the model in real-time trading environments is critical to assessing its practical reliability and robustness, and to ensure that it can transition from a promising research framework to a viable tool for real-world financial decision-making.

Acknowledgement

The authors would like to thank all collaborators and reviewers for their valuable feedback and support throughout this research. This work was conducted with funding from the Department of Education of Guizhou Province, and the authors declare no conflicts of interest.

Appendix I

Table 1: Detail of DCAGAT

Layer	shape	Connected to
InputLayer	(None, market size, history length, 5)	/
Dense1	150, ReLU	InputLayer
Dense2	8, ReLU	Dense1
Adjacency Input	(None, market size, market size)	/
GAT	Channels = 8, Heads = 4, ReLU	[Dense2, Inputgraph]
Dense3	150, ReLU	GAT
Reconstruction Output	(None, market size, history length, 5)	Dense3
OutputLayer	(None, market size, 3)	GAT

Reference

[1] T. Marwala, "Opportunity Seeking Versus Risk Aversion," in The Balancing Problem in the Governance of Artificial Intelligence: Springer, 2024, pp. 51-67, DOI: 10.1007/978-981-97-9251-1_4.

[2] A. Rangapur, H. Wang, and K. Shu, "Investigating online financial misinformation and its consequences: A computational perspective," arXiv preprint arXiv:2309.12363, 2023.

[3] A. Zik-Rullahi, I. Jide, and E. Onuh, "Behavioural Finance: Exploring the Psychology and Economic Aspects of Financial Decision-Making," European Review in Accounting and Finance, 7 (3), 50-64. DOI: https://doi. org/10.5281/zenodo, vol. 8320116, 2023.

[4] N. Ahmad, L. Youjin, S. Žiković, and Z. Belyaeva, "The effects of technological innovation on sustainable development and environmental degradation: Evidence from China," Technology in Society, vol. 72, p. 102184, 2023, DOI: 10.1016/j.techsoc.2022.102184.

[5] T. Sugadev, N. S. Hameed, S. Vijayakumar, P. Tamilarasan, and M. S. Islam, "Portfolio optimization using machine learning techniques," in 2023 4th International Conference on Computation, Automation and Knowledge Management (ICCAKM), 2023: IEEE, pp. 1-7, DOI: 10.1109/ICCAKM58659.2023.10449598.

[6] Y. Li, "Essays in financial technology: banking efficiency and application of machine learning models in Supply Chain Finance and credit risk assessment," University of Glasgow, 2024.

[7] T. Kongkaew and C. Saengpao, "Investment Strategies in Covid-19 Pandemic: Comparative Analysis of Stock Investment Performance in the SETTHSI Index and the SET100 Index Utilizing Dollar Cost Averaging (DCA), Lump Sum (LS), and Value Averaging (VA) Strategies," Journal of Business, Innovation and Sustainability (JBIS), vol. 19, no. 2, 2024.

[8] E. Vasileiou and E. Hadad, "Benefits and drawbacks of the cost average plan as an alternative investment strategy in EMU countries," Cogent Economics & Finance, vol. 12, no. 1, p. 2426539, 2024, DOI: 10.1080/23322039.2024.2426539.

[9] H. C. Carvalho, "Riding the Waves A Comprehensive Study of 20/50 EMA and Dollar Cost Averaging in Equities, Stock Indexes, and Forex Market," Universidade NOVA de Lisboa (Portugal), 2024.

[10] T. Kaneko, H. Kameoka, K. Tanaka, and N. Hojo, "Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion," in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019: IEEE, pp. 6820-6824, DOI: 10.1109/ICASSP.2019.8682897.

[11] J. Gu, T. S. Yang, J. C. Ye, and D. H. Yang, "CycleGAN denoising of extreme low-dose cardiac CT using wavelet-assisted noise disentanglement," Medical image analysis, vol. 74, p. 102209, 2021, DOI: 10.1016/j.media.2021.102209.

[12] Z. Dai, L. Zhao, K. Wang, and Y. Zhou, "Mode standardization: A practical countermeasure against mode collapse of GAN-based signal synthesis," Appl. Soft Comput., vol. 150, p. 111089, 2024, DOI: 10.1016/j.asoc.2023.111089.

[13] J. Deng, G. Pang, Z. Zhang, Z. Pang, H. Yang, and G. Yang, "cGAN based facial expression recognition for human-robot interaction," IEEE Access, vol. 7, pp. 9848-9859, 2019, DOI: 10.1109/ACCESS.2019.2891668.

[14] X. Hu, Y. Cao, T. Tang, and Y. Sun, "Data-driven technology of fault diagnosis in railway point machines: review and challenges," Transportation Safety and Environment, vol. 4, no. 4, 2022, doi: 10.1093/tse/tdac036.

[15] L. Cao, "Ai in finance: challenges, techniques, and opportunities," ACM Computing Surveys (CSUR), vol. 55, no. 3, pp. 1-38, 2022, DOI: 10.1145/3502289.

[16] Y. Liu and B. Pan, "Profit estimation model and financial risk prediction combining multi-scale convolutional feature extractor and BGRU model," Informatica, vol. 48, no. 11, 2024, DOI: 10.31449/inf.v48i11.5941.

[17] R. K. Dash, T. N. Nguyen, K. Cengiz, and A. Sharma, "Fine-tuned support vector regression model for stock predictions," Neural Computing and Applications, vol. 35, no. 32, pp. 23295-23309, 2023, DOI: 10.1007/s00521-021-05842-w.

[18] J. C. Tellez Gaytan et al., "AI-based prediction of capital structure: Performance comparison of ANN SVM and LR models," Computational intelligence and neuroscience, vol. 2022, no. 1, p. 8334927, 2022, DOI: 10.1155/2022/8334927.

[19] S. Ahmed, M. M. Alshater, A. El Ammari, and H. Hammami, "Artificial intelligence and machine learning in finance: A bibliometric review," Research in International Business and Finance, vol. 61, p. 101646, 2022, DOI: 10.1016/j.ribaf.2022.101646.

[20] T. Šestanović and T. Kalinić Milićević, "Identification of the Optimal Neural Network Architecture for Prediction of Bitcoin Return," Informatica, vol. 36, no. 1, pp. 175-196, 2025, DOI: 10.15388/24-INFOR561.

[21] C. Milana and A. Ashta, "Artificial intelligence techniques in finance and financial markets: a survey of the literature," Strategic Change, vol. 30, no. 3, pp. 189-209, 2021, DOI: 10.1002/jsc.2403.

[22] S. Joshi, "Review of Gen AI models for financial risk management," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 11, no. 1, pp. 709-723, 2025, DOI: 10.32628/CSEIT2511114.

[23] H. Pallathadka, E. H. Ramirez-Asis, T. P. Loli-Poma, K. Kaliyaperumal, R. J. M. Ventayen, and M. Naved, "Applications of artificial intelligence in business management, e-commerce and finance," Materials Today: Proceedings, vol. 80, pp. 2610-2613, 2023, DOI: 10.1016/j.matpr.2021.06.419.

[24] D. Mohanty, A. K. Parida, and S. S. Khuntia, "Financial market prediction under deep learning framework using auto encoder and kernel extreme learning machine," Appl. Soft Comput., vol. 99, p. 106898, 2021, DOI: 10.1016/j.asoc.2020.106898.

[25] Y. Cheng, G. Li, X. Zhou, and S. Ye, "Research on Time Series Forecasting Models Based on Hybrid Attention Mechanism and Graph Neural Networks," Informatica, vol. 49, no. 21, 2025, DOI: 10.31449/inf.v49i21.7580..

[26] K. H. Tuhin, A. Nobi, M. H. Rakib, and J. W. Lee, "Long short-term memory autoencoder based network of financial indices," Humanities and Social Sciences Communications, vol. 12, no. 1, p. 100, 2025/01/29 2025, doi: 10.1057/s41599-025-04412-y.

[27] B. Bieganowski and R. Slepaczuk, "Supervised Autoencoder MLP for Financial Time Series Forecasting," arXiv preprint arXiv:2404.01866, 2024.

[28] Z. You, Z. Shi, H. Bo, J. Cartlidge, L. Zhang, and Y. Ge, "DGDNN: Decoupled graph diffusion neural network for stock movement prediction," arXiv preprint arXiv:2401.01846, 2024.

[29] Q. Zhang, Y. Zhang, X. Yao, S. Li, C. Zhang, and P. Liu, "A Dynamic Attributes-driven Graph Attention Network Modeling on Behavioral Finance for Stock Prediction," ACM Trans. Knowl. Discov. Data, vol. 18, no. 1, p. Article 16, 2023, doi: 10.1145/3611311.

[30] X. Hou, K. Wang, C. Zhong, and Z. Wei, "ST-Trader: A Spatial-Temporal Deep Neural Network for Modeling Stock Market Movement," IEEE/CAA Journal of Automatica Sinica, vol. 8, pp. 1015-1024, 05/01 2021, doi: 10.1109/JAS.2021.1003976.

[31] Kraken, "59% of Crypto Investors Use Dollar-Cost Averaging as Their Primary Investment Strategy," 2024. [Online]. Available: https://www.kraken.com/features/dollar-cost-averaging.

[32] S. Miao, M. Liu, and P. Li, "Interpretable and generalizable graph learning via stochastic attention mechanism," in International Conference on Machine Learning, 2022: PMLR, pp. 15524-15543, DOI: 10.48550/arXiv.2201.12987.

[33] Z. Liu, T.-K. Nguyen, and Y. Fang, "Tail-gnn: Tail-node graph neural networks," in Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 1109-1119, DOI: 10.1145/3447548.3467276.

[34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," arXiv preprint arXiv:1710.10903, 2017.

[35] P. N. Achyutha, S. Chaudhury, S. C. Bose, R. Kler, J. Surve, and K. Kaliyaperumal, "User Classification and Stock Market - Based Recommendation Engine Based on Machine Learning and Twitter Analysis," Mathematical Problems in Engineering, vol. 2022, no. 1, p. 4644855, 2022, DOI: 10.1155/2022/4644855.

[36] R. Feng, S. Jiang, X. Liang, and M. Xia, "STGAT: Spatial–Temporal Graph Attention Neural Network for Stock Prediction," Applied Sciences, vol. 15, no. 8, p. 4315, 2025, DOI: 10.3390/app15084315.

Multiclass Classification of Retinal and Optic Nerve Disorders Using Time-Domain Features from Band-Separated PERG Signals

Sathees Kumar Nataraj
Department of Mechatronics Engineering, University of Technology Bahrain Salmabad, Bahrain
E-mail: sknataraj@utb.edu.bh

Keywords: pattern electroretinogram (PERG), time-domain features, frequency band analysis, multiclass classification, AdaBoost, machine learning, neural networks

Received: June 11, 2025

Pattern Electroretinogram (PERG) signals offer critical insights into the functional status of retinal ganglion cells and are widely used for diagnosing optic nerve and retinal pathologies. While existing research has predominantly emphasized binary classification or full-spectrum signal analysis, limited studies have addressed the use of frequency band-based feature extraction in multiclass classification scenarios. This study introduces a novel approach that segments PERG signals into defined frequency bands and extracts a comprehensive set of time-domain features to support five-class classification. The dataset, originally comprising 15 diagnostic classes, was consolidated into five clinically relevant categories to ensure balanced representation. From each frequency band, more than 30 time-domain features were derived per eye, capturing key waveform characteristics. A consolidated dataset of five clinically relevant diagnostic classes have been used. Multiple machine learning models were evaluated, with AdaBoost-ECOC achieving the best performance: 99.11% ± 1.26 accuracy, 82.92% ± 0.59 precision, and 83.03% ± 0.55 F1-score. These results demonstrate the effectiveness and efficiency of frequency-based time-domain feature extraction for PERG signal classification.

Povzetek: Študija prikazuje, da je mogoče PERG signale uspešno razvrščati z uporabo značilk iz različnih frekvenčnih pasov.

1 Introduction

Ocular electrophysiology provides an objective method to evaluate visual function by measuring electrical responses from the retina and visual pathways in response to light or pattern stimuli. Core diagnostic tools such as flash electroretinography (ERG) and visual evoked potentials (VEP) have been instrumental in clinical evaluations [1]. Flash ERG measures photoreceptor (a-wave) and bipolar cell (b-wave) activity, while VEP evaluates cortical responses, thereby assisting in optic nerve assessment [2]–[5]. While imaging techniques such as optical coherence tomography (OCT) and visual field analysis provide structural insights, they often detect pathology only in advanced stages. Electrophysiological techniques, including multifocal ERG (mfERG) and multifocal VEP, enable earlier detection by identifying functional impairments preceding structural changes [6].

1.1 Background

Electrophysiological signals represent the measurable electrical activity of biological tissues, including neurons and sensory organs [7]. These signals can be captured using non-invasive skin electrodes, invasive probes, or multielectrode arrays [8]–[10]. In ophthalmology, ERG and its variant pattern electroretinography (PERG) are widely used [11].

Electrophysiological signals are the measurable electrical activities generated by biological cells and tissues, such as neurons, muscle fibers, and sensory organs [7][8]. These signals can be captured using non-invasive skin electrodes, invasive probes, or multielectrode arrays. In ophthalmology, ERG and its derivative, pattern electroretinography (PERG), are used to record retinal responses [9] [10]. PERG is particularly effective in detecting early dysfunction of retinal ganglion cells (RGCs), which is critical in glaucoma diagnosis. It employs contrast-reversing checkerboard stimuli focused on the macula, generating P50 and N95 waveforms that reflect outer retinal and RGC activity, respectively. [11]. Pattern electroretinography (PERG) stands out for its sensitivity in detecting early dysfunction of retinal ganglion cells (RGCs), particularly in glaucoma [12]. PERG utilizes contrast-reversing checkerboard patterns focused on the macula to generate responses characterized by P50 and N95 waveforms, which reflect outer retinal and RGC activity, respectively [13], [14]. Recordings are obtained using corneal electrodes or skin-mounted sensors, with patients maintaining fixation on a central target [15]. These signals, however, are sensitive to noise from eye movements and fixation errors. To address this, standardized acquisition protocols set by the International Society for Clinical Electrophysiology of Vision (ISCEV) are combined with modern signal processing techniques such as signal averaging and noise reduction [16], [17].

Pattern electroretinography (PERG) has become a critical tool for assessing retinal and optic nerve function, with computational preprocessing techniques playing a pivotal role in enhancing signal quality. Recent studies have demonstrated the effectiveness of multi-wavelet

transformations and deep learning architectures in isolating PERG components from noise caused by eye movements, electrical interference, and poor fixation [1], [18]. For example, research using Ricker, Gaussian, and Morlet wavelets achieved up to 91% classification accuracy for specific ERG protocols by creating 3D wavelet stacks that capture complementary signal features [19]. These technical advancements have enabled more reliable detection of pathological patterns across diverse conditions from common disorders like glaucoma and retinitis pigmentosa to rare entities such as birdshot chorioretinopathy and paracentral acute middle maculopathy [1].

In addition to differentiating macular diseases from optic nerve disorders, PERG has gained traction for early detection in inherited maculopathies [20], [21]. Canedo et al. (2023) proposed novel wavelet-based modeling approaches to improve PERG signal analysis [22]. Studies by Asanad and Karanjia [20] and Holder et al. [21] also highlight its clinical value. Although ISCEV standards (e.g., Bach et al. [16]) ensure consistency, their implementation in automated, multi-disease classification systems remains limited [23].

Most existing machine learning (ML) studies apply binary classification (e.g., glaucoma vs. healthy) using ERG or OCT data, with few utilizing PERG for multi-pathology classification [24]. Although wavelet-based features like Daubechies transforms have shown promise in glaucoma staging [25]. their application to broader PERG-based disease classification is limited. Furthermore, the impact of varying band-pass filter ranges (e.g., 1–100 Hz vs. 5–50 Hz) remains underexplored [26].

1.2 Proposed methodology

While existing research has established machine learning to diagnose individual retinal diseases, most focus on binary classification tasks using limited datasets [27], This study utilizes the IOBA-PERG dataset, a comprehensive repository of 1,354 transient pattern electroretinography (PERG) responses [1]. The study address broader diagnostic challenges through two key approaches:

a. Frequency specific signal segmentation

Advanced signal processing techniques are applied to perform a detailed analysis of waveform morphology, focusing on features such as N35–P50–N95 latencies, oscillatory potentials, and spectral characteristics. Drawing from methodologies used in EEG signal processing, PERG signals are divided into five clinically relevant frequency bands to enhance diagnostic interpretability as shown in Figure 1.

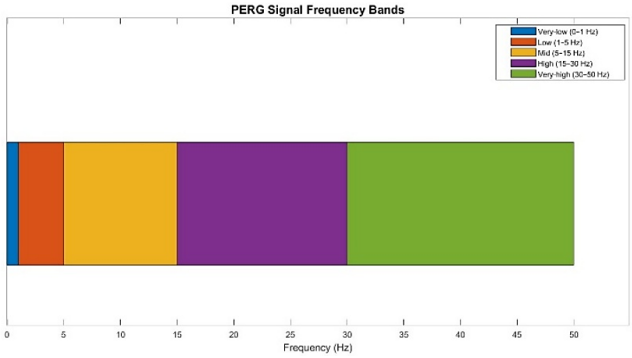


Figure 1: Block diagram of the proposed classification of retinal pathologies using machine learning

The signals are segmented into five clinically relevant frequency bands to enhance interpretability:

1. **Very-low (0–1 Hz):** Captures slow oscillatory components associated with metabolic retinal processes
2. **Low (1–5 Hz):** Isolates the N95 wave, indicative of inner retinal dysfunction
3. **Mid (5–15 Hz):** Emphasizes the P50 wave, representing macular function
4. **High (15–30 Hz):** Associated with transient glaucomatous responses
5. **Very-high (30–50 Hz):** May encode high-frequency oscillations in ischemic retinopathies

b. Multi-class classification framework

A frequency-optimized Machine Learning (ML) model [28] is developed to classify 15 retinal pathologies based on features from both time and frequency domains. The goal is to establish a machine learning-driven framework that incorporates band-optimized PERG features for the diagnosis of various retinal diseases, including early-stage glaucoma, diabetic macular edema, and retinitis pigmentosa. The study systematically evaluates the relevance of each frequency band, highlighting the advantages of frequency-domain over time-domain analysis.

1.3 Significance and contribution

Literature in this domain affirms the clinical relevance of PERG signals in assessing retinal function, yet significant gaps remain in analytical techniques used for classification and early detection. Most notably, while machine learning and artificial intelligence have been applied to ocular diagnostics, frequency-domain analysis of PERG signals remains underexplored. This research addresses that gap by investigating the diagnostic value of PERG frequency components through an ML framework trained on an open-access dataset from PhysioNet.

The proposed system, illustrated in Figure 2, employs machine learning algorithms to analyze the IOBA-PERG dataset and deliver multi-pathology classification outcomes. The implementation methodology and results are discussed in detail in the following sections of this article.

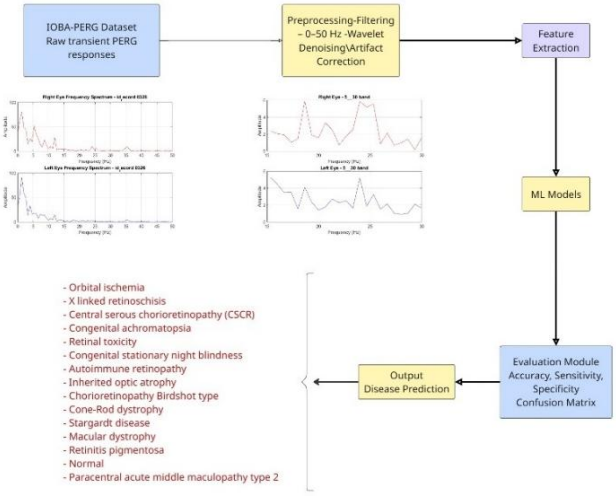


Figure 2: Block diagram of the proposed classification of retinal pathologies using machine learning

2 Methodology

This study aims to develop a robust multiclass classification framework for Pattern Electroretinogram (PERG) signals by leveraging frequency band-based feature extraction and supervised machine learning techniques. The research is driven by three core objectives:

1. to extract and evaluate meaningful time domain features from selected PERG frequency bands
2. to assess the performance of various machine learning classifiers, including neural networks and ensemble models, in accurately distinguishing five clinically relevant diagnostic categories, and
3. to explore the influence of class imbalance, feature dimensionality, and model interpretability in practical diagnostic scenarios.

These objectives provide a structured foundation for the study and guide the methodology described in the subsequent sections, including data collection, signal processing, classification, and performance evaluation.

2.1 Dataset description

In this research, the publicly available IOBA-PERG dataset from PhysioNet [1] have been used, which comprises 1,354 transient pattern electroretinogram (PERG) responses recorded from 304 individuals, encompassing a total of 336 independent session records. Each record contains bilateral retinal signals sampled at a rate of 170 Hz, alongside extensive clinical metadata including age, sex, and multiple diagnostic labels (diagnosis1, diagnosis2, diagnosis3). In this analysis, only the primary diagnostic field (diagnosis1) was used to define the ground truth labels for supervised multi-class classification system.

Although the complete dataset includes a broad range of retinal and optic nerve pathologies, a curated subset consisting of 15 clinically significant disease categories, including healthy controls, have been chosen. This selection was primarily motivated by the need to balance class representation and mitigate the adverse effects of sample size disparities. Categories with fewer than five recordings were excluded to prevent data imbalance, which can compromise classifier performance. Future work will aim to extend the analysis to include underrepresented classes once the present methodology has been validated.

The final dataset used for classification included 280 records, stratified across 15 diagnostic categories. Table 1 presents a summary of the most prevalent pathologies retained for analysis, along with brief clinical descriptions.

Table 1: Summary of Selected Diagnoses from the IOBA-PERG Dataset

Sl. No.	Name of the Disease	No. of Records	Explanation
1	Normal	106	No clinical signs of retinal dysfunction.
2	Retinitis Pigmentosa	47	Genetic retinal dystrophy affecting photoreceptor response.
3	Macular Dystrophy	33	Degenerative or inherited disorder of the macula.
4	Stargardt Disease	16	Juvenile macular degeneration caused by ABCA4 mutations.
5	Cone-Rod Dystrophy	14	Affects cone and rod photoreceptors, leading to progressive vision loss.
6	Birdshot Chorioretinopathy	12	Rare bilateral posterior uveitis characterized by creamy lesions.
7	Inherited Optic Atrophy	10	Genetically mediated degeneration of retinal ganglion cells.
8	Congenital Stationary Night Blindness	8	Non-progressive dysfunction of rod photoreceptors.
9	Retinal Toxicity	7	Damage induced by systemic or intravitreal medications.
10–15	Other Rare Disorders*	27	Autoimmune retinopathy, ischemia, achromatopsia, etc.
	Total	280	

*Includes: autoimmune retinopathy, orbital ischemia, X-linked retinoschisis, central serous chorioretinopathy, congenital achromatopsia, and paracentral acute middle maculopathy.

2.2 Preprocessing of PERG signals

All PERG recordings were initially subjected to preprocessing to enhance signal fidelity and minimize noise. The signals, acquired at 170 Hz, capture the dynamic activity of retinal cells within the relevant electrophysiological frequency range (0.5–50 Hz). Each session provided both left and right eye recordings. The raw signals were depicted in Figure 3 and Figure 4, represents a sample normal PERG signal and Macular Dystrophy signal accordingly.

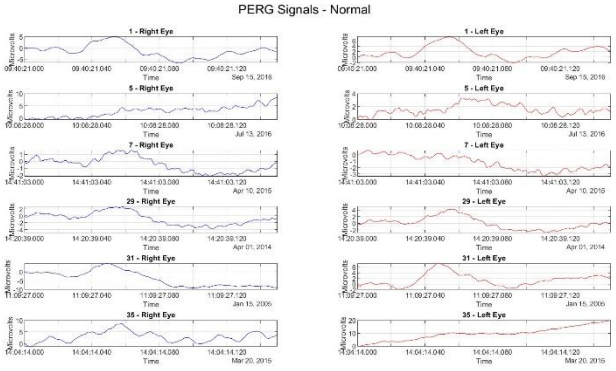


Figure 3: Raw PERG signal (Normal Subject)

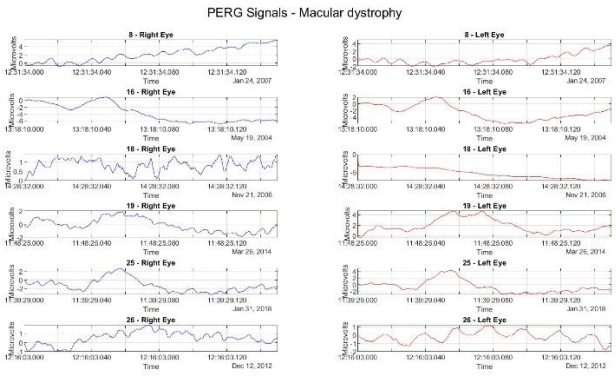


Figure 4: Raw PERG signal (Macular Dystrophy)

In addition, raw signals were filtered using a fourth-order zero-phase Butterworth bandpass filter (0.5–50 Hz). This choice was guided by standard electrophysiological analysis practices [2], effectively suppressing baseline drift and high-frequency electrical artifacts without distorting phase information. The zero-phase implementation (via MATLAB's filtfilt function) ensures that temporal relationships between signal components are preserved [29]. Post-filtering, spectral content was examined through Fast Fourier Transform (FFT) to assess signal quality and identify dominant frequency components [30]. This preprocessing step served both to denoise the data and to prepare it for subsequent band-specific decomposition and feature extraction. Figure 5 presents a sample raw PERG signal, filtered and with FFT applied, the frequency components in the 0–50 Hz range.

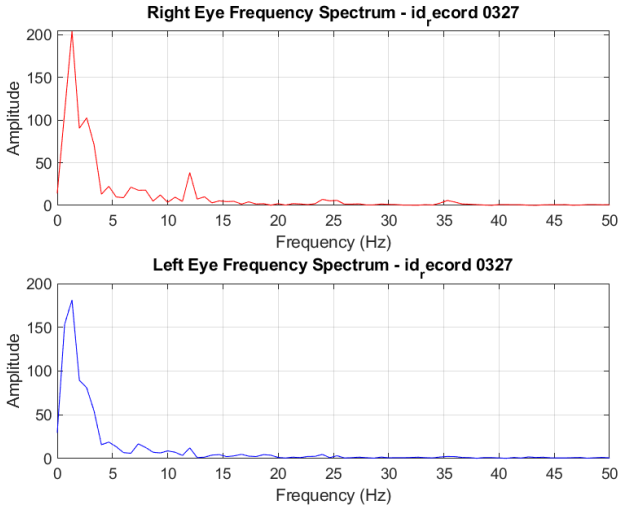


Figure 5: Frequency Spectrum of a raw PERG signal (0–50 Hz)

2.3 Frequency band decomposition

Subsequently, to limit and segregate physiologically meaningful signal components, the filtered PERG signals were decomposed into five distinct frequency bands, each reflecting different retinal or neurophysiological phenomena [25]. Table 2 list the proposed frequency band limits and its functional relevance.

Table 2: Frequency band range (Hz) - functional relevance

Band Label	Frequency Range (Hz)	Interpretation
Very low	0 - 1	Baseline fluctuations, non-neural drift
Low	1 - 5	Principal PERG components (e.g., N35, P50, N95)
Mid	5 - 15	Flicker responses, ganglion cell activity
High	15 - 30	Oscillatory potentials, secondary components
Very high	30 - 50	Noise-prone region, possible residual oscillations

Frequency spectra were analyzed for both eyes independently, and magnitude responses were retained for subsequent feature extraction. Figure 6–10: depicts the Band-specific frequency plots for both eyes.

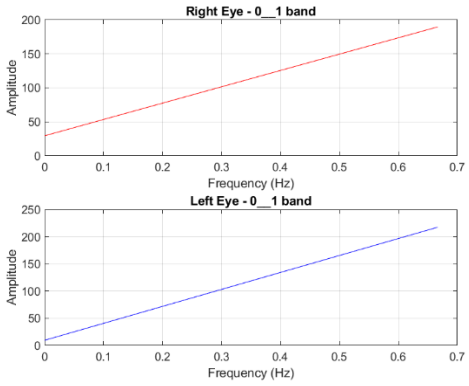


Figure 6: Frequency plot - (0 to 1 Hz) Right and Left eye

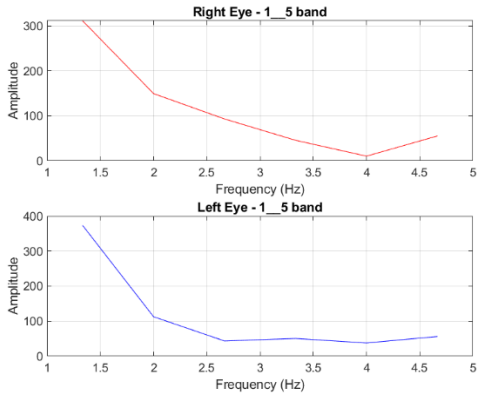


Figure 7: Frequency plot - (1 to 5 Hz) Right and Left eye

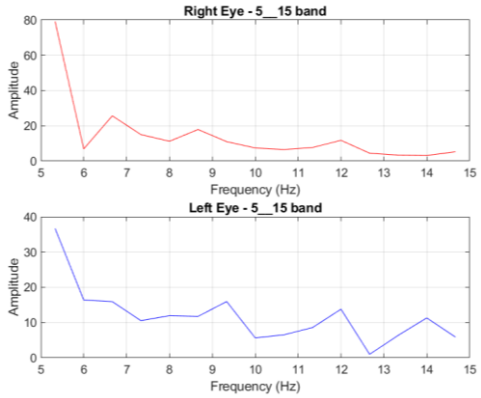


Figure 8: Frequency plot - (5 to 15 Hz) Right and Left eye

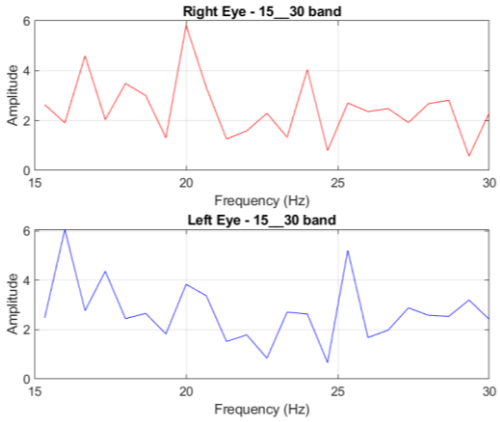


Figure 9: Frequency plot - (15 to 30 Hz) Right and Left eye

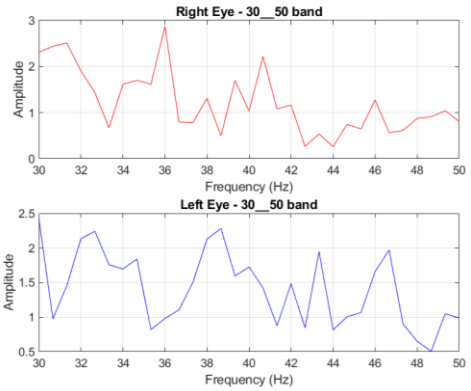


Figure 10: Frequency plot (15 to 30 Hz) Right and Left eye

This band-based separation enables detailed feature analysis across different physiological domains.

2.4 Time-domaing feature extraction

A comprehensive set of time-domain features was computed across each frequency band and for each eye separately. These features were selected to capture the statistical characteristics of the PERG waveforms. In total, over 11 scalar features were extracted per band per eye, including both traditional metrics and signal-processing-derived descriptors. Table 2 summarizes the key features used in this study.

Although the signals were filtered and segmented by frequency bands, the analysis in this work is limited to time-domain features. An initial comparative assessment considered features from time, frequency, and time-frequency domains. Among these, time-domain features demonstrated the most consistent and informative patterns, offering valuable insight into waveform behavior across different conditions. Based on these findings, the current analysis focuses exclusively on time-domain characteristics to provide a clear and interpretable foundation [31], [32].

Additional analysis involving frequency and time-frequency domain features will be presented in a subsequent study to expand the scope and diagnostic capabilities.

Table 3: Summary of extracted time-domain features

Feature	Description / Formula
Mean (μ)	Arithmetic average of signal amplitudes
Variance (σ^2)	Measure of amplitude dispersion
Standard deviation (σ)	variation of a amplitude dispersion
Frequency Variance (μ_f)	power-weighted variance of the frequency components
Root Mean Square (RMS)	Energy-based feature sensitive to amplitude
Peak-to-Peak	Difference between maximum and minimum values
Skewness	Degree of asymmetry in amplitude distribution
Kurtosis	Peakedness or flatness of amplitude distribution
Slope	Mean first-order derivative
Mean Absolute Value	Average of absolute signal values
Area Under Curve (AUC)	Integral of the waveform via trapezoidal approximation

The time-domain features used in this study were chosen to capture a broad spectrum of signal characteristics relevant to physiological interpretation. The mean and variance represent basic statistical measures, reflecting the average amplitude and the overall variability of the PERG signal. Root Mean Square (RMS) offers a measure of signal power, emphasizing higher amplitudes, while peak-to-peak captures the full dynamic range of the waveform by calculating the difference between its maximum and minimum values. To assess the shape and distribution of the signal, skewness is used to measure asymmetry, and kurtosis helps identify sharp peaks or flatness in the waveform. The slope feature captures the rate of change, which is important for detecting transitions or trends in signal morphology. Measures like mean absolute value (MAV) and area under the curve (AUC) summarize the magnitude and energy content of the signal.

Thus, Features were extracted separately for each eye across three frequency bands: B2 (1–5 Hz), B3 (5–15 Hz), and B4 (15–30 Hz), with 11 features per band. Bands B1 (0–1 Hz) and B5 (30–50 Hz) were excluded due to baseline drift and noise, respectively. This yielded 33 features per eye and 66 features per subject. The final dataset consisted of 280 subject recordings, each represented by a 66-dimensional feature vector.

2.5 Feature classification

The feature dataset used in this study was derived from Pattern Electroretinogram (PERG) signals recorded from both the left and right eyes of human subjects. Each subject's data included a variety of time-domain and frequency-domain features extracted from three predefined frequency bands: 1–5 Hz, 5–15 Hz, and 15–30 Hz.

For each subject, a total of 66 features were extracted. These included statistical metrics. An additional class label was assigned to each feature vector. The features were organized into a consolidated table, with dimensional alignment ensured by padding missing entries with NaN values. Each row represented a unique trial per subject, identified by a generated SubjectID.

The original dataset contained 15 distinct class labels corresponding to various clinical or experimental conditions. To address class imbalance and enhance classification robustness, these were regrouped into five broader categories:

Class 1: Normal signals (106 records)

Class 2: Retinitis Pigmentosa (47 records)

Class 3: Macular Dystrophy (33 records)

Class 4: Stargardt Disease (16 records)

Class 5: Combined group from original Classes 5–15, representing general or baseline cases

This relabeling strategy reduced the impact of underrepresented classes and improved classifier stability across trials. The final class distribution was verified to ensure balanced representation.

2.6 Data normalization

Prior to training, all feature values were normalized using Min-Max scaling (range normalization) to ensure uniformity in feature contribution and accelerate model convergence. Normalization was performed after class filtering but prior to data splitting, using MATLAB's normalize function with the 'range' method.

2.7 Classification models

This study evaluated the discriminative potential of the extracted features using five supervised machine learning classifiers under a consistent experimental protocol. The models included a Neural Network (PatternNet) with two hidden layers (15 and 10 neurons), trained using the Levenberg–Marquardt backpropagation algorithm and softmax output for multiclass prediction via one-hot encoding. Ensemble methods included Random Forest (100 trees with out-of-bag prediction) and Bagged Trees. To address class imbalance, the RUSBoost algorithm was employed with 100 decision tree learners and random undersampling of majority classes. The final model, AdaBoost [33], was implemented within an Error-Correcting Output Codes (ECOC) framework using decision trees (max split of 20) and a one-vs-all coding scheme.

All classifiers were trained and tested across 10 independent trials using an 80–20 train-test split. Performance was evaluated using standard metrics: accuracy, precision, recall, specificity, F1-score, training time, and model complexity (measured via trainable parameters, number of learners, or ECOC structure, depending on the model). Classifier interpretability was also assessed qualitatively. Confusion matrices were saved in both graphical and CSV formats to support class-wise error analysis. This structured setup enabled a rigorous comparison of neural and ensemble classifiers, offering insights into their relative accuracy, generalization, and applicability to multiclass PERG signal classification.

3 Results and discussion

The original dataset included 15 distinct class labels, which are listed in Table 4 along with their percentage distributions. These classes were later reorganized and grouped into five broader categories.

Table 4: Retinal pathology class distribution before merging

Value	Count	Percent (%)	Value	Count	Percent (%)	Value	Count	Percent (%)
1	5	1.79	6	7	2.50	11	14	5.00
2	16	5.71	7	8	2.86	12	33	11.79
3	5	1.79	8	8	2.86	13	47	16.79
4	5	1.79	9	10	3.57	14	106	37.86
5	3	1.07	10	12	4.29	15	1	0.36

Initial analysis was carried out using all 15 classes. Based on the results, the highest classification accuracy was 89.29%, achieved using the AdaBoost classifier. The associated performance metrics included a precision of 67.52 ± 6.18, recall of 68.83 ± 6.58, specificity of 99.23 ± 0.35, F1-score of 68.14 ± 6.20, and a training time of 0.13 ± 0.08 seconds. Other classifiers did not perform as well across these metrics. The analysis also revealed limitations in the dataset, particularly the low and uneven sample distribution across classes, one of which contained only a single sample. Detailed results from this 15-class classification are presented in Table 5.

Table 5: Performance comparison of the five classifiers used for the retinal pathology classification (before merging)

Trial	Accuracy			Precision			Recall			Specificity			F1			Training Time		
NN-LM	51.43	±	10.61	20.76	±	9.61	23.91	±	7.24	96	±	0.83	21.79	±	7.77	3.95	±	2.47
Random Forest	58.75	±	5.8	14.27	±	3.31	18.2	±	3.31	96.63	±	0.32	15.96	±	3.23	0.44	±	0.11
Bagged Trees	60	±	8.08	16.94	±	3.32	20.13	±	3.65	96.73	±	0.56	18.31	±	3.26	0.64	±	0.12
RUSBoost	17.5	±	21.31	16.89	±	20.68	18.25	±	19.72	94.11	±	1.31	16.86	±	20.56	1.02	±	0.4
AdaBoost	89.29	±	5.12	67.52	±	6.18	68.83	±	6.58	99.23	±	0.35	68.14	±	6.2	0.13	±	0.08

The revised five-class distribution, along with the number of samples and their respective percentages, is shown in Table 6.

Table 6: Retinal pathology class distribution after merging

Value	Count	Percent (%)
1	78	27.86
2	16	5.71
3	33	11.79
4	47	16.79
5	106	37.86

The performance of each classifier is presented from Table 7 to Table 11. The table 7 presents the results of 10 trials using the Multilayer Neural Network with Levenberg–Marquardt (MLNN-LM) algorithm for five-class classification. All metrics—accuracy, precision, recall, specificity, and F1-score—are reported in percentage terms. The model achieved high performance, with perfect scores in Trials 1, 7, and 9. An average accuracy of 96.43% ± 3.04 indicates strong predictive ability and reasonable stability. However, higher variability in precision (±9.56%) and recall (±8.79%), especially in Trial 6, points to challenges with class imbalance. The average F1-score of 93.12% ± 8.85 reflects a good balance between precision and recall. Training time ranged from 3.43 to 7.58 seconds, averaging 4.61 ± 1.19 seconds.

Table 7: Performance evaluation of retinal pathology classification using Multi layer neural

Trial	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 (%)	Training Time (seconds)
1	100	100	100	100	100	4.83
2	94.64	94.95	87.00	98.64	90.80	3.50
3	94.64	97.15	88.18	98.34	92.45	5.10
4	94.64	96.82	94.44	98.45	95.62	4.42
5	98.21	99.13	98.33	99.41	98.73	4.39
6	94.64	68.18	73.33	98.81	70.66	4.23
7	100	100	100	100	100	4.80
8	96.43	96.85	90.83	99.09	93.74	7.58
9	100	100	100	100	100	3.43
10	91.07	95.61	83.61	97.11	89.21	3.81
Average	96.43 ± 3.04	94.87 ± 9.56	91.57 ± 8.79	98.99 ± 0.92	93.12 ± 8.85	4.61 ± 1.19

Overall, the classifier is effective but sensitive to class distribution, suggesting the need for data balancing in future work.

Table 8: Performance evaluation of retinal pathology classification using Random Forest

Trial	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 (%)	Training Time (seconds)	Model Complexity
1	83.93	52.78	54.71	96.67	53.73	0.31	100
2	91.07	75.17	68.75	98.22	71.82	0.32	100
3	89.29	71.04	63.76	97.65	67.21	0.28	100
4	94.64	80.00	77.78	98.86	78.87	0.27	100
5	91.07	71.15	70.56	98.21	70.85	0.30	100
6	96.43	60.98	66.67	99.32	63.70	0.29	100
7	87.50	70.93	62.46	97.55	66.42	0.27	100
8	89.29	58.77	62.50	97.73	60.58	0.27	100
9	94.64	61.18	66.67	98.86	63.81	0.29	100
10	76.79	56.61	52.28	95.16	54.36	0.29	100
Average	89.46 ± 5.8	65.86 ± 8.94	64.61 ± 7.41	97.82 ± 1.21	65.13 ± 7.77	0.29 ± 0.02	100 ± 0

The table 8 summarizes the performance of the Random Forest (Tree Bagger) model across 10 trials for the five-class classification task. All evaluation metrics are expressed in percentages. The model achieved a maximum accuracy of 96.43% and a minimum of 76.79%, with an average of 89.46% ± 5.8, indicating moderate classification performance with some variability. Precision and recall showed noticeable fluctuations (65.86% ± 8.94 and 64.61% ± 7.41, respectively), highlighting occasional difficulty in consistently identifying positive cases. The F1-score averaged 65.13% ± 7.77, reflecting modest balance between precision and recall. Specificity remained high and consistent at 97.82% ± 1.21, confirming reliable identification of negative instances. Training time was efficient and consistent, averaging 0.29 ± 0.02 seconds, with a fixed model complexity. Overall, the classifier performed reasonably well but showed sensitivity to data imbalance and class variation.

Table 9: Performance evaluation of retinal pathology classification using Bagged Trees (Bootstrap Aggregation)

Trial	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 (%)	Training Time (seconds)
1	89.29	58.25	59.93	97.71	59.08	0.65
2	92.86	76.79	72.92	98.42	74.80	0.69
3	91.07	73.28	69.32	98.17	71.24	0.77
4	92.86	79.17	75.93	98.48	77.51	0.75
5	92.86	73.89	73.89	98.59	73.89	0.66
6	94.64	58.33	65.00	99.01	61.49	0.70
7	87.50	54.40	57.86	97.53	56.07	0.70
8	92.86	78.61	70.14	98.41	74.13	0.69
9	94.64	63.43	66.67	98.67	65.01	0.58
10	87.50	71.03	65.97	97.36	68.41	0.63
Average	91.61 ± 2.67	68.72 ± 9.28	67.76 ± 5.87	98.24 ± 0.54	68.16 ± 7.4	0.68 ± 0.06

Table 9 presents results from 10 trials using the Bagged Trees (Bootstrap Aggregation) model for five-class classification. Performance metrics are expressed as percentages. The model showed consistent accuracy, ranging from 87.50% to 94.64%, with an average of 91.61% ± 2.67, indicating good stability. Precision and recall averaged 68.72% ± 9.28 and 67.76% ± 5.87, respectively, suggesting moderate ability to detect positive instances with some variation across trials. The F1-score averaged 68.16% ± 7.4, reflecting reasonable balance between precision and recall. Specificity remained high at 98.24% ± 0.54, confirming consistent recognition of negative cases. Training times were efficient, averaging 0.68 ± 0.06 seconds, with constant model complexity. Overall, the classifier performed reliably, offering a balanced trade-off between accuracy and computational cost.

Table 10: Performance evaluation of retinal pathoalogy classification using RUS Boost

Trial	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 (%)	Training Time (seconds)
1	21.43	15.91	13.90	83.62	14.84	0.54
2	37.50	6.25	16.67	83.33	9.09	0.02
3	30.36	5.06	16.67	83.33	7.76	0.02
4	42.86	25.00	26.74	87.88	25.84	0.50
5	32.14	28.68	25.51	86.51	27.00	0.62
6	48.21	29.39	26.06	89.17	27.63	0.50
7	42.86	27.64	27.59	88.29	27.61	0.49
8	37.50	6.25	16.67	83.33	9.09	0.02
9	35.71	27.92	23.31	87.08	25.41	0.45
10	33.93	18.44	18.47	86.11	18.46	0.53
Average	36.25 ± 7.53	19.05 ± 10.12	21.16 ± 5.17	85.87 ± 2.29	19.27 ± 8.43	0.37 ± 0.24

Table 10 shows the results of 10 trials using the RUSBoost classifier for five-class classification. Compared to other models, RUSBoost delivered substantially lower performance, with an average accuracy of 36.25% ± 7.53. Precision and recall were also low, averaging 19.05% ± 10.12 and 21.16% ± 5.17, respectively, reflecting poor identification of true positives. The F1-score averaged 19.27% ± 8.43, indicating weak overall classification quality. Although specificity remained relatively stable at 85.87% ± 2.29, this reflects the model's tendency to favor the majority (negative) classes. Training time averaged 0.37 ± 0.24 seconds, with model complexity fluctuating (average 70 ± 48.3). These results suggest that RUSBoost struggled with the dataset, likely due to severe class imbalance and insufficient representative samples for minority classes.

Table 11 summarizes the performance of the AdaBoost classifier implemented via the ECOC (Error-Correcting Output Codes) framework. The model showed exceptionally high and consistent results, achieving an average accuracy of 99.11% ± 1.26 across 10 trials. Precision, recall, and F1-score were all highly stable, averaging 82.92% ± 0.59, 83.15% ± 0.59, and 83.03% ± 0.55, respectively. Specificity was nearly perfect at 99.77% ± 0.32, indicating excellent identification of negative cases. Training time remained minimal (0.05 ± 0.01 seconds) and model complexity low (5.7 ± 0.48), highlighting the efficiency of this approach. Overall,

AdaBoost-ECOC delivered the best balance of performance, consistency, and computational efficiency among all classifiers tested.

Table 11: Performance evaluation of retinal pathoalogy classification using AdaBoost (via ECOC Framework)

Trial	Accuracy (%)	Precision (%)	Recall (%)	Specificity (%)	F1 (%)	Training Time (seconds)
1	100	83.33	83.33	100	83.33	0.05
2	98.21	82.58	83.33	99.52	82.95	0.04
3	98.21	82.41	83.33	99.57	82.87	0.05
4	100	83.33	83.33	100	83.33	0.05
5	100	83.33	83.33	100	83.33	0.06
6	100	83.33	83.33	100	83.33	0.05
7	96.43	81.67	81.48	99.12	81.57	0.04
8	98.21	82.58	83.33	99.52	82.95	0.03
9	100	83.33	83.33	100	83.33	0.05
10	100	83.33	83.33	100	83.33	0.05
Average	99.11 ± 1.26	82.92 ± 0.59	83.15 ± 0.59	99.77 ± 0.32	83.03 ± 0.55	0.05 ± 0.01

Table 12: Performance comparison of the five classifiers used for the retinal pathology classifications

Classifier	Accuracy			Precision			Recall			Specificity			F1			Training Time		
NN-LM	96.43	±	3	94.87	±	9.56	91.57	±	8.8	98.99	±	0.9	93.12	±	8.9	4.6	±	1.2
Random Forest	89.46	±	5.80	65.86	±	8.94	64.61	±	7.41	97.82	±	1.21	65.13	±	7.77	0.29	±	0.02
Bagged Trees	91.61	±	2.67	68.72	±	9.28	67.76	±	5.87	98.24	±	0.54	68.16	±	7.40	0.68	±	0.06
RUSBoost	36.25	±	7.53	19.05	±	10.12	21.16	±	5.17	85.87	±	2.29	19.27	±	8.43	0.37	±	0.24
AdaBoost	99.11	±	1.26	82.92	±	0.59	83.15	±	0.59	99.77	±	0.32	83.03	±	0.55	0.05	±	0.01

Table 12 presents a comparative analysis of five classification models evaluated over 10 independent trials using key performance metrics: accuracy, precision, recall, specificity, F1-score, and training time.

The AdaBoost (ECOC) model outperformed all other classifiers, achieving the highest average accuracy (99.11% ± 1.26) and exceptional consistency across all metrics. Its precision, recall, and F1-score remained tightly clustered around 83%, with minimal variability, while requiring the least training time (0.05 ± 0.01 seconds) and low model complexity, making it both highly effective and computationally efficient.

The Neural Network with Levenberg–Marquardt (NN-LM) also delivered strong performance, with an average accuracy of 96.43% ± 3.04 and the highest precision (94.87% ± 9.56) among all models. However, its training time was significantly higher (4.61 ± 1.19 seconds), indicating greater computational cost.

Bagged Trees provided a good balance, achieving 91.61% ± 2.67 accuracy with moderate precision and recall values around 68%, and reasonable training efficiency. In contrast, Random Forest (Tree Bagger) showed slightly lower accuracy (89.46% ± 5.80) and higher variability across precision and recall, reflecting less stable behavior across trials.

The RUSBoost classifier, which incorporates random under sampling to address class imbalance, underperformed significantly in all metrics, with accuracy at 36.25% ± 7.53 and an F1-score of just 19.27% ± 8.43. While the intention of addressing imbalance was valid, the aggressive under sampling likely led to loss of information, reducing classification quality.

Overall, AdaBoost via ECOC emerged as the most robust and efficient model, followed by NN-LM for accuracy-focused scenarios and Bagged Trees for balance between performance and training time,. RUSBoost was

not suitable in this context due to its consistently poor results.

The confusion matrices of the AdaBoost and MLNN-LM classifiers corresponding to their lowest classification accuracy trials were analyzed to identify specific patterns of misclassification. These are presented in Figure 11 and Figure 12, respectively. From the AdaBoost confusion matrix, it is evident that the class formed by merging original classes 5–15 exhibited a misclassification rate of 11.1%, corresponding to 2 incorrectly predicted samples. In the case of the MLNN-LM classifier, Class 2 (Retinitis Pigmentosa) showed a misclassification rate of 33.3% (1 out of 3 samples), while Class 3 (Macular Dystrophy) had a misclassification rate of 11.1% (1 out of 9 samples). These results highlight specific weaknesses in class-level prediction, particularly for classes with fewer training samples.

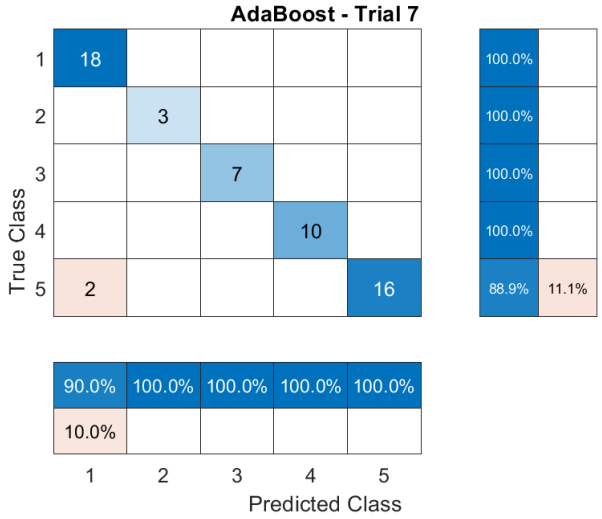


Figure 11: Confusion Matrix of AdaBoost via ECOC with Minimum Accuracy of 96.43%

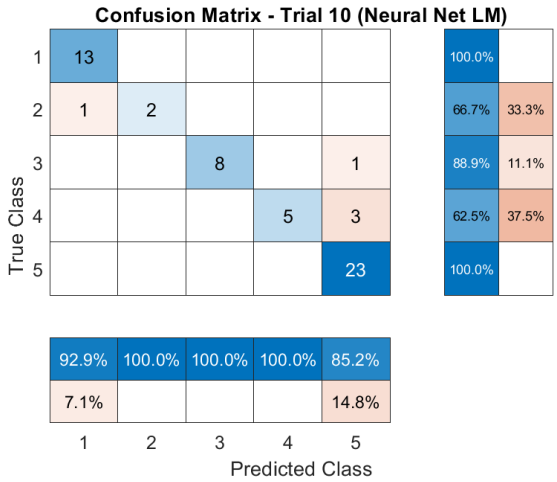


Figure 12: Confusion Matrix of MLNN-LM with Minimum Accuracy of 91.07%

4 Conclusion

This study presented a five-class classification framework for distinguishing retinal and optic nerve pathologies using time-domain features extracted from band-separated PERG signals. Among the various models evaluated, AdaBoost via the ECOC framework demonstrated the highest and most consistent classification performance, while the neural network using the Levenberg–Marquardt algorithm also achieved high accuracy with strong precision. The comparative evaluation revealed the strengths and limitations of each classifier, particularly in handling class imbalance and small sample sizes.

Despite challenges posed by uneven class distributions and limited data for certain categories, the results highlight the feasibility and promise of the proposed approach. The time-domain analysis alone has produced positive outcomes, establishing a strong foundation for further exploration.

As this work represents an initial phase, future research will build upon these findings by incorporating more advanced and focused analyses. Specifically, feature extraction will be refined by using selected frequency bands rather than the full spectrum, enabling targeted comparisons through the use of intelligent techniques and optimization algorithms. In addition to time-domain analysis, future studies will introduce frequency-domain and time-frequency domain features to improve model discrimination. Another planned direction involves the segmentation and analysis of specific waveform components—namely the N35, P50, and N95 waves—on a per-record basis, with the goal of enhancing physiological interpretability and improving classification performance.

References

[1] I. Fernández, R. Cuadrado-Asensio, Y. Larriba, C. Rueda, and R. M. Coco-Martín, ‘A comprehensive dataset of pattern electroretinograms for ocular electrophysiology research’, *Sci. Data*, vol. 11, no. 1, p. 1013, 2024. https://doi.org/10.1038/s41597-024-03857-1

[2] J. V. Odom *et al.*, ‘ISCEV standard for clinical visual evoked potentials (2009 update)’, *Doc. Ophthalmol.*, vol. 120, pp. 111–119, 2010. https://doi.org/10.1007/s10633-009-9195-4

[3] D. L. McCulloch *et al.*, ‘ISCEV Standard for full-field clinical electroretinography (2015 update)’, *Doc. Ophthalmol.*, vol. 130, no. 1, pp. 1–12, 2015. https://doi.org/10.1007/s10633-015-9504-z

[4] S. K. S. K. Nataraj *et al.*, ‘Statistical cross-correlation band features based thought controlled communication system’, *AI Commun.*, vol. 8, no. 4, pp. 20–28, Aug. 2016, doi: 10.3233/AIC-160703. https://doi.org/10.3233/aic-160703

[5] R. Hamilton *et al.*, ‘VEP estimation of visual

acuity: a systematic review', *Doc. Ophthalmol.*, vol. 142, pp. 25–74, 2021. https://doi.org/10.1007/s10633-020-09770-3

[6] D. C. Hood *et al.*, 'ISCEV standard for clinical multifocal electroretinography (mfERG)(2011 edition)', *Doc. Ophthalmol.*, vol. 124, pp. 1–13, 2012. https://doi.org/10.1007/s10633-011-9296-8

[7] J. Bures, M. Petran, and J. Zachar, *Electrophysiological methods in biological research*. Academic Press, 2013. https://doi.org/10.1016/c2013-0-10442-4

[8] R. N. Weinreb and P. T. Khaw, 'Primary open-angle glaucoma', *Lancet*, vol. 363, no. 9422, pp. 1711–1720, 2004. https://doi.org/10.3109/9780203007204-15

[9] N. V Thakor, 'Biopotentials and electrophysiology measurement', *Meas. Instrumentation, Sensors Handb.*, vol. 74, 1999. https://doi.org/10.1201/9780415876179.ch74

[10] J. Rettinger, S. Schwarz, and W. Schwarz, *Electrophysiology*. Springer, 2022. https://doi.org/10.1007/978-3-030-86482-8_3

[11] D. A. Thompson, M. Bach, J. J. McAnany, M. Šuštar Habjan, S. Viswanathan, and A. G. Robson, 'ISCEV standard for clinical pattern electroretinography (2024 update)', *Doc. Ophthalmol.*, vol. 148, no. 2, pp. 75–85, 2024.

[12] A. B. Suttle *et al.*, 'Assessment of the drug interaction potential and single-and repeat-dose pharmacokinetics of the BRAF inhibitor dabrafenib', *J. Clin. Pharmacol.*, vol. 55, no. 4, pp. 392–400, 2015. https://doi.org/10.1007/s10633-024-09970-1

[13] G. A. Omura, 'Progress in gynecologic cancer research: the Gynecologic Oncology Group experience', in *Seminars in oncology*, 2008, vol. 35, no. 5, pp. 507–521. https://doi.org/10.1053/j.seminoncol.2008.07.007

[14] V. Parisi, 'Electrophysiological assessment of glaucomatous visual dysfunction during treatment with cytidine-5′-diphosphocholine (citicoline): a study of 8 years of follow-up', *Doc. Ophthalmol.*, vol. 110, pp. 91–102, 2005. https://doi.org/10.1007/s10633-005-7348-7

[15] G. E. Holder, 'Pattern electroretinography (PERG) and an integrated approach to visual pathway diagnosis', *Prog. Retin. Eye Res.*, vol. 20, no. 4, pp. 531–561, 2001. https://doi.org/10.1016/s1350-9462(00)00030-6

[16] M. Bach and M. B. Hoffmann, 'Update on the pattern electroretinogram in glaucoma', *Optom. Vis. Sci.*, vol. 85, no. 6, pp. 386–395, 2008. https://doi.org/10.1097/opx.0b013e318177ebf3

[17] A. G. Robson *et al.*, 'ISCEV Standard for full-field clinical electroretinography (2022 update)', *Doc. Ophthalmol.*, vol. 144, no. 3, pp. 165–177, 2022. https://doi.org/10.1007/s10633-022-09872-0

[18] A. L. Goldberger *et al.*, 'PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals', *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. https://doi.org/10.1161/01.cir.101.23.e215

[19] M. Kulyabin, A. Zhdanov, A. Dolganov, M. Ronkin, V. Borisov, and A. Maier, 'Enhancing electroretinogram classification with multi-wavelet analysis and visual transformer', *Sensors*, vol. 23, no. 21, p. 8727, 2023. https://doi.org/10.3390/s23218727

[20] S. Asanad and R. Karanjia, 'Multifocal electroretinogram', in *StatPearls [Internet]*, StatPearls Publishing, 2022. https://doi.org/10.5005/jp/books/10550_15

[21] G. E. Holder, A. G. Robson, C. R. Hogg, M. Kurz-Levin, N. Lois, and A. C. Bird, 'Pattern ERG: clinical overview, and some observations on associated fundus autofluorescence imaging in inherited maculopathy', *Doc. Ophthalmol.*, vol. 106, no. 1, p. 17, 2003. https://doi.org/10.1023/a:1022471623467

[22] C. Canedo, I. Fernández, R. M. Coco, R. Cuadrado, and C. Rueda, 'Novel modeling proposals for the analysis of pattern electroretinogram signals', in *Statistical Methods at the Forefront of Biomedical Advances*, Springer, 2023, pp. 255–273. https://doi.org/10.1007/978-3-031-32729-2_11

[23] P. A. Constable, M. Bach, L. J. Frishman, B. G. Jeffrey, A. G. Robson, and I. S. for C. E. of V. http://www. iscev. org, 'ISCEV Standard for clinical electro-oculography (2017 update)', *Doc. Ophthalmol.*, vol. 134, pp. 1–9, 2017. https://doi.org/10.1007/s10633-017-9573-2

[24] E. J. Ahn, Y. I. Shin, Y. K. Kim, J. W. Jeoung, and K. H. Park, 'Hemifield-based analysis of pattern electroretinography in normal subjects and patients with preperimetric glaucoma', *Sci. Rep.*, vol. 14, no. 1, p. 5116, 2024. https://doi.org/10.1038/s41598-024-55601-9

[25] M. Bach *et al.*, 'ISCEV standard for clinical pattern electroretinography (PERG): 2012 update', *Doc. Ophthalmol.*, vol. 126, pp. 1–7, 2013. https://doi.org/10.1007/s10633-012-9353-y

[26] A. Aykut, B. Akgün, A. S. Sezenöz, M. O. Sevik, and Ö. Şahin, 'Diagnosing retinal disorders with artificial intelligence: the role of large language models in interpreting pattern electroretinography data', *J. Heal. Sci. Med.*, vol. 7, no. 5, pp. 538–542, 2024. https://doi.org/10.32322/jhsm.1506378

[27] M. Kulyabin *et al.*, 'Synthetic Electroretinogram Signal Generation Using Conditional Generative Adversarial Network for Enhancing Classification of Autism Spectrum Disorder', *arXiv Prepr. arXiv2407.08166*, 2024. https://doi.org/10.1007/s10633-025-10019-0

[28] Q. Li and J. Zhou, ‘A Comparative Analysis of Extreme Gradient Boosting, Decision Tree, Support Vector Machines, and Random Forest Algorithm in Data Analysis of College Students’ Psychological Health’, *Informatica*, vol. 49, no. 15, 2025. https://doi.org/10.31449/inf.v49i15.7004

[29] O. Garćıa-Alarćon, S. Puga-Guzḿan, and J. Moreno-Valenzuela, ‘On parameter identification of the Furuta pendulum’, *Procedia Eng.*, vol. 35, pp. 77–84, 2012. https://doi.org/10.1016/j.proeng.2012.04.167

[30] T. Harčarik, J. Bocko, and K. Masláková, ‘Frequency analysis of acoustic signal using the Fast Fourier Transformation in MATLAB’, *Procedia Eng.*, vol. 48, pp. 199–204, 2012. https://doi.org/10.1016/j.proeng.2012.09.505

[31] P. Monsalve *et al.*, ‘Next generation PERG method: expanding the response dynamic range and capturing response adaptation’, *Transl. Vis. Sci. Technol.*, vol. 6, no. 3, p. 5, 2017. https://doi.org/10.1167/tvst.6.3.5

[32] E. D. Übeyli, ‘Statistics over features: EEG signals analysis’, *Comput. Biol. Med.*, vol. 39, no. 8, pp. 733–741, 2009. https://doi.org/10.1016/j.compbiomed.2009.06.001

[33] R.-Y. Rong *et al.*, ‘Machine-learning model of eye metastasis in nasopharyngeal carcinoma based on the AdaBoost method’, 2023. https://doi.org/10.21203/rs.3.rs-3300766/v1

Multi-Level CNN Feature Fusion from ResNet50 for Near-Duplicate Image Detection in Real Estate Imagery

Taras Panchenko[1*], Artem Bozhok[1,2] and Volodymyr Kubytskyi[1]
[1]Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska Street, 01601 Kyiv, Ukraine
[2]LUN.ua
E-mail: taras.panchenko@knu.ua
[*]Corresponding author

Keywords: image vector representation, image descriptor, near-duplicate images, image near similarity, convolutional neural network, intermediate layers, ResNet50, image embedding

Received: September 25, 2025

The volume of images uploaded to the internet is increasing at an unprecedented rate, making image deduplication, through accurate near-duplicate detection, a critical task in computer vision. However, comparing images for similarity remains challenging due to complex visual structures and subtle appearance variations.

We propose a novel embedding method for image similarity detection. It constructs an enriched representation by concatenating outputs from multiple intermediate layers of a pre-trained ResNet50 convolutional neural network and trains a lightweight decision network on top to classify image pairs. Unlike aggregation approaches that average or sum intermediate features, our method preserves both low-level and high-level information in a single descriptor and maintains feature diversity. The multi-level embedding is further normalized to balance feature contributions and is evaluated against classical keypoint descriptors, a DCT-based perceptual hash, and a standard single-layer ResNet50 embedding.

We evaluate this method on three real-world image deduplication tasks derived from real estate listings, covering (a) near-identical property photos with graphical overlays, (b) interior room photographs captured from different angles, and (c) schematic floor plan images. The proposed embedding achieves F1-scores of 0.96, 0.87, and 0.77, representing a 10-15% absolute improvement over baseline methods, including classical feature descriptors and standard ResNet50 final-layer embeddings.

This approach has been successfully deployed in production on a large-scale real estate platform, reducing duplicate images and improving search quality. The results demonstrate that multi-layer CNN embeddings with explicit feature preservation offer a robust and scalable solution for near-duplicate image detection in structured domains such as real estate photography and schematic floor plans.

Povzetek: Predlagan je nov pristop za primerjanje slik, ki izboljša zaznavanje podvojenih slik in se izkaže kot učinkovit v praksi.

1 Introduction

Recent advancements in artificial intelligence have driven significant progress in image processing technologies. For example, generative models like OpenAI's DALL-E [1] and DALL-E 2 [2] can create realistic images and art from text prompts. At the same time, AI-based vision models have achieved or even surpassed human-level accuracy on certain image recognition benchmarks [3]. Despite these breakthroughs, there remain many practical image-focused tasks beyond generation and classification that are crucial for industry. One such problem is image deduplication, which relies on near-duplicate detection methods and image similarity modeling [4-7]. This task involves determining whether two images depict the same content or scene, even when they differ in resolution, perspective, lighting, or undergo minor edits. Efficient solutions to image deduplication are increasingly important as internet platforms deal with billions of images, where removing duplicates can save storage and improve user experience.

On real-estate platforms, near-duplicate photos are common; reliable detection improves storage efficiency and user experience.

Several public benchmarks define the problem space for near-duplicate detection, including the DISC21 dataset for large-scale copy detection tasks. These datasets capture common real-world transformations such as cropping, overlays, and perspective shifts, and provide standardized baselines for evaluating and comparing methods.

However, reliably detecting near-duplicate images is challenging. Images may undergo various transformations (scaling, rotation, color changes, etc.),

and defining a robust measure of visual similarity is non-trivial. Early approaches relied on hand-crafted feature descriptors such as SIFT [8, 9], SURF [10], and ORB [11]. These methods are effective for local structure matching but exhibit reduced robustness when confronted with complex, high-level patterns and real-world variations. These limitations motivated the shift toward learned representations. Numerous research efforts have extended these ideas, including geometric invariant features in transform domains [12] and perceptual hashing techniques [13,14]. However, existing approaches still show reduced accuracy and robustness under real-world transformations such as rotations, overlays, and scale variations.

In the last decade, deep learning approaches have revolutionized image representation learning [15-42]. Convolutional Neural Networks (CNNs) in particular can automatically learn rich feature hierarchies from images, and their use for image representation has become widespread. Typically, a pre-trained CNN (e.g., ResNet [37]) is used as a feature extractor by taking the output of its final layer (or an intermediate pooling layer) as a global image descriptor. CNN-based embeddings have consistently outperformed hand-crafted features in image retrieval and recognition tasks. For instance, the output of the penultimate layer of a pre-trained ResNet50 [37] (2048-dimensional feature) can serve as a descriptor for an image pair similarity comparison. Relying solely on a network's classifier head to represent images is suboptimal for similarity. That stage is optimized to separate categories, not to preserve nuanced cues that distinguish look-alikes.

In this paper, we address these shortcomings by proposing a novel image embedding technique that leverages intermediate CNN layer outputs. By combining features from multiple depths of the network, we obtain a more descriptive vector representation that encapsulates both low-level and high-level image content.

$$f_{inter} = \phi(Flatten(A^{(l_1)}), \ldots, Flatten(A^{(l_k)})) \quad (1)$$

where: f_{inter} - feature map from layer l_i (tensor of size $C_i \times H_i \times W_i$), $Flatten(\cdot)$ - operation of tensor alignment into a 1D vector, $\phi(\cdot)$ - aggregating function (Concatenate, AveragePool, MaxPool), k - number of intermediate layers used, f_{inter} - image feature vector built from convolutions.

We apply this representation to the image deduplication problem and demonstrate significant improvements in detection accuracy (as measured by F1-score) over the evaluated baselines: classical keypoint descriptors (SIFT/SURF/ORB), a DCT-based perceptual hash, and a single-layer ResNet50 embedding [4, 5, 31-37]. We validate our approach on three distinct use-cases of image similarity in the real estate domain, showing that our method can robustly handle different types of near-duplicate image scenarios. Moreover, our solution has been deployed in a production environment (at a large real estate listing company) to automatically eliminate duplicate images, underlining its practical relevance.

In this work, we use the term image deduplication to refer to the practical task of detecting and removing redundant or duplicate content from large-scale image collections. Near-duplicate detection refers more specifically to identifying images of the same scene or object that differ slightly due to edits, overlays, or viewpoint changes. Image similarity is the broader technical problem underlying both, which involves quantifying how visually alike two images are.

2 Related work

Existing research on near-duplicate image detection spans classical descriptors, perceptual hashing, and deep learning-based embeddings. Below, we summarize key approaches and highlight the gaps our work addresses.

Hand-Crafted Feature Methods: Early approaches to near-duplicate image detection relied on hand-crafted visual features and descriptors. One of the most influential classical methods is the Scale-Invariant Feature Transform (SIFT) by Lowe [8], which detects salient local keypoints and encodes them as invariant descriptors. SIFT features, and related descriptors like SURF [10] and ORB [11], have been successfully used to match images that contain identical objects or scenes, even under moderate transformations. For example, Chum et al. [9] demonstrated a system for finding near-identical images by efficiently matching SIFT keypoints in a large database. Other works improved robustness by incorporating geometric verification or by focusing on specific invariant features. Ke et al. [15, 16] proposed methods for fast near-duplicate detection and sub-image retrieval that break images into parts and use local feature matching with optimizations for speed. While traditional feature-based methods remain effective for simple transformations, their precision and recall degrade substantially when handling appearance variations or tasks requiring high-level semantic matching. A recent survey by Thyagharajan and Kalaiarasi [4] provides a comprehensive review of image near-duplicate detection techniques, concluding that while keypoint-based methods are useful, they may not suffice for more complex similarity tasks involving, for instance, different perspectives or non-identical but related content.

Global and Perceptual Hashing Techniques: Another line of work for image deduplication involves computing compact global signatures (or "hashes") for images that preserve visual similarity. Perceptual hashing methods, such as those based on Discrete Cosine Transform (DCT) or wavelet transforms, compress an image into a short binary code that changes only slightly for minor image modifications. One popular approach is the average hash or DCT hash, which was used as a baseline in our experiments. A DCT-based hash is obtained by resizing an image, applying the DCT, and quantizing the low-frequency coefficients into bits [13, 14]. The resulting binary hash can be compared via Hamming distance - a small Hamming distance indicates two images are likely near-duplicates. Perceptual hashes are computationally efficient, enabling rapid filtering of clearly dissimilar images in large-scale datasets. For

instance, if an image's hash differs significantly from a query hash, that image can be immediately excluded from further consideration, as it is unlikely to be a duplicate. Systems like Facebook's image DNA and pHash leverage such techniques to handle web-scale image matching. However, these hashes offer limited discriminative power; they are prone to collisions and insensitive to fine-grained differences, resulting in reduced recall for near-duplicate detection. Thus, perceptual hashing is often used in combination with more detailed comparisons.

Classical and Learning-based Similarity Measures: Beyond local features and hashing, researchers have explored various other strategies. Some methods use statistical measures and entropy of pixel patterns [18, 19] or exploit geometric invariants, such as features in the Radon transform domain [12], to identify duplicates from transformed images. Others applied bag-of-visual-words models and indexing structures for scalable retrieval of duplicate images [21]. Liu et al. [22], for example, introduced a variable-length signature for image matching, adapting the representation length to image content. Clustering techniques have also been employed; Xie et al. [23] used affinity propagation clustering on deep features to rapidly narrow down candidates for near-duplicates in a web-scale ImageWeb database. These methods contributed useful ideas (e.g., using dimensionality reduction, indexing, clustering) to tackle the efficiency challenges in large-scale duplicate search. Nonetheless, the advent of deep learning brought a substantial improvement in representation quality, as described next.

CNN-Based Embeddings for Near-Duplicates: With deep CNNs achieving high results in image recognition, they naturally became attractive for image similarity tasks. A straightforward approach is to use features from a CNN pre-trained on ImageNet [3] as image descriptors. For instance, a common baseline is to take the 2048-dimensional output of the global average pooling layer of ResNet50 (which precedes the classification layer) as the image's feature vector. In our experiments, we use this as one baseline, referred to as the "ResNet50 embedding" method. Such deep features generally outperform hand-crafted features on retrieval and matching tasks because they encode more semantic information. However, they may sometimes be too coarse for distinguishing near-duplicates, as they emphasize high-level content over low-level differences. Recent studies have proposed multiple enhancements to CNN-based embeddings for near-duplicate image detection. Zhou et al. [31] proposed a coarse-to-fine scheme that uses both global CNN features and local region features: an initial comparison on global features finds candidates, and then a more detailed local feature matching (using CNN-based keypoints) confirms the duplicates. This approach enables real-time near-duplicate detection by eliminating costly comparisons between dissimilar pairs. Kordopatis-Zilos et al. [32] introduced the idea of aggregating multiple intermediate CNN layer outputs for near-duplicate video frame retrieval. By summing or concatenating feature maps from different layers of a CNN, their system captured multi-level characteristics of frames and improved retrieval accuracy. Our approach is inspired by a similar intuition, though we apply it to static images and use concatenation of intermediate layers in ResNet50 to form a single descriptor. Zhang et al. [33] tackled a related problem of cross-modal (infrared vs. visible light) near-duplicate image detection by comparing CNN embeddings after aligning images with a spatial transformer network, illustrating the breadth of scenarios where learned embeddings are useful.

Transformers and Other Modern Approaches: In 2023 and 2024, new state-of-the-art techniques have emerged for image copy detection and similarity, taking advantage of advances in deep learning architectures. Notably, some works incorporate Vision Transformers (ViT) and Siamese networks to improve robustness. Fawzy et al. [43] combined a ViT-based feature extractor with dynamic data augmentation and efficient sampling to excel in an image copy-detection challenge. Their method augmented images and learned transformer-based embeddings that proved more invariant to edits and noise, achieving top accuracy on the DISC21 copy-detection dataset. Other researchers have integrated traditional perceptual hashing with deep learning: for example, by using a ViT to generate perceptual hash codes [44] or by using a Siamese CNN that learns to predict if two images are duplicates, then refining that with a hash filter [45]. Another frontier is the exploration of quantum algorithms for image similarity: Yang et al. proposed a quantum inner-product method to compute image similarities more efficiently, encoding classical CNN-derived feature vectors into quantum states. While such approaches are still experimental, they indicate the growing interest in accelerating and improving near-duplicate detection through unconventional means.

In summary, the field has evolved from keypoint matching and hashing toward learned embeddings and hybrid systems that integrate efficiency with accuracy. Yet, there remains a gap in achieving both high recall and precision across diverse image transformations in deployment-constrained settings. Our work contributes to this landscape by introducing an intermediate-layer CNN embedding approach evaluated against baselines that are commonly used or straightforward to integrate in industrial deduplication pipelines (classical keypoints, DCT-based perceptual hashing, and a single-layer ResNet50 embedding). Rather than competing directly with the latest transformer-based or task-specific copy-detection systems, we focus on quantifying and explaining the gains obtained by multi-layer CNN feature fusion over these widely adopted baselines in the context of real estate imagery.

To contextualize our approach, Table 1 summarizes representative families of methods used in near-duplicate image detection. For clarity, the table also includes an illustrative F1-score measured in our own evaluation on Task A (near-identical images with graphical overlays), using a consistent train/validation/test protocol. These values reflect evaluation on our dataset and are included to show how the method families behave under the same

conditions. The full quantitative comparison across Tasks A-C, with confidence intervals, is reported in Table 3.

Table 1: Representative method families for near-duplicate image detection, with Task A performance from our evaluation

Method	Feature Type	Robust to Transformations	Application Domain	F1-score (Task A)	Score source	Key Limitation
Classical SIFT/SURF/ORB	Local keypoints (hand-crafted)	Moderate (scale/rotation only)	General image matching/retrieval	0.81	This work (reimplementation)	Limited semantic understanding
Perceptual Hash	Global binary hash	Low (sensitive to overlays, edits)	Web-scale deduplication	0.70	This work (reimplementation)	Poor fine-grained discrimination
Single-layer ResNet50	Deep features (penultimate layer)	High for natural images	Image retrieval, deduplication	0.85	This work (reimplementation)	Misses low/mid-level details
Proposed method	Multi-layer ResNet50 + decision network	High (low-mid-high features)	Real estate (photos + floor plans)	0.96	This work (proposed)	Higher compute cost but scalable

3 Methodology

Our proposed solution for image deduplication is centered on a multi-layer CNN embedding that serves as a robust image descriptor, coupled with an efficient workflow for comparing image pairs. The methodology is organized into three stages: feature extraction from a pre-trained CNN, embedding construction (concatenation and normalization), and similarity decision modeling. Efficient storage and indexing are described separately in the efficiency subsection. Figure 1 summarizes the full pipeline and shows how these stages connect.

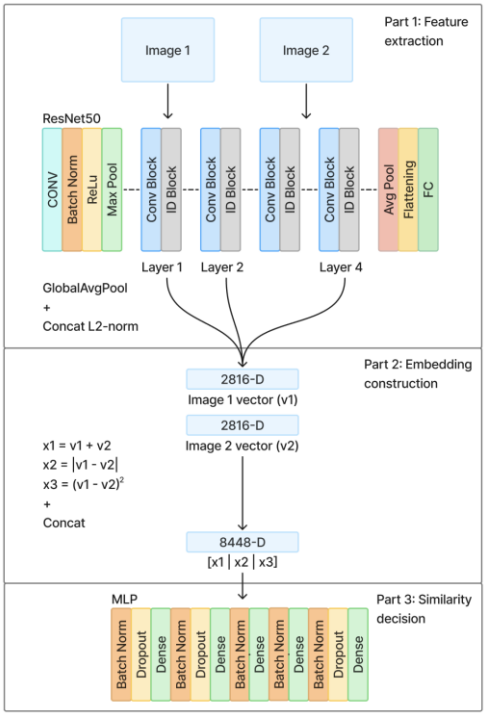


Figure 1: Pipeline overview

Algorithm 1 summarizes the complete pipeline in a reproducible, implementation-agnostic form.

Algorithm 1: End-to-end pipeline pseudocode

Input: labeled image pairs (I_a, I_b, y).

Output: trained decision model and inference procedure.

(1) Feature extraction (frozen backbone)

For each image I:

1. Decode to RGB (3 channels). If an input is single-channel (e.g., some floor plans), replicate to 3 channels.
2. Resize to 224×224 (bilinear interpolation).
3. Apply ImageNet normalization per channel: mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225].
4. Run frozen ResNet-50 (IMAGENET1K_V2) and tap stages C2=layer1, C3=layer2, C5=layer4.
5. Apply global average pooling to each tapped feature map to obtain vectors v_{C2}, v_{C3}, v_{C5}.
6. Concatenate $e = [v_{C2}; v_{C3}; v_{C5}]$ (2816-D).
7. L2-normalize $e \leftarrow \frac{e}{\|e\|_2}$. Store e.

(2) Pair fusion

Given embeddings e_1, e_2:

$z = [e_1 + e_2; |e_1 - e_2|; (e_1 - e_2)^2]$ (8448-D).

(3) Decision model training

1. Train the MLP on z with BCE loss and Adam (lr =10^{-5}), batch size 64, up to 30 epochs, using the specified class weights.
2. Select the best checkpoint by validation accuracy.
3. Choose decision threshold τ on validation to maximize F1; evaluate once on the held-out test split.

(4) Inference

For a query image I_a: compute embedding e_a. For each candidate image I_i (optionally shortlisted by a hash pre-filter), compute $p = MLP(z(e_a, e_i))$. Predict duplicate if $p \geq \tau$.

The remaining subsections detail each stage of Algorithm 1.

3.1 Image embedding construction using intermediate CNN layers

To build the proposed vector image representations, a pre-trained ResNet50 [3] CNN - originally meant for image classification - is employed. The model's final output layer is removed, and activations from selected intermediate layers are extracted and concatenated into a unified feature vector, which serves as the image embedding. This vector is then normalized so that the magnitude of any one feature (numerical component) does not dominate the similarity measure. The resulting representation is robust to geometric transformations, illumination changes, noise, and watermarks, enabling reliable similarity-based comparisons.

$$f = \frac{f}{\sqrt{\sum_{i=1}^{d} f_i^2 + \epsilon}} \quad (2)$$

where: f - is the initial feature vector, f_i - i-th component of the vector, d - vector dimension, ϵ - small addition to avoid division by zero (e.g., 10-8), f - normalized vector (unit length).

The detailed steps for creating vector representations of images using a combination of intermediate layers from a pre-trained ResNet50 are outlined below.

1. Load the pre-trained ResNet50 model.

The first step is to load a ResNet50 model that has been pre-trained on a large-scale image-classification dataset such as ImageNet [3]. This allows us to reuse the learned weights to extract high-, mid-, and low-level features from our input images.

2. Remove the final fully-connected layer.

Because the pre-trained ResNet50 is meant for image classification, its last layer is a fully-connected head that outputs class probabilities. We discard this layer, as it is not needed for building an image embedding.

3. Obtain activations from intermediate layers.

We extract the activations of ResNet50's intermediate layers, since each layer captures features at a different level of abstraction. These activations can be viewed as feature maps that reveal how strongly each learned filter responds to the input. To gather them, we pass the image through the network and record the activations at every chosen intermediate layer.

Selecting which layers to extract:

The choice of intermediate layers depends on the required level of abstraction and feature complexity for the specific task. In this work, we tap three stages of the torchvision ResNet-50: layer1 (C2), layer2 (C3), and layer4 (C5). For each stage's activation tensor (with channel counts 256, 512, and 2048, respectively), we apply global average pooling over spatial dimensions to obtain vectors of sizes 256, 512, and 2048. We then concatenate these into a 2816-D descriptor and apply L2-normalization. This choice provides a balanced mix of low-level (edges/textures), mid-level (geometric parts/layout), and high-level (object/scene) cues, while keeping the descriptor compact enough for large-scale indexing.

$$f^{(l)} = GlobalAvgPool(A^{(l)}), A^{(l)} \in R^{C_l \times H_l \times W_l} \quad (3)$$

where: $A^{(l)}$ - feature map on the layer l with C_l channels, height H_l and width W_l, $GlobalAvgPool$ - global average convolution along spatial axes, $f^{(l)}$ - vector R^{C_l}, that aggregates the layer.

For comparison, the single-layer baseline that uses only the penultimate stage of ResNet-50 produces a 2048-D embedding. Because memory usage scales with embedding size, the proposed 2816-D descriptor increases storage by approximately 1.38×. Distance computations in ANN indexes also scale roughly linearly with dimensionality, so the per-comparison arithmetic cost increases by a similar factor. Overall latency is mitigated by candidate selection via a hash pre-filter, after which only a small subset undergoes full vector comparison (see "Efficient Embedding Storage and Indexing").

Empirical layer selection and ablation (overview). We evaluated several candidates tap sets (e.g., {C2,C3,C4}, {C2,C3,C5}, {C2,C4,C5}) on a held-out validation split and selected {C2,C3,C5} as the default configuration. A full ablation study quantifying the contribution of each stage and their combinations - single-stage, two-stage, and three-stage descriptors - is reported in "Experiments and Results" (Tables 6-7), including per-task F1 and threshold-independent ROC-AUC/PR-AUC for Task A.

4. Merging the intermediate-layer activations.

The selected intermediate activations are aggregated into a single vector that serves as the image representation. We evaluated several aggregation strategies, including averaging/max pooling, Generalized Mean (GeM) pooling, and NetVLAD. We adopt concatenation of per-layer GAP descriptors because it is training-free and stable across heterogeneous data (photos and schematic floor plans), preserves complementary cues from different depths, and has predictable latency and memory costs for large-scale indexing. In pilot experiments on our validation split, GeM and NetVLAD did not yield consistent improvements over concatenation and added compute and implementation overhead; therefore, we use concatenation in the main method.

In summary, our vector representation leverages a pre-trained CNN, removes its classification head, and concatenates the chosen intermediate features (after per-layer global average pooling) into one normalized feature vector used for similarity.

Such embeddings support many comparison scenarios, including:

- Image retrieval. Given a query image, its vector can be used to retrieve similar images from a database.
- Similarity measurement. Two embeddings can be compared directly to quantify how alike the underlying images are.
- Classification. The vector can serve as input to a classifier that assigns the image to predefined categories.

3.2 Efficient embedding storage and indexing

For efficient similarity search, it is important to determine how to index image representations in a database. Indexing enables efficient retrieval and can be implemented using algorithms such as KD-trees, Ball trees, Annoy hash comparison, or DCT-based methods [13].

A DCT hash comparison can serve as a preliminary filter when working with multi-million-image databases, removing images that differ substantially from the query. A DCT hash is a compact, reliable image descriptor obtained by applying the discrete cosine transform (DCT) to a small, down-sampled version of the image. The resulting DCT coefficients are quantized and combined into a single DCT hash [14].

DCT hashes are compared by computing the bitwise distance between the query image's DCT hash and the DCT hashes stored in the database. This distance is typically measured with the Hamming metric - the number of bit positions at which the two binary strings differ.

$$b_i = sign(DCT(x_i) - \mu_{DCT}), d_H(b_1, b_2) = \\ = \sum_{j=1}^{n} I[b_1^{(j)} \neq b_2^{(j)}] \quad (4)$$

where: x_i - pixel or block of an image, $DCT(x_i)$ - coefficient after cosine transformation, μ_{DCT} - average value of DCT coefficients, b_i - the corresponding hash bit (0 or 1), d_H - Hamming distance between hashes, $I[\cdot]$ - indicator function (1, if true).

Images with large Hamming distances are treated as dissimilar and excluded from further embedding comparisons. Doing so greatly reduces the search space and therefore improves efficiency.

3.3 Similarity decision model

To compare two images, we use their embeddings $e_1, e_2 \in R^d$ and construct a fused pair representation using three element-wise channels: $e_1 + e_2$ (sum), $|e_1 - e_2|$ (absolute difference), and $(e_1 - e_2)^2$ (squared difference). These three channels are concatenated into a single vector $z \in R^{3d}$. With the proposed multi-layer descriptor $d = 2816$, the decision network input dimension is therefore $3d = 8448$.

Backbone usage. The ResNet-50 backbone is used as a fixed feature extractor: it is frozen and not fine-tuned during decision-model training. We use torchvision ResNet-50 pretrained weights IMAGENET1K_V2. The MLP is trained on pairs of precomputed embeddings.

Decision network architecture. The fused vector z is fed to a fully connected MLP with Batch Normalization (BN) after each Dense layer, ELU activations in hidden layers, and a final sigmoid producing a match probability $p \in [0, 1]$. The architecture used in all experiments is specified in Table 2.

Table 2: Decision MLP architecture

Block	Operation	Output dim	Activation	Dropout
Input	Pair-fusion concat $[e_1 + e_2, \lvert e_1 - e_2 \rvert, (e_1 - e_2)^2]$	8448	-	-
Preprocess	BN	8448	-	-
FC1	Dropout → Dense → BN	2048	ELU	0.50
FC2	Dropout → Dense → BN	1024	ELU	0.50
FC3	Dense → BN	512	ELU	-
FC4	Dense → BN → Dropout	256	ELU	0.25
Output	Dense	1	Sigmoid	-

Training objective and optimization. The model is trained with binary cross-entropy (BCE). We use Adam with a learning rate 1×10^{-5}, batch size 64, and train for 30 epochs. We select the best checkpoint based on validation accuracy (ModelCheckpoint with $monitor = val_accuracy$, $mode = max$), and use that checkpoint for evaluation.

Class imbalance handling. We apply class weighting during training. The weights used are: class 0 weight $= 0.9947$, class 1 weight $= 1.0053$, indicating a near-balanced training split (approximately negative:positive $\approx 1.01:1$ under standard "balanced" class-weight computation).

Inference and thresholding. At inference time, the network outputs a probability p. The decision threshold τ is selected on the validation split (maximizing F1 for the corresponding task) and then fixed for evaluation on the held-out test split.

4 Experiments and results

We evaluated the proposed method on three real-world image deduplication scenarios, drawn from an industry dataset of real estate images. These scenarios correspond to distinct sub-problems frequently encountered in practice:

Task A: Near-identical images with different contexts. These are image pairs that are essentially the same photograph or graphic, but one might have additional graphical elements (such as watermarks, borders, or slight cropping differences). Examples of such image pairs are shown in Figure 2. We assembled a dataset of 10,000 such image pairs [47], with labels indicating duplicates vs. non-duplicates. This dataset is publicly available and is the benchmark used for Task A; the corresponding results are reported in Table 3. Threshold-independent ROC-AUC and PR-AUC for Task A are also reported in Table 4.



Figure 2: An example of near-identical images with different context

Task B: Multi-angle images of rooms. Here, the goal is to detect if two photos are of the same room or interior space, even if taken from different viewpoints or at different times. This task is challenging because, although the images are not pixel-identical, they depict the same physical space and can be recognized as such by humans. An example of such image pairs are shown in Figure 3. Our test dataset contains 12,500 image pairs of interiors, covering both matching pairs (same room) and non-matching pairs (different rooms), based on an apartment listing database.



Figure 3: An example of multi-angle images of rooms

Task C: Schematic floor plan matching. In this task, we deal with binary or line-drawing floor plan images. Two floor plans should be detected as duplicates if they represent the same layout, even if their orientations, scale, or slight drawing details differ. An example of such image pairs are shown in Figure 4. The test dataset includes 8,800 pairs of floor plan images, with ground truth on which pairs depict the same property layout.



Figure 4: An example of the same schematic floor plans

Dataset characterization, labeling, and reproducibility. We evaluate on three task-specific datasets constructed from real-estate imagery. Task A is public; Tasks B and C are derived from proprietary listing content. Although Tasks B/C cannot be redistributed due to licensing and privacy considerations (interior photographs may contain sensitive information), we provide sufficient characterization and a fully specified split strategy so that the methodology can be reproduced on alternative datasets.

Labeling. All three tasks use manually created labels for duplicate/non-duplicate image pairs. During annotation, human annotators assigned each pair to the positive (match/near-duplicate) or negative class using task-specific definitions. For Task A, labels correspond to near-identical images that may differ by graphical overlays, borders, minor crops, resizing, or compression. For Task B, labels indicate whether two interior photos depict the same physical room across viewpoint and illumination changes. For Task C, labels indicate whether two floor plans depict the same layout, allowing for rotation, scale changes, redraw differences, and added text or graphics.

Class balance. For reproducibility and stable threshold selection, the datasets were sampled to be approximately balanced: the positive/negative pair ratio is near 1:1 in train and validation split for all tasks.

Image resolution and preprocessing. Original images vary in size (shortest side ~320 px to longest side ~1920 px). For embedding extraction, each image is decoded to RGB, resized to 224×224, and normalized with the standard ImageNet channel statistics (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]). We do not apply synthetic augmentations (e.g., random crops/flips) during decision-network training because the classifier is trained on precomputed embeddings; instead, each task's labeled pairs already contain natural variations relevant to deployment (overlays, viewpoint changes, rotation/scale differences in floor plans, etc.).

Typical variation per task:

- Task A (near-identical with graphical context changes): overlays (watermarks, borders, text), mild cropping, resizing, compression artifacts, and small color/contrast changes.
- Task B (multi-angle interiors): viewpoint changes, partial overlaps, occlusions by

furniture, clutter changes, and lighting/exposure shifts.
- Task C (floor plans): rotation and scale variation, redraw differences, compression artifacts, and added symbols/text labels.

We evaluated four approaches: (1) the proposed intermediate-layer ResNet50 embedding with a learned decision model, (2) a single-layer ResNet50 embedding using the penultimate layer (Global CNN Embedding), (3) classical feature descriptors based on SIFT, SURF, and ORB (Keypoint Descriptors), and (4) a perceptual hashing baseline using a DCT-based hash (DCT Hash).

Baseline implementation and tuning protocol. To ensure fair comparison across baselines, we used a single train/validation/test split per task (70/10/20) with group-wise separation to prevent leakage, and we performed all model selection and threshold tuning on the validation split only, reporting final metrics on the held-out test split.

Global CNN embedding (single-layer ResNet50). We extract the 2048-D feature vector from the penultimate stage of a pre-trained ResNet50 and L2-normalize it. For a pair of images, we compute similarity using cosine similarity, then select a decision threshold on the validation set that maximizes F1 for the given task.

Perceptual-hash (DCT hash). For each image we compute a DCT-based perceptual hash and compare pairs using Hamming distance. A validation-tuned Hamming threshold determines whether a pair is classified as duplicate/non-duplicate.

Keypoint-descriptor (SIFT/SURF/ORB). We implement a standard local-feature matching pipeline using classical descriptors (SIFT, SURF, ORB). For each image pair we compute a match score based on descriptor correspondences (with standard match filtering and optional geometric consistency checks where applicable). The final duplicate/non-duplicate decision is made by thresholding this match score; the threshold (and, where relevant, the specific descriptor variant) is selected on the validation split to maximize F1, and then fixed for test evaluation.

Across all baselines, the validation-only tuning protocol ensures that reported test-set numbers reflect generalization rather than threshold overfitting.

To assess the effectiveness and statistical robustness of the proposed method, we conducted evaluations across three mutually exclusive test sets, corresponding to:
- Task A: Near-identical property photos with graphical overlays.
- Task B: Multi-angle interior photographs.
- Task C: Schematic floor plan images.

For each task (A-C), we use a held-out test set and split pairs into 70% train / 10% validation / 20% test. To prevent leakage, we apply group-wise splitting using a grouping key based on (listing id, image category), ensuring that all pairs belonging to the same listing and category are assigned to the same split. This prevents near-duplicate images originating from the same listing from appearing across different splits, and also respects the structural differences between categories such as interior photographs versus floor plans. Model selection and threshold tuning are performed on the validation split only, and final performance is reported on the held-out test split.

Across tasks, datasets are mutually exclusive. Mutual exclusivity between Task A (near-identical property photos) and Task B (multi-angle interiors) was ensured through manual verification during annotation, where these image types were reviewed side-by-side to avoid overlap. For Task C (floor plans), we first separated images using an image category classification algorithm, followed by additional manual verification during annotation to ensure that no floor-plan images or pairs overlap with Tasks A/B.

For Task A (near-duplicate images), we additionally report threshold-independent ROC-AUC and PR-AUC on the held-out test split (Table 4).

Table 3: The F1-scores for each approach on the three tasks

Task \ Method	**(1)****	**(2)**	**(3)**	**(4)**
Near-duplicate images with different graphic contexts (dataset size: 10 000 image pairs)	**0.96 ± 0.01***	**0.85 ± 0.02**	**0.81 ± 0.03**	**0.70 ± 0.03**
Multi-angle room photographs (dataset size: 12 500 image pairs)	**0.87 ± 0.02***	0.78 ± 0.03	0.76 ± 0.04	0.65 ± 0.04
Schematic layouts / floor plans (dataset size: 8 800 image pairs)	**0.77 ± 0.02***	0.68 ± 0.03	0.65 ± 0.03	0.63 ± 0.03

* Bold indicates the best result for each task (higher F1 is better).
** Column meanings: (1) F1-score of the proposed multi-layer descriptor with the learned MLP decision model, (2) F1-score of a single-layer ResNet50 embedding evaluated via cosine similarity + validation-tuned threshold (no MLP), (3) F1-score of SIFT/SURF/ORB descriptors, (4) F1-score of the DCT perceptual hash.

Table 4: ROC-AUC, PR-AUC metrics on the held-out test split

Method	ROC-AUC	PR-AUC
Proposed (multi-layer + MLP)	0.982 [0.981, 0.986]	0.975 [0.974, 0.981]
Global CNN embedding	0.923 [0.919, 0.934]	0.885 [0.880, 0.907]
Keypoint descriptors	0.894 [0.887, 0.911]	0.848 [0.843, 0.874]
DCT perceptual hash	0.807 [0.801, 0.830]	0.727 [0.723, 0.761]

* Values shown as median [95% CI] over 100 bootstrap resamples of the Task A test set.

Table 3 reports the mean F1-score ± 95% CI across the three tasks. The proposed intermediate-layer embedding method consistently achieves the highest performance, confirming its robustness.

Since the central contribution of this work is multi-level aggregation of intermediate CNN stages, we conduct a structured ablation across the ResNet-50 tap points used in the proposed descriptor: C2 (layer1), C3 (layer2), and C5 (layer4). For each task (A-C), we evaluate: (i) each single stage alone, (ii) each two-stage combination, and (iii) the full three-stage descriptor. Protocol. For every ablation setting, we follow the same train/validation/test split and leakage prevention described earlier. To isolate the effect of representation depth, all ablation variants are evaluated with the same pair-fusion and the same MLP decision head (sum, absolute difference, squared difference channels, followed by the MLP), with the decision threshold tuned on validation only and metrics reported on the held-out test set.

Metrics. We report F1 for all tasks (A-C). For Task A, we additionally report threshold-independent ROC-AUC and PR-AUC for each ablation setting.

Table 5: Ablation of ResNet-50 tap points (C2/C3/C5). F1 on held-out test sets (95% CI)

Descriptor variant	Task A F1	Task B F1	Task C F1
C2 only	0.77 ± 0.03	0.68 ± 0.03	0.63 ± 0.04
C3 only	0.80 ± 0.03	0.70 ± 0.02	0.67 ± 0.03
C5 only	0.87 ± 0.02	0.80 ± 0.02	0.69 ± 0.02
C2 + C3	0.82 ± 0.02	0.73 ± 0.03	0.69 ± 0.03
C2 + C5	0.89 ± 0.02	0.83 ± 0.02	0.71 ± 0.02
C3 + C5	0.92 ± 0.02	0.85 ± 0.02	0.74 ± 0.03
C2 + C3 + C5	**0.96 ± 0.01**	**0.87 ± 0.02**	**0.77 ± 0.02**

Table 6: Threshold-independent ablation results on task A (median [95% CI] over 100 bootstrap resamples)

Descriptor variant	ROC-AUC	PR-AUC
C2 only	0.902 [0.895, 0.908]	0.841 [0.829, 0.854]
C3 only	0.925 [0.919, 0.929]	0.877 [0.866, 0.888]
C5 only	0.965 [0.962, 0.968]	0.943 [0.937, 0.949]
C2 + C3	0.937 [0.932, 0.941]	0.896 [0.886, 0.906]
C2 + C5	0.975 [0.973, 0.977]	0.959 [0.954, 0.963]
C3 + C5	0.980 [0.978, 0.981]	0.973 [0.970, 0.976]
C2 + C3 + C5	**0.982 [0.981, 0.986]**	**0.975 [0.974, 0.981]**

The ablation results show that C5 is the strongest single stage across all tasks, indicating that high-level semantic structure provides a strong baseline for duplicate detection. However, adding intermediate stages yields consistent gains: the best two-stage variant is C3+C5, and the full C2+C3+C5 descriptor achieves the best overall performance. The improvements are not purely dominated by one stage: C3 provides a large complementary boost over C5, while C2 adds a smaller but consistent additional gain

when combined with C3+C5. The threshold-independent ROC-AUC/PR-AUC trends in Table 6 mirror the F1 ordering, confirming that the triplet improves the underlying ranking quality rather than only benefiting from threshold selection.

The "Global CNN embedding" baseline in Table 3 is evaluated using direct cosine similarity between single-layer embeddings with a tuned threshold, whereas the ablation settings here use the same MLP decision head for all variants (including C5-only). Consequently, the C5-only ablation results should not be interpreted as identical to the cosine-similarity single-layer baseline; the ablation is designed to isolate representation depth under a fixed decision function.

We report runtimes on an Intel Core i7 12-gen class server, batch size = 32, images are resized to 224×224. The pipeline follows our design with a hash pre-filter to shortlist candidates and a learned MLP decision network operating on fused embedding channels. The results are shown in Table 7.

Table 7: Runtime and throughput on CPU (single-threaded). Mean ± std; 95% CI for the mean in brackets

Stage	Mean time	95% CI	Throughput	Notes
Embedding generation (ResNet-50, multi-layer), batch = 32	2.40 ± 0.06 s/batch (≈ 75 ms/img)	[2.383, 2.417] s/batch	~13.3 img/s	Three tapped stages, GAP + concat, L2-norm
Pairwise decision (fused channels + MLP)	2.7 ± 0.3 ms/pair	[2.64, 2.76] ms/pair	~370 pairs/s	Using precomputed embeddings
End-to-end query (hash pre-filter + shortlist + MLP)	141 ± 12 ms/query	[138, 143] ms/query	~7.1qps	Typical production shortlist from the hash stage

The multi-layer descriptor increases extraction cost and descriptor dimensionality; therefore, operational efficiency depends on reducing the number of expensive comparisons. We use a DCT-hash pre-filter as a first-stage candidate selector. In our production setting, the DCT stage filters out approximately 99% of candidates and produces an average shortlist size in the hundreds. This converts the matching problem from comparing a query against an extremely large inventory to evaluating only $O(10^2)$ candidates per query in the MLP stage, which is the main reason the overall pipeline is feasible at scale.

A direct "with vs. without pre-filter" end-to-end speedup measured on a small offline dataset is not representative: without pre-filtering, latency would be dominated by the number of database candidates, and a realistic comparison requires a multi-million-scale inventory. For this reason, we report shortlist size and filtered fraction as the most informative and transferable indicators of the pre-filter's operational benefit.

Overall, across all tasks, the proposed intermediate-layer embedding yields a relative improvement of 10-15% in F1-score over using the standard single-layer CNN embedding alone, and even larger gains (up to ~20 percentage points absolute) over the classical keypoint and DCT hash baselines. Within the scope of these evaluated methods, these results validate that combining features from multiple CNN layers provides a more powerful representation for detecting image similarity. Moreover, the consistency of our method's superiority across three very different scenarios (photographic near-duplicates, cross-view photos, and schematic drawings) suggests that the approach generalizes well within the considered application domain. The pipeline relies only on generic visual features and no domain-specific metadata or heuristics, indicating that it can be applied to other image collections with minimal adaptation - typically threshold retuning and, if needed, light re-training of the decision head on a small labeled set.

Scope and limitations of baseline comparison. We emphasize that our baseline set is restricted to methods that are (i) mature and widely used in practice, and (ii) compatible with the compute and engineering constraints of our production environment. In particular, we do not include more recent transformer-based copy-detection systems, learned pooling schemes such as GeM/R-MAC, NetVLAD-style retrieval architectures, or self-supervised ViT/CLIP-style embeddings. Incorporating and fairly tuning such models would require dedicated GPU services, significant engineering integration, and access to larger labeled or pseudo-labeled datasets than were available within this deployment-focused study. Consequently, we frame our empirical claims as improvements over the evaluated baselines rather than over the full current state of the art in copy or near-duplicate detection.

Finally, as a real-world validation, the pipeline has been integrated into production on a large real-estate platform operating over a large image inventory. While exact traffic and inventory figures cannot be disclosed due to confidentiality constraints, the system is used continuously to suppress redundant content during listing ingestion and moderation. In production, the DCT pre-filter is an integral component, filtering

out approximately 99% of candidates and yielding shortlist sizes in the hundreds, after which the MLP decision stage is applied to the shortlisted pairs. Operationally, the deployment reduced moderator actions by ~40% and decreased the number of duplicate listings by ~25%, supporting the real-world impact of the proposed approach.

5 Discussion

The experimental results confirm that our approach outperforms the evaluated baselines - classical keypoint descriptors, a DCT-based perceptual hash, and a single-layer ResNet50 embedding - on all three tasks. In this section, we discuss the reasons for this improved performance relative to these methods, the novelty of our method in the context of related work, and some insights on its applicability and limitations. We focus on a carefully defined comparison against baselines that are representative, interpretable, and deployable under real-world constraints.

Why intermediate-layer embedding excels: Traditional CNN-based embeddings typically use the deepest layer of the network (just before classification) to represent an image. While this captures high-level semantic information, it can omit lower-level cues needed to distinguish near-duplicates. By incorporating intermediate CNN layers, our embedding spans multiple abstraction levels: early layers capture edges, textures, and simple patterns, whereas later layers encode object parts and global scene semantics. Concatenating these representations allows the method to exploit similarity in low-level structure and high-level arrangement within a single descriptor. Unlike approaches that average intermediate responses, our method preserves layer-specific contributions through concatenation and maintains feature diversity with a modest increase in dimensionality. In the real-estate setting, this design improved performance on floor-plan graphics and cross-angle room photos. Prior work [46] outlined combining intermediate CNN features for near-duplicate detection; we extend it by (i) specifying and justifying the ResNet-50 tap points - layer1 (C2), layer2 (C3), and layer4 (C5) - supported by ablations, and (ii) formalizing a three-channel pair-fusion (sum, absolute difference, squared difference) with a fully specified MLP decision model trained with BCE. In contrast to Kordopatis-Zilos et al. [25], who aggregate intermediate features for video keyframes, we concatenate per-layer GAP descriptors for static images to preserve layer-specific cues. To the best of our knowledge, this is the first application of this design to image deduplication in real estate imagery with quantified gains over a single-layer embedding.

Comparative analysis with recent methods: Our approach fits within the broader trend of enhancing image representations for similarity tasks. The coarse-to-fine CNN matching method of Zhou et al. [24] augments a global CNN feature with local feature matching. In contrast, we represent each image with a single multi-level embedding, so matching reduces to a vector comparison and integrates naturally with indexing and scalable retrieval frameworks. Transformer-based approaches (e.g., the ViT method of Fawzy et al., 2024) learn task-specific features for copy detection and can perform strongly on benchmark challenges, but they typically require large training sets and substantial compute. Our method reuses a pre-trained CNN and trains only a small decision network, which is comparatively lightweight and data-efficient. In our experiments, training on tens of thousands of labeled pairs was sufficient because ResNet50 provides strong base features.

When compared with classical keypoint and hashing approaches, the learned embedding is more discriminative and robust. Classical methods remain useful, particularly as an efficiency layer; therefore, we integrate a hash-based pre-filter to reduce the candidate set. Similar hybrid pipelines are common in practical systems, where hashing or clustering is used for candidate shortlisting and deep models make the final decision. In our setting, the hash stage eliminated a large fraction of irrelevant images, allowing the embedding comparison stage to focus on a smaller set of candidates.

Novelty and contribution: The core novelty of our work lies in the feature representation - the enriched image embedding derived from multiple CNN layers - and in demonstrating its benefits over strong but practically oriented baselines. This is a departure from many duplicate-detection studies that either use a single global CNN feature [24, 27] or a set of local features [9, 18], but not a single unified vector encoding multi-level information. By demonstrating that such an embedding can outperform these particular baselines, we provide evidence that multi-layer feature fusion is a promising design choice for image similarity under deployment constraints. In addition, our use of a learned decision model on top of the embeddings adds a layer of adaptability; rather than fixing a similarity metric, we let the data inform how to best combine similarity measures. While learned metrics for similarity (such as Siamese networks) have been explored, our approach is lightweight in that it does not require modifying the backbone CNN or conducting metric learning during the feature extraction phase - we only learn the final decision function. A comprehensive empirical comparison against the latest transformer- and ViT-based copy-detection and retrieval models is beyond the scope of this work and remains an important direction for future research.

We also emphasize that our method proved effective in a production environment. Integration with LUN's platform showed that the approach can handle the variability of user-uploaded data, including varying resolutions and occasional label noise. The system's ability to reduce duplicate images in this setting supports its robustness.

Discussion of failure cases: No method is without limitations. Despite these improvements, several limitations were identified. First, heavy texture or style transformations (e.g., painterly filters) produced embeddings with lower cosine similarity, leading to false negatives. Second, isolated common objects across otherwise different scenes (e.g., the same chair in different rooms) sometimes caused false positives, particularly when high-level features dominated similarity. Third, floor plan rotation and non-uniform scaling, while largely handled by intermediate features, occasionally caused mismatches in borderline cases. These errors reveal specific transformation types and object-level similarities to which the embedding remains sensitive. While the decision network partially mitigates such errors by learning contextual patterns, they remain the primary source of misclassifications and represent the most promising direction for targeted improvements. These findings suggest directions for future work, such as incorporating attention mechanisms to capture global image context or refining the feature extractor through additional training on hard negatives.

Per-task error levels. Using the operating threshold τ selected on validation, the proposed system yields the following absolute error rates on the held-out test sets: Task A: FP = 2.65%, FN = 3.24%; Task B: FP = 4.39%, FN = 4.89%; Task C: FP = 9.25%, FN = 7.65%. These values contextualize the qualitative breakdown below by showing the overall frequency of false matches and missed matches per task.

Among false positives (FPs), approximately 62% arose from different rooms with nearly identical layout/furniture but only small local differences (e.g., tile color, chandelier shape, presence/absence of small items, socket/switch placement). A further 18% were due to an isolated common object dominating the match signal (e.g., the same chair or lamp across different rooms). 9% resulted from lighting/white-balance artifacts that made surfaces appear more similar than they are, and 11% from label ambiguity at the boundary of "same vs. different" rooms. Overall, ~80% of FPs are attributable to small-object/local-detail similarity, consistent with our qualitative observations.

For false negatives (FNs), the primary causes were large viewpoint changes or occlusions (~41%), substantial clutter/furniture rearrangements (~25%), strong lighting/exposure shifts (~19%), and low resolution/blur (~15%). Notably, rooms without renovation - with repetitive textures and minimal distinctive cues - were over-represented in both FP and FN groups.

Other difficult cases we observed include mirror reflections (duplicating objects and confusing spatial cues), heavy watermark overlays (obscuring small discriminative details), and repeated developer-standard layouts across different apartments that produce near-identical geometry despite being distinct spaces.

Ablation findings. The ablation study in "Experiments and Results" (Tables 5-6) quantifies the contribution of each tapped stage and their combinations. Overall, the results show that multi-stage aggregation consistently improves performance over single-stage descriptors, and that the selected {C2,C3,C5} triplet offers the best average trade-off across the three tasks. The threshold-independent ROC-AUC/PR-AUC results on Task A further confirm that the improvement is not driven solely by threshold selection but reflects a stronger underlying ranking of matching vs. non-matching pairs.

In conclusion, the proposed intermediate-layer embedding method successfully advances the performance in image deduplication tasks, thanks to its comprehensive feature representation and learned comparison strategy. The method stands out for its combination of high accuracy, generality across tasks, and practical deployability.

6 Conclusion

This study addressed the problem of near-duplicate image detection and introduced an image descriptor generation method based on intermediate-layer CNN features that improves image similarity modeling over widely used baselines. We evaluated the method on three real estate image deduplication tasks and demonstrated that existing techniques in our comparison - ResNet50 global embeddings, classical keypoint descriptors, and DCT perceptual hashing - often exhibit limited accuracy in challenging settings. To address this limitation, we proposed an extended vector representation constructed by combining intermediate-layer outputs from a pre-trained ResNet50. We detailed a data preparation method and a step-by-step algorithm for building the model, which includes extracting and normalizing the multi-layer embeddings, storing them efficiently, and applying a learned decision neural network for comparing image pairs.

The use of intermediate CNN layers to construct the image embedding represents the key innovation and primary contribution of this work. This enriched embedding captures a more comprehensive set of image features than traditional single-layer embeddings, enabling superior discrimination between similar and dissimilar image pairs. The proposed method outperformed the evaluated baselines (ResNet50 global embedding, keypoint descriptors, and DCT hash) across all tasks on our datasets, with particularly strong gains in challenging scenarios such as cross-angle photo matching. The proposed model improved upon existing solutions, yielding more accurate and confident detection of near-duplicate images. In quantitative terms, it achieved F1-scores of 0.96, 0.87, and 0.77 on the respective tasks, outperforming baseline methods by a wide margin in each case. These results were validated on real-world data, and the model has been successfully deployed in

a production environment, demonstrating its practical applicability.

These findings suggest several promising directions for future research. One direction is to explore alternative backbone architectures or hybrid layer combinations (e.g., integrating ResNet50 and Vision Transformer features) to enhance descriptor performance. Another area is optimizing the embedding size - using dimensionality reduction techniques like PCA or autoencoders on the concatenated vector could retain the performance while reducing storage and computation needs.

Additionally, while our method was evaluated on distinct tasks separately, an interesting extension would be a unified deduplication system that can handle multiple types of near-duplicate criteria simultaneously (context duplicates, cross-angle, etc.) possibly by multi-task learning.

Future work could explore adaptive weighting of intermediate layers, allowing the contribution of each layer to be dynamically adjusted based on the image domain or transformation type.

In summary, this work presents an effective approach to image similarity modeling and deduplication by leveraging intermediate CNN features to construct a multi-level embedding. The approach aligns with industrial practice by combining deep representations with efficient candidate filtering. The method is applicable to domains that require detecting duplicate or near-duplicate images and provides a clear baseline for future work on improved representations and retrieval pipelines.

References

[1] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ... & Sutskever, I. Zero-shot text-to-image generation. In International conference on machine learning 2021, pp. 8821-8831.

[2] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. Hierarchical text-conditional image generation with clip latents 2022, 1(2), 3. https://doi.org/10.48550/arXiv.2204.06125

[3] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition 2009, pp. 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[4] Thyagharajan, K. K., & Kalaiarasi, G. A review on near-duplicate detection of images using computer vision techniques. Archives of Computational Methods in Engineering 2021, 28(3), pp. 897-916. https://doi.org/10.1007/s11831-020-09422-6

[5] Kaur, G., & Devgan, M. S. Data deduplication methods: a review. International Journal of Information Technology and Computer Science 2017, 10, pp. 29-36. https://doi.org/10.5815/ijitcs.2017.10.03

[6] Islam, S. M., & Debnath, R. A comparative evaluation of feature extraction and similarity measurement methods for content-based image retrieval. International Journal of Image, Graphics and Signal Processing 2020, 10(6), 19. https://doi.org/10.5815/ijigsp.2020.06.03

[7] Bajaj, E. N., Gill, E. J. S., & Kumar, R. An approach for similarity matching and comparison in content-based image retrieval system. IJ Inf. Eng. Electron. Bus. 2015, pp. 48-54. https://doi.org/10.5815/ijieeb.2015.05.07

[8] Lowe, D. G. Distinctive image features from scale-invariant keypoints. International journal of computer vision 2004, 60, pp. 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[9] Chum, O., Philbin, J., Isard, M., & Zisserman, A. Scalable near identical image and shot detection. In Proceedings of the 6th ACM international conference on Image and video retrieval 2007, pp. 549-556. https://doi.org/10.1145/1282280.1282359

[10] Bay, H., Tuytelaars, T., & Van Gool, L. Surf: Speeded up robust features. In Computer Vision-ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9 2006, pp. 404-417. https://doi.org/10.1007/11744023_32

[11] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. ORB: An efficient alternative to SIFT or SURF. In 2011 International conference on computer vision 2011, pp. 2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

[12] Lei, Y., Zheng, L., & Huang, J. Geometric invariant features in the Radon transform domain for near-duplicate image detection. Pattern recognition 2014, 47(11), 3630-3640. https://doi.org/10.1016/j.patcog.2014.05.009

[13] Tang, Z., Yang, F., Huang, L., & Zhang, X. Robust image hashing with dominant DCT coefficients. Optik 2014, 125(18), pp. 5102-5107. https://doi.org/10.1016/j.ijleo.2014.04.079

[14] Jie, Z. A novel block-DCT and PCA based image perceptual hashing algorithm. arXiv preprint arXiv:1306.4079 2013. https://doi.org/10.48550/arXiv.1306.4079

[15] Zeng, J. A Novel Block-DCT and PCA Based Image Perceptual Hashing Algorithm. arXiv preprint arXiv:1306.4079 2013. https://doi.org/10.48550/arXiv.1306.4079

[16] Ke, Y., Sukthankar, R., & Huston, L. An efficient parts-based near-duplicate and sub-image retrieval system. In Proceedings of the 12th annual ACM international conference on Multimedia 2004, pp. 869-876. https://doi.org/10.1145/1027527.1027729

[17] Nian, F., Li, T., Wu, X., Gao, Q., & Li, F. Efficient near-duplicate image detection with a

local-based binary representation. Multimedia Tools and Applications 2016, 75, 2435-2452. https://doi.org/10.1007/s11042-015-2472-1

[18] Zhao, W. L., Ngo, C. W., Tan, H. K., & Wu, X. Near-duplicate keyframe identification with interest point matching and pattern learning. IEEE Transactions on Multimedia 2007, 9(5), 1037-1048. https://doi.org/10.1109/TMM.2007.898928

[19] Zhao, W. L., & Ngo, C. W. Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. IEEE Transactions on Image Processing 2009, 18(2), 412-423. https://doi.org/10.1109/TIP.2008.2008900

[20] Li, Y. A fast algorithm for near-duplicate image detection. In 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID) 2021, pp. 360-363).

[21] Chum, O., Philbin, J., & Zisserman, A. Near duplicate image detection: Min-hash and TF-IDF weighting. In Bmvc 2008, 810, pp. 812-815. https://doi.org/10.5244/C.22.50

[22] Liu, L., Lu, Y., & Suen, C. Y. Variable-length signature for near-duplicate image matching. IEEE Transactions on Image Processing 2015, 24(4), pp. 1282-1296. https://doi.org/10.1109/TIP.2015.2396208

[23] Xie, L., Tian, Q., Zhou, W., & Zhang, B. Fast and accurate near-duplicate image search with affinity propagation on the ImageWeb. Computer Vision and Image Understanding 2014, 124, 31-41. https://doi.org/10.1016/j.cviu.2013.12.011

[24] Zhou, Z., Lin, K., Cao, Y., Yang, C. N., & Liu, Y. Near-duplicate image detection system using coarse-to-fine matching scheme based on global and local CNN features. Mathematics 2020, 8(4), 644. https://doi.org/10.3390/math8040644

[25] Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., & Kompatsiaris, Y. Near-duplicate video retrieval by aggregating intermediate cnn layers. In MultiMedia Modeling: 23rd International Conference, MMM 2017, Reykjavik, Iceland, January 4-6, 2017, Proceedings, Part I 23 2017, pp. 251-263. https://doi.org/10.1007/978-3-319-51811-4_21

[26] Zhang, Y., Zhang, S., Li, Y., & Zhang, Y. Single-and cross-modality near duplicate image pairs detection via spatial transformer comparing CNN. Sensors 2021, 21(1), 255. https://doi.org/10.3390/s21010255

[27] Barz, B., & Denzler, J. Do we train on test data? purging cifar of near-duplicates. Journal of Imaging 2020, 6(6), 41. https://doi.org/10.3390/jimaging6060041

[28] Matatov, H., Naaman, M., & Amir, O. Dataset and case studies for visual near-duplicates detection in the context of social media 2022. https://doi.org/10.48550/arXiv.2203.07167

[29] Tralic, D., Zupancic, I., Grgic, S., & Grgic, M. CoMoFoD - New database for copy-move forgery detection. In Proceedings ELMAR-2013 2013, pp. 49-54.

[30] Morra, L., & Lamberti, F. Benchmarking unsupervised near-duplicate image detection. Expert Systems with Applications 2019, 135, pp. 313-326. https://doi.org/10.1016/j.eswa.2019.05.002

[31] Barz, B., & Denzler, J. Hierarchy-based image embeddings for semantic image retrieval. In 2019 IEEE winter conference on applications of computer vision (WACV) 2019, pp. 638-647. https://doi.org/10.1109/WACV.2019.00073

[32] Berman, M., Jégou, H., Vedaldi, A., Kokkinos, I., & Douze, M. Multigrain: a unified image embedding for classes and instances. arXiv preprint arXiv:1902.05509 2019. https://doi.org/10.48550/arXiv.1902.05509

[33] Yu, Z., Zheng, J., Lian, D., Zhou, Z., & Gao, S. Single-image piece-wise planar 3d reconstruction via associative embedding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, pp. 1029-1037. https://doi.org/10.1109/CVPR.2019.00112

[34] Rau, A., Garcia-Hernando, G., Stoyanov, D., Brostow, G. J., & Turmukhambetov, D. Predicting visual overlap of images through interpretable non-metric box embeddings. In Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V 16 2020, pp. 629-646. https://doi.org/10.1007/978-3-030-58558-7_37

[35] Feng, G., Hu, Z., Zhang, L., & Lu, H. Encoder fusion network with co-attention embedding for referring image segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2021, pp. 15506-15515. https://doi.org/10.1109/CVPR46437.2021.01525

[36] Asadi-Aghbolaghi, M., Azad, R., Fathy, M., & Escalera, S. Multi-level context gating of embedded collective knowledge for medical image segmentation 2020. https://doi.org/10.48550/arXiv.2003.05056

[37] He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition 2016, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[38] Lytvynenko, T. I., Panchenko, T. V., & Redko, V. D. Sales forecasting using data mining methods. Вісник Київського національного університету імені Тараса Шевченка. Серія: Фізико-математичні науки 2015, 4, 148-155.

[39] Bieda, I., & Panchenko, T. A systematic mapping study on artificial intelligence tools used in video editing. International Journal of Computer Science & Network Security, 2022. 22(3), 312-318.

https://doi.org/10.22937/IJCSNS.2022.22.3.40
[40] Bieda, I., Kisil, A., & Panchenko, T. An approach to scene change detection. In 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) 2021, 1, pp. 489-493. https://doi.org/10.1109/IDAACS53288.2021.9660887
[41] Panchenko, T., & Bieda, I. A Comparison of scene change localization methods over the open video scene detection dataset. International Journal of Computer Science & Network Security 2022, 22(6), 1-6.
[42] Kubytskyi, V., & Panchenko, T. An Effective Approach to Image Embeddings for E-Commerce. In IT&I 2022, pp. 341-349.
[43] Fawzy, M., Tawfik, N. S., & Saleh, S. N. Enhancing Image Copy Detection through Dynamic Augmentation and Efficient Sampling with Minimal Data. Electronics 2024, 13(16), 3125. https://doi.org/10.3390/electronics13163125
[44] Chandrasiri, M. D. N., & Talagala, P. D. Cross-ViT: Cross-attention Vision Transformer for Image Duplicate Detection. In 2023 8th International Conference on Information Technology Research (ICITR) 2023, pp. 1-6. https://doi.org/10.1109/ICITR61062.2023.10382916
[45] Qin, Y., Ye, O., & Fu, Y. An automatic near-duplicate video data cleaning method based on a consistent feature hash ring. Electronics 2024, 13(8), 1522. https://doi.org/10.3390/electronics13081522
[46] Kubytskyi, V.; Panchenko, T. Enriched Image Embeddings as a Combined Outputs from Different Layers of CNN for Various Image Similarity Problems More Precise Solution. In Advances in Artificial Systems for Logistics Engineering III; Hu, Z., et al. (Eds.); Lecture Notes in Data Engineering and Communications Technologies; Springer: Cham, Switzerland, 2023; 180, pp. 1-13. https://doi.org/10.1007/978-3-031-36115-9_30
[47] Image Similarity Dataset: 10000 labelled image pairs. Kaggle 2025. Available online: https://www.kaggle.com/datasets/pantaras/near-duplicate-images/data

Performance Assessment of LSTM Networks for Short-Term Load Forecasting Based on Temporal Feature Engineering

Yin Zheng
School of Software Engineering, Jilin Technology College of Electronic Information
Jilin 132000, Jilin, China
E-mail: 13523838668@163.com

Keywords: short-term load forecasting (STLF), neural networks, long short-term memory (LSTM), machine learning (ML) models, energy demand forecasting

Received: May 20,2025

Accurate short-term load forecasting (STLF) is essential for maintaining grid stability, reducing operational costs, and optimizing energy dispatch within modern power systems. The article discusses Long Short-Term Memory (LSTM) neural network to forecast hourly load and compares the results of the neural network with four popular machine learning tools, namely Support Vector Regression (SVR), Gradient Boosting (GB), Extra Trees (ET), and Random Forest (RF). The relevant temporal features were also engineered using a large-scale, real-world dataset that comprised of over 145,000 hourly observations, and seasonal, weekly, and daily variations. All models were tuned using a structured optimization process and evaluated using standard error and correlation-based metrics. The empirical findings demonstrate that the LSTM model significantly outperforms all competing approaches, achieving an R^2 of 0.9977 and an error margin below 1% in terms of MAPE, representing a substantial improvement over ensemble- and kernel-based regressors. Visual diagnostic analysis further confirms the LSTM's ability to accurately reproduce peak and off-peak load behavior while maintaining low predictive dispersion. The results suggest that deep sequence-based models, when properly configured, provide exceptional advantages for STLF applications. The study concludes by outlining opportunities to improve performance through the integration of exogenous features and advanced hybrid architectures.

Povzetek: Študija kaže, da LSTM nevronska mreža bistveno izboljša kratkoročno napovedovanje porabe električne energije v primerjavi z drugimi metodami strojnega učenja.

1 Introduction

Energy consumption prediction has been among the most researched areas in the field of electrical engineering in recent times [1]. To add to this, electricity is generated in power plants by various technologies like gas turbines, hydro turbines, or solar photovoltaics. This electricity is then transmitted over vast distances through transmission lines and made available for commercial use [2]. The dynamic power consumption of end-users relies on various factors. Peak hours refer to the particular periods in the day when consumption normally exceeds other periods. Moreover, the consumption also varies depending on the days of the week, weekends, and further [3]. Similarly, population expansion has a significant impact, particularly in load forecasting. The supply business demands both short- and long-term predictions that span between a few minutes, hours and days ahead, and yearlong predictions (up to 20 years ahead) [4]. The advent of competitive energy markets has made temporary forecasts applicable [5], [6]. Over the recent past, many countries have pursued the privatization and deregulation of their power systems and electricity has been turned into a commodity which is sold and bought at the market price [7]. The supply business depends on load projections because they greatly affect price composition [1]. Load forecasting is not a simple task. The load series is multifaceted and has multiple layers of seasonality. The load at any given hour is determined by the load in the last hour, the load in the last day at the same hour and the load in the same hour of the same day in the last week. Besides, other major external factors that will be important to consider are mainly those of the weather conditions [8]. STLF is instrumental in maintaining grid stability and optimizing resource management, since its forecast of electrical demand load over short terms, usually minutes to days ahead-represents an important tool for this purpose [9]. Conventional methods, which consist of statistical models such as ARIMA and machine learning-based ones such as Support Vector Machines tend to be ineffective in case of modeling complex, nonlinear, time-varying trends [10].

A detailed overview of the related studies and their comparison, in terms of methodology, data properties, main contributions and findings, is presented in Table 1.

Table 1: Summary of related papers and their attributes

Study	Methodology / Model Type	Data Characteristics	Key Contribution / Strength	Key Findings / Performance Outcome
Wei et al. [11]	WM algorithm + transfer learning	Short-term load time-series	Reduces negative transfer risk	Outperformed LSTM, Informer, and Autoformer
Waheed & Xu [12]	Deep neural network + time-series and feature selection	Real-world load data	Integrates feature optimization	Higher accuracy than baseline ML methods
Sun et al. [13]	Graph Neural Network + dilated 1D-CNN (GLFN-TC)	Temporal-spatial load data	Learns variable relationships automatically	Lower MSE than baseline deep models
Liu et al. [14]	Fuzzy c-means + improved LSTM	Short-term load forecasting	Captures uncertainty & cluster-based learning	Higher prediction accuracy than standard LSTM
Asiri et al. [15]	Hybrid Deep Learning + Beluga Whale Optimization	FE & Dayton datasets	Hyper-parameter optimization via meta-heuristic	Achieved a minimal average error of 3.43 and 2.26
Lu et al. [16]	CNN + LSTM + NeuralProphet + Bayesian optimization	Electricity datasets	Hybrid ensemble for robustness	Significant improvement vs. traditional models
Smyl et al. [17]	Enhanced ES-dRNN w/ attention	Multiple benchmark datasets	Contextually-driven deep learning	Superior accuracy vs existing models
Hossain & Mahmood [18]	LSTM model	ERCOT hourly load + weather	Neural sequence-learning	Higher accuracy vs GRNN and ELM
Masood et al. [19]	Cluster-based probabilistic LSTM	Household-level consumption	Robustness improvement	Outperformed benchmark probabilistic models
Khayat et al [20]	ML model comparison (ANN vs LSTM)	Microgrid datasets	Historical data prediction	LSTM is superior on peak value accuracy
Su et al. [21]	Attention-based spatio-temporal GCN	Regional energy system	Integrates spatial relationships	MAPE improved by 5.6% vs peers
Wang et al. [22]	SSA-CNN-LSTM	Time-series load data	Advanced evolutionary hybrid	Errors significantly reduced vs traditional methods
Fan et al. [23]	EWTCNN-LSTM hybrid	Historical consumption data	Feature extraction statistics + Bayesian tuning	Achieved highly accurate load estimation

Regardless of the variety of studies examining advanced short-term load forecasting models, such as deep learning, hybrid neural networks, optimization-based models, and graph-based learning, two significant gaps remain in the literature. First, the majority of studies have concentrated more on architectural complexity and algorithmic improvement without exploring comprehensively how temporal feature engineering can be used to obtain predictive performance improvements, especially when historical consumption-based sequential learning is used. Second, although many hybrid models have been proposed, their performance is frequently obtained at the cost of model simplicity, computation costs, and reproducibility, and error measures in available literature tend to exceed the 1% MAPE. Thus, a completely reproducible, single-architecture deep learning solution with near-real-time generalization, reduction of forecasting error on interpretable sequential inputs, and validation of its superiority with statistically rigorous comparative performance is still desired.

The main purpose of this research is to train and test a streamlined Long Short-Term Memory (LSTM) neural network to attain a high degree of accuracy in short-term electrical load prediction through temporal feature engineering and past consumption indicators. To attain this, the current study aims to (1) establish and preprocess the large-scale hourly load dataset, (2) design the LSTM-based forecasting architecture that can learn the multi-scale temporal dependencies, (3) compare its results with the state-of-the-art machine learning models (SVR,

Gradient Boosting, Extra Trees, and Random Forest) in terms of standard statistical accuracy and dispersion measures, and (4) demonstrate the ability of the proposed method to make sound generalization decisions based on diagnostic visualization and statistical hypothesis testing.

2 Methodology

The significant novelty in this work is the employment of the LSTM algorithm for the improvement of short-term load forecasting. The utilized dataset for modeling includes load consumption data that have been meticulously collected and organized, enabling precise training and evaluation of the load forecasting models [18]. The dataset comprises hourly electrical load observations of the AEP area in the PJM network with 145,367 observations between 31 December 2002 and 2 January 2018 [24]. It has two primary columns: the date (Datetime) and the load (PJME MW) respectively. According to the timestamp, there were numerical time-related characteristics, including hour (0-23), day of the week (0-6), and month of the year (1-12) to reflect the intra-day, weekly and seasonal consumption patterns. Data has been sorted out chronologically, and any missing or duplicated records were eliminated to ensure uniformity. MinMaxScaler was used to normalize all of the features, including the derived temporal variables, to the range [0, 1] so that the magnitude of inputs is consistent across models which is also a critical step when working with an LSTM or classical regressors. This was followed by a sequential division of the dataset into 80% of training data and 20% of test data to avoid data leakage and maintain the temporal structure. This architecture gives a clean and well-organized time-series basis of short-term load prediction, allowing models to capture cyclical demand behavior and long-term dependency trends required in accurate prediction. Table 2 provides a summary of the dataset details.

Periodically, data is gathered on an hourly or daily basis and used to train and evaluate machine learning models to predict load accurately (Table 2). Python (version 3.10.11) was used as the experimental workflow, and the numerical operations and data preprocessing were run with NumPy (2.1.3) and Pandas (2.3.2). TensorFlow (2.19.0) was used in developing and training the model, and Scikit-learn (1.7.1) facilitated feature transformation, data partition, and benchmarking with classical machine learning models. Matplotlib (3.10.3) was used to generate visualization and diagnostic plots. All the computations were carried out on a Windows 10 operating system (build 10.0.19045), with an Intel 64-bit processor (Family 6, Model 158, Stepping 9) and four logical computing cores, without using any GPU acceleration.

Table 2: A summary of the details of the dataset

Category	Details
Artificial Intelligence	The dataset is utilized to train AI models, particularly focusing on deep learning algorithms like LSTM.
Machine Learning	Employed in various ML techniques for regression analysis to predict future load based on historical data.
Power and Energy	Contains crucial information on power consumption, aiding in efficient energy management and distribution.
Smart Grid	Supports smart grid applications by providing data to forecast demand and optimize grid operations.

The methodology section involved comparison of different machine learning models, such as, Support Vector Regression (SVR), Gradient Boosting (GB), Random Forest (RF), and Extra Trees (ET) against the chosen LSTM approach. The performance of these models was assessed using standard evaluation metrics and visualized through plot fit and plot regression charts for both training and testing phases of each method. These charts are effective ways to illustrate the capability and accuracy of the models when it comes to short-term load forecasting. The results of this comparison indicate that LSTM model is more efficient in predicting load accurately hence it is more effective as compared to other machine learning methods.

Table 3 summarizes the statistics for all performance measures related to forecasting models: MAE and RMSE yield an idea of the average error of prediction, Pearson's Correlation Coefficient (R^2) estimates the linear relationship between predicted and real value, Max Error gives the maximum deviation inside predictions, while Std represents the standard deviation of prediction errors, providing a general overview of the model's accuracy, stability, and reliability.

Table 3: The mathematical equation related to R2, RMSE, MAE, MAPE, EV, TIC, Std, and Max error metrics.

Performance metric	formula
Mean absolute error	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert$
Coefficient of Pearson's correlation	$R^2 = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}}$
Maximum of error	$Max\ error = \max(\lvert y_i - \hat{y}_i\rvert)$
Standard deviation	$Std = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}$
Theory – implied correlation	$TIC = \frac{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_t - y_t)^2}}{\sqrt{\frac{1}{n}\sum_{t=1}^{n} y_t^2}}$
Root mean square of error	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
Mean absolute percent error of n forecasting results	$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$
Enterprise Value	$EV = 1 - \frac{Var(y - \hat{y})}{Var\ (y)}$

3 Machine learning

Machine learning, mainly by the use of Long Short-Term Memory neural networks, is crucial in the forecast of short-term energy demand. LSTMs operate on time-series data by modeling complex temporal dependencies and patterns-much-needed for accurate energy demand forecasts. This leading capability of LSTMs enables them, by active learning from past consumption data, weather conditions, and calendar events, to outperform the general performance of traditional methods. Thus, LSTM-based models yield more reliable and precise predictions, with improved energy management and grid stability.

3.1 Support Vector Regression (SVR)

The machine learning model involved in predicting continuous values is Support Vector Regression or SVR. It extends the principles of SVM, usually applied to classification problems, to regression ones: the aim in SVR is to find a function approximating the relation between input features and the target variable so that the prediction error is minimized within a certain tolerance called the epsilon margin. It tries to fit as many data points as possible, at this margin with penalty on those that are outside of it. Such methods as the use of kernel functions enable SVR to work with nonlinear relationships effectively and make it an effective tool to address complex regression problems.

SVR is used in STLF (short-term load forecasting) to predict electricity demand in the near future based on the analysis of the trend of past load data along with other relevant parameters, which can be obtained from weather status, time of day, and occurrence of special events. The support vector regression (SVR) algorithms are used to extrapolate complex and non-linear relationships between the input features by mapping them into high-dimensional spaces. Thus, SVR models can effectively generate a more accurate and robust forecast of power consumption. Precision in grid management needs accuracy; thus, it would give a tailored electrical supply while optimizing generation with a minimum expense for operation. SVR would enable utilities to further optimize the overall energy efficiency while building up better forecasting to attain peak periods of demand.

3.2 Gradient boosting

Gradient Boosting is a sophisticated ensemble learning method, which addresses regression and classification problems where successively built models strive to enhance the accuracy of prediction. Such a technique combines several weak models, usually decision trees, into their strong predictive model configuration. The base model will initiate this process and will make some predictions over the input data. Further models are then trained on the errors made by previous models, and their predictions have a major focus on the residuals between the actual and predicted values. Each new model contributes to the improvement of the overall prediction; therefore, the final output is the weighted average of all the individual models. It is the process of iterative refinement that allows Gradient Boosting to gain knowledge of complex patterns in the data.

In this regard, Gradient Boosting will be used in short-term load forecasting to forecast future electrical consumption. This would be determined by historical load data and weather conditions, time of day and special events. Gradient Boosting does an effective job of uncovering complicated nonlinear correlations and variable interactions by progressively constructing and

combining numerous weak predictive models. iterative refinement thus makes the forecasts more accurate and robust, and therefore gives utilities a chance to estimate their changes in demand more accurately. Gradient Boosting thus ensures optimization of electricity generation and distribution, reducing the cost of operation and increasing the stability of the grid, owing to the fact that it can predict the short-term load precisely.

3.3 Extra trees

Extra Trees is an ensemble learning methodology that aims at enhancing the predictive accuracy of multiple decision trees. Extra Trees uses the same approach of traditional decision trees but uses more randomness in constructing the trees and in selecting the split points. Each tree is built from random subsets of features and thresholds. This increases the variability between trees and reduces overfitting. In the case of a regression problem, the overall prediction of the model is simply an average of the outputs of all the trees; for classification problems, it depends on a majority vote. This can be far more accurate and more efficient in computation, and Extra Trees can be a highly effective and resilient method of taking on general machine learning tasks.

Extra Trees can be used in short-term load forecasting to predict future energy consumption. This efficiency is achieved by using past consumption information against a situational parameter, such as weather, time of day, and events. This model, Extra Trees, does not require an expert to model complex patterns and relationships within the data because multiple decision trees with a high degree of randomization in feature selection and splitting thresholds are created. The diversity of the type of trees will also enhance the accuracy and reliability of the forecasts, besides lowering the overfitting risk. Extra Trees have been established to offer utilities with precise short-term load forecasts that enable in enhancing energy production, maintaining grid stability, and minimizing the cost of operation due to proper demand forecasts.

3.4 Random Forest

Random Forest (RF) is an ensemble machine learning algorithm that is used to solve both regression and classification problems, and that works on the idea of constructing various independent decision trees and combining their outputs to create a more accurate and stable prediction. RF takes long-term load use history and environmental conditions as inputs in predicting short-term loads, and in the latter scenario, additional contextual variables like time of day, weather, etc., are employed to model complex nonlinear load demand curves. The ensemble trees have been trained on a random sample of both the dataset and feature space and they therefore reduce the variance of an individual decision tree. The last forecast is the average of the total outcomes of all the trees leading to higher robustness and less over-fitting. RF is very scalable and resilient to outliers as well as nonlinear interaction without feature engineering. Applied to STLF, the algorithm improves the reliability of the forecasting by combining diversified structural trees, which allows forecasting unseen data more effectively, but at the same time, the algorithm remains computationally efficient. Its greater interpretability, by extracting the importance of features, gives great insight to planners of power systems and grid operators.

3.5 Long Short-Term Memory (LSTM)

Long Short-term Memory (LSTM) networks are an enhanced type of a recurrent neural network (RNN), which is designed to overcome the drawback of the previous RNNs that could learn long-term dependencies. LSTMs control information flow using memory cells, gating mechanisms, including input, output, and forget gates, to store the information about the relevant signals during long sequences and remove the noise. This makes them very effective in modelling complex time-varying patterns in tasks such as time-series forecasting, natural language processing, and speech recognition as presented in Fig. 1. LSTMs are especially efficient in short-term load prediction because of their high capacity to capture long-range dependencies and crop patterns of time-series information. Their ability to retain information in a long sequence is applicable in modeling the behavior of loads with history consumption, seasonality, weather, and variations in time of the day. Consequently, LSTMs offer more accurate short-term demand forecasts, which allow the utilities to predict the consumption, more efficiently distribute energy, and organize the grid.

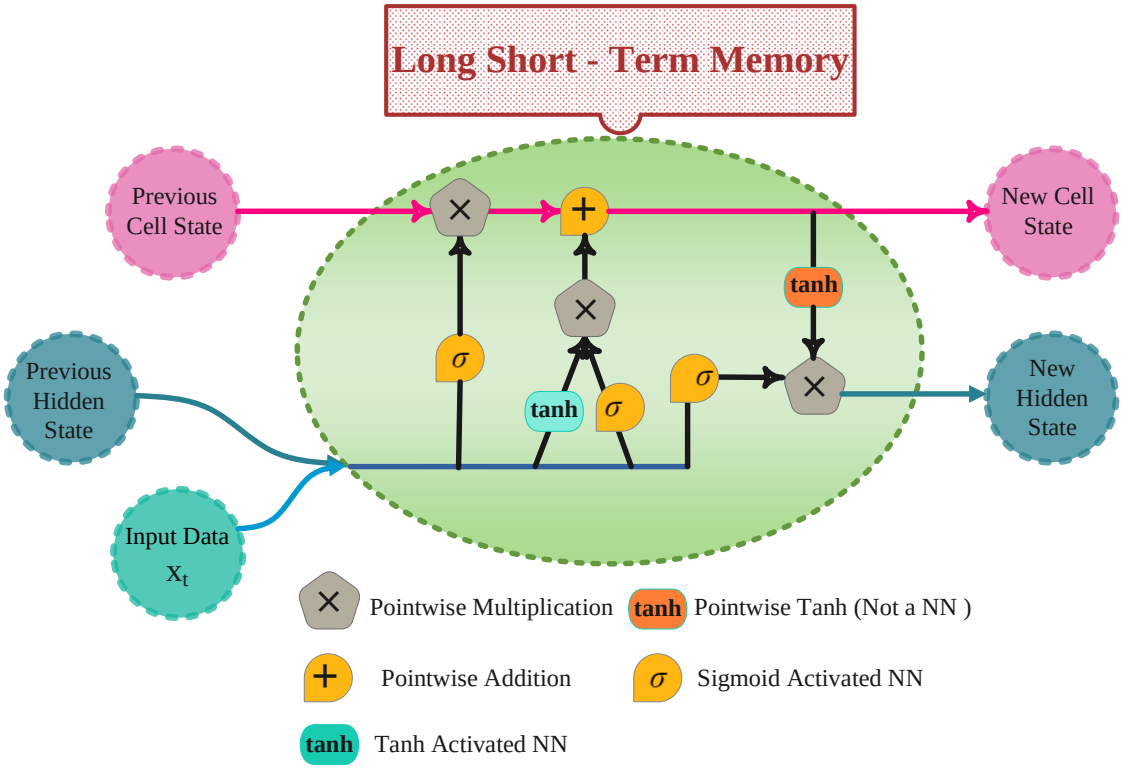


Figure 1: Structure of long-short-term memory

The full architecture and training setup of the LSTM network employed in this paper are presented below to ensure that the best-performing model can be replicated. The structure of the model involved two layers of LSTM, where the first layer has 128 units with return_sequences = True to allow sequence propagation, and the second layer has 64 units and return_sequences = False. A dropout rate 0.2 is used between the LSTM layers in order to avoid overfitting. The network continues with a Dense layer of 32 neurons with the ReLU activation function and a single-neuron linear output layer for regression.

This model was optimized using the Adam optimizer and a learning rate of 0.001 and a loss of mean squared error (MSE). The input sequence length was specified as 168-time steps (seven days), the batch size was 32 and the overall number of training epochs was 10. In order to enhance generalization and to eliminate overfitting, early stopping (patience = 10) and learning rate reduction (factor = 0.5, patience = 5) were implemented on the basis of validation loss.

4 Hyperparameter optimization

To determine the best model configuration, with respect to predictive performance and generalization, all models were hyperparameter-tuned with an exhaustive Grid Search strategy. In classical machine learning models, search grids contained the number of estimators, tree depth, and learning rate of Gradient Boosting; number of estimators and randomization controls of Extra Trees; no limits on the number of trees of Random Forest; penalty parameter (C), margin tolerance (ε), and bandwidth of the kernel (γ) for SVR. The LSTM hyperparameters were optimized based on network depth, units, learning rate, and dropout rate, with early stopping and adaptive learning-rate decay to avoid overfitting and stability convergence, as illustrated in Table 4. All preprocessing, such as scaling, was only fitted to the training set and applied uniformly to the test split to reduce the risk of data leakage.

Table 4: Hyperparameter optimization of ML models

Model	Hyperparameters
LSTM	Input window: 168 time steps (7 days); Features: load + hour + day of week + month; LSTM layer 1: 128 units, return sequences=True; Dropout: 0.2; LSTM layer 2: 64 units; Dense: 32 units, activation = 'relu'; Output: Dense(1), linear; Optimizer: Adam (lr=0.001); Loss: MSE; Batch size: 32; Max epochs: 10; Validation split: 0.1; Early Stopping (patience = 10, monitor = 'val loss', restore best weights = True); Reduce LR On Plateau (factor = 0.5, patience=5, monitor='val loss').
Gradient Boosting Regressor	n estimators=500; max depth=5; learning rate=0.03

Extra Trees Regressor	n estimators=500; random state=42; n jobs=-1
Random Forest Regressor	n estimators = 500; max depth= None; random state = 42; n jobs = -1;
SVR (RBF)	Kernel = 'rbf'; C=10; epsilon = 0.05; gamma = 'scale'.

5 Results

Results section provides a comparative analysis of various machine learning models used in short-term load forecasting with a special focus on the Long Short-Term Memory (LSTM) neural network. To guarantee the accuracy and reliability of the findings, the performance comparison will be conducted with the support of MAE, RMSE, Pearson correlation coefficient, Max Error, and Standard Deviation. As presented in Table 5, the quantitative findings demonstrate that there is a distinct performance difference between deep learning and classical tree-based and kernel-based regressors. Performance variations are low between the traditional models (GB, ET, RF, SVR), with ET and RF showing a slight advantage on training and testing data in terms of R 2, MAE, RMSE, MAPE, as well as the dispersion of residuals, which is appropriate to ensure a good balance of bias and variance. Their moderate level of generalization is manifested, however, by R^2 values in the range of 0.65 and fairly large MAE and RMSE values on unseen data. On the contrary, the LSTM model has significantly better predictive accuracy and stability. It has near-perfect predictive accuracy with an R^2 of over 0.998 on training and testing data and a reduction in MAE, RMSE, and residual standard deviation by over an order of magnitude over all other models. In addition, LSTM provides the lowest Theil U2 score, showing a better forecasting efficiency compared to the baseline models. Its strong ability to reduce the maximum error and percentage-based error measures further proves its strength and effectiveness in short-term load forecasting in extremely dynamic temporal settings.

Table 5: Results of model evaluation for short-term load forecasting.

Model	Dataset	R^2	MAE (MW)	RMSE (MW)	MAPE %	Max Error (MW)	Explained Variance	Theil's U2	Residual Std Dev (MW)
GB	Train	0.7425	2398.1546	10657462.9500	7.2495	18602.8625	0.7425	0.0496	3264.5910
	Test	0.6536	3048.3386	14587151.5600	9.9164	16925.1397	0.6922	0.0591	3600.2638
ET	Train	0.7441	2388.7142	10593567.5400	7.2179	18328.7368	0.7441	0.0495	3254.7901
	Test	0.6543	3042.7879	14555993.9500	9.8952	16572.5636	0.6929	0.0590	3596.1531
RF	Train	0.7441	2388.6542	10593940.6700	7.2175	18322.8437	0.7441	0.0495	3254.8473
	Test	0.6543	3042.5509	14554311.6100	9.8943	16560.6549	0.6928	0.0590	3596.2954
SVR	Train	0.7366	2441.1014	10903555.0700	7.3893	18986.9203	0.7367	0.0503	3301.6434
	Test	0.6557	3050.3049	14497628.2100	9.9461	17066.6997	0.6907	0.0590	3609.0133
LSTM	Train	0.9981	201.6500	79881.5986	0.6178	8276.2188	0.9981	0.0043	278.8473
	Test	0.9977	226.2666	95925.3630	0.7215	2961.6387	0.9977	0.0049	309.6872

Based on the R^2 plots (Fig. 2), there is an apparent disparity in model fidelity and generalization. The tree-based regressors (GB, ET, RF) cluster around moderate R^2, and have training performance of about 0.74 with test performance of about 0.65, which shows a reasonable fit but poor generalization. They have broader scatter patterns around the 1:1 line that indicate systematic under- and over-prediction, especially in higher-load areas. Conversely, the LSTM model demonstrates an almost flawless correlation between predicted and actual values of training and test sets, with R^2 values of over 0.997. The point clouds fold up closely to the diagonal, which implies that there is little residual variance and much stronger capturing of temporal dependences. This graphical and statistical data prove that the LSTM architecture internalizes the long-range structure in the load series better than the GB, ET, RF, or SVR, as their performance is limited by their inability to capture more complex sequential dynamics.

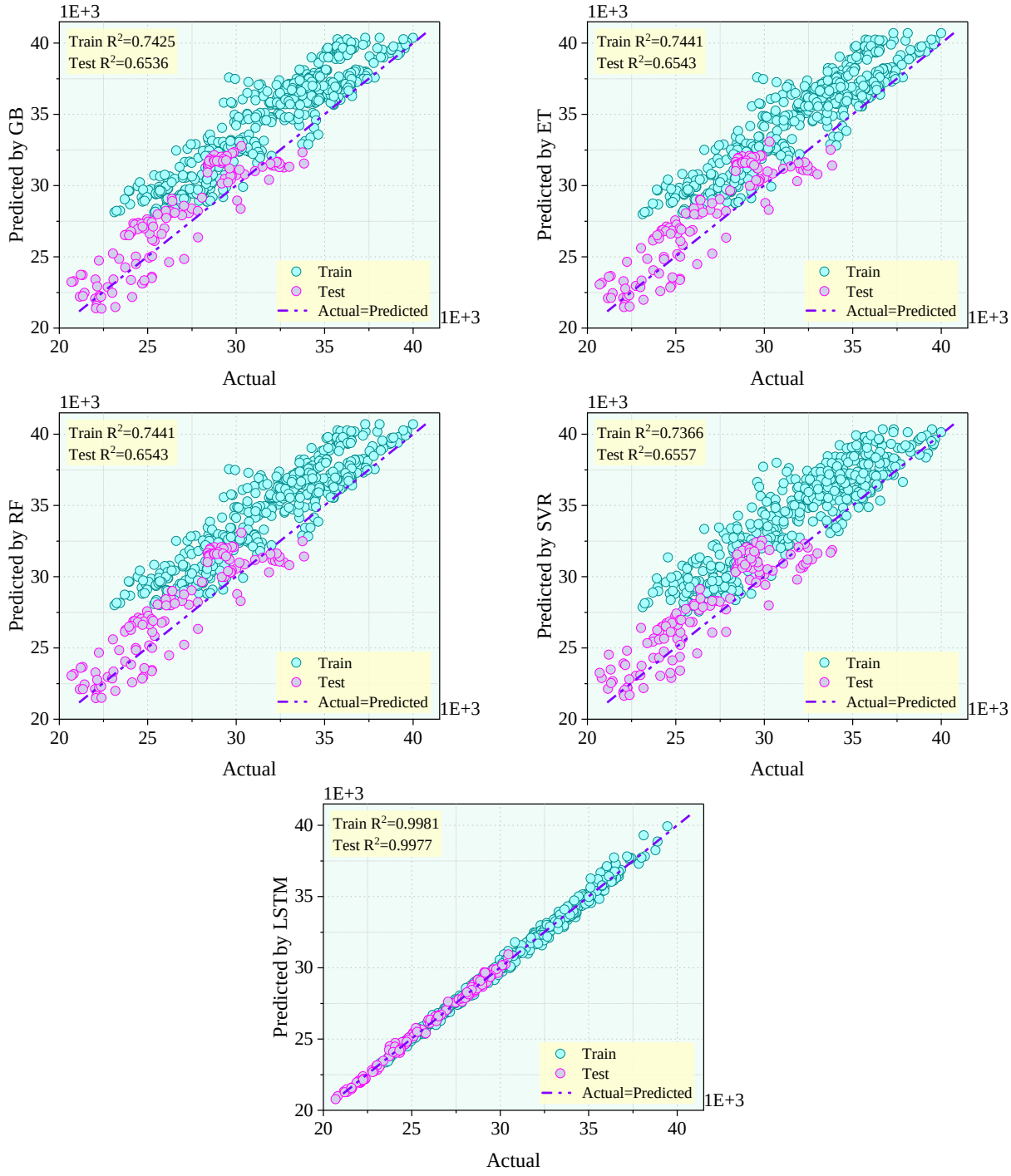


Figure 2: Comparison of the ML models based on R^2

The value plots (Fig. 3) indicate that all classical models tend to measure the seasonal and short-term load changes, but they show significant amplitude disparities and phase offset, especially in the test segment. GB and ET are prone to generate over-smoothed forecasts with less dynamic range, whereas RF and SVR are more predictive of short-term changes but remain characterized by systematic lag and error build-up when the load level is changed. However, the LSTM model is almost in agreement with the real signal both during the training and testing phases, capturing all the peaks and valleys and the transition dynamics of the signal. This shows that tree-based and kernel-based regressors are restricted in capturing the sequential structure, and thus, deep temporal representation allows LSTM to maintain high-frequency patterns and long-term dependencies, which cause reduced generalization when there is a shift in the load distribution.

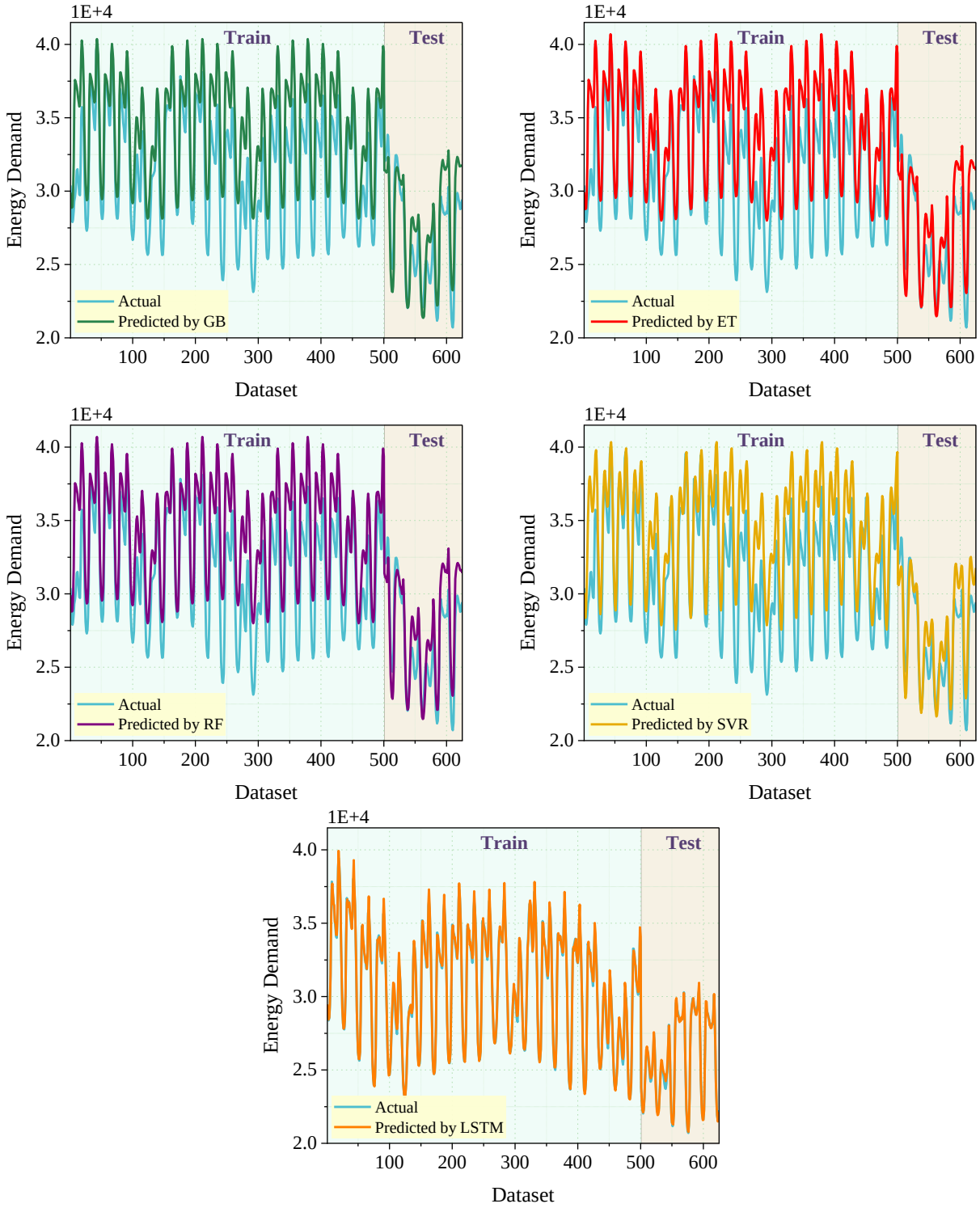


Figure 3: Comparison of ML models based on value plots.

The provided box plots in Fig. 4 reveal that interquartile ranges and central tendencies of GB, ET, RF, and SVR in training and testing sets are similar, which is a sign of moderate prediction spread and limited model divergence. Nevertheless, they all have several high-value outliers, indicating sensitivity to the peaks of demand and lower stability in non-stationary situations. LSTM has a smaller IQR and median spread, especially in the test set, which shows higher consistency and less variation. Even though LSTM tends to display more explicit outliers, the underlying distribution is still tight and concentrated, which proves that it has better generalization and better control over predictive dispersion than in classical models.

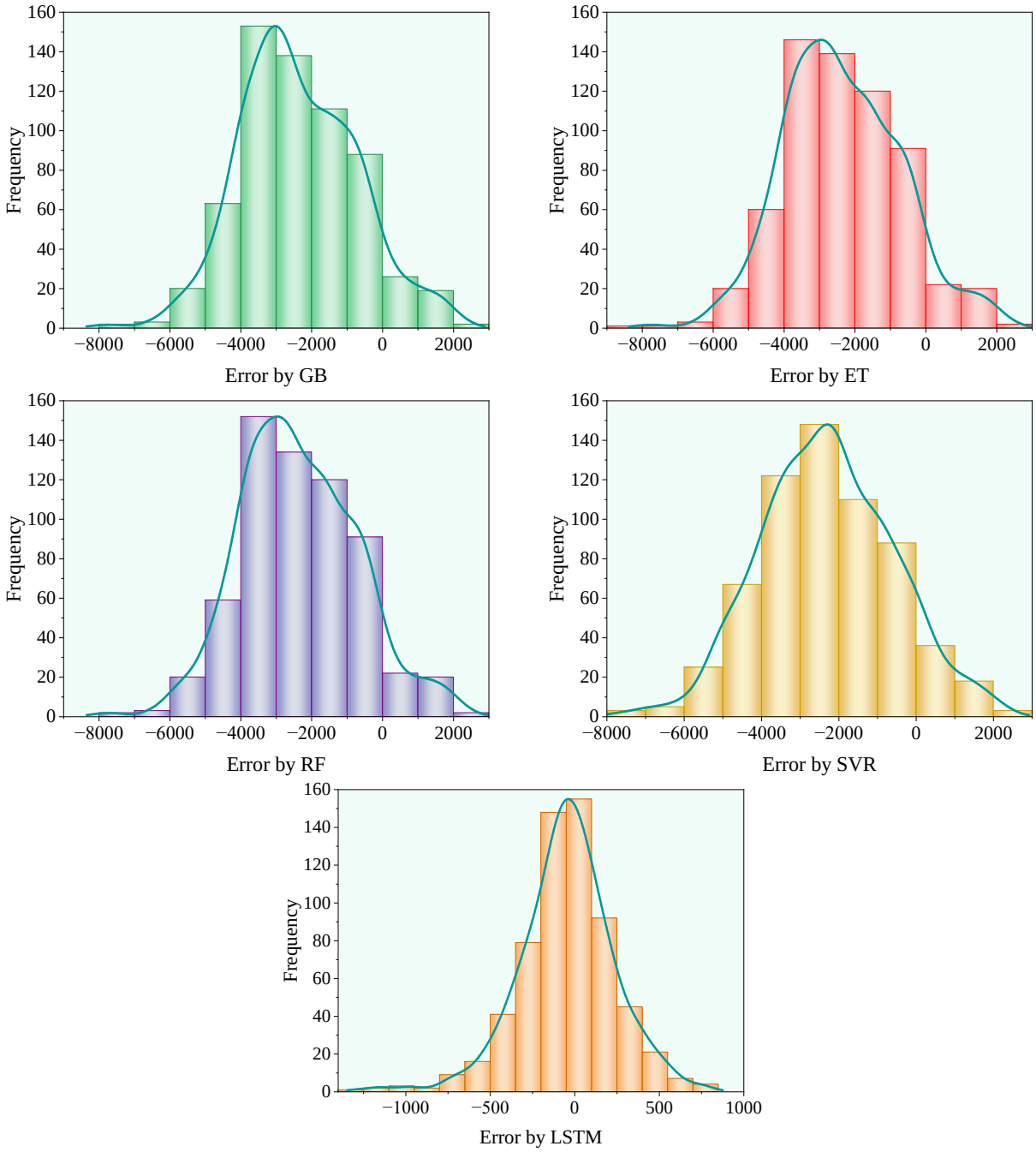


Figure 4: Comparison of the ML models based on Error plots

The box plots as provided in Fig. 5 show similar interquartile ranges and central tendencies for GB, ET, RF, and SVR in both training and testing sets, indicating moderate prediction spread with limited model divergence. Nevertheless, there are several high-value outliers in all of them, which are indicative of sensitivity to demand peaks and low stability in non-stationary conditions. LSTM displays a narrower IQR and lower median spread, particularly in the test set, demonstrating stronger consistency and reduced variability. Although LSTM shows more visible outliers, the core distribution remains compact and centered, confirming better generalization and superior control over predictive dispersion compared with classical models.

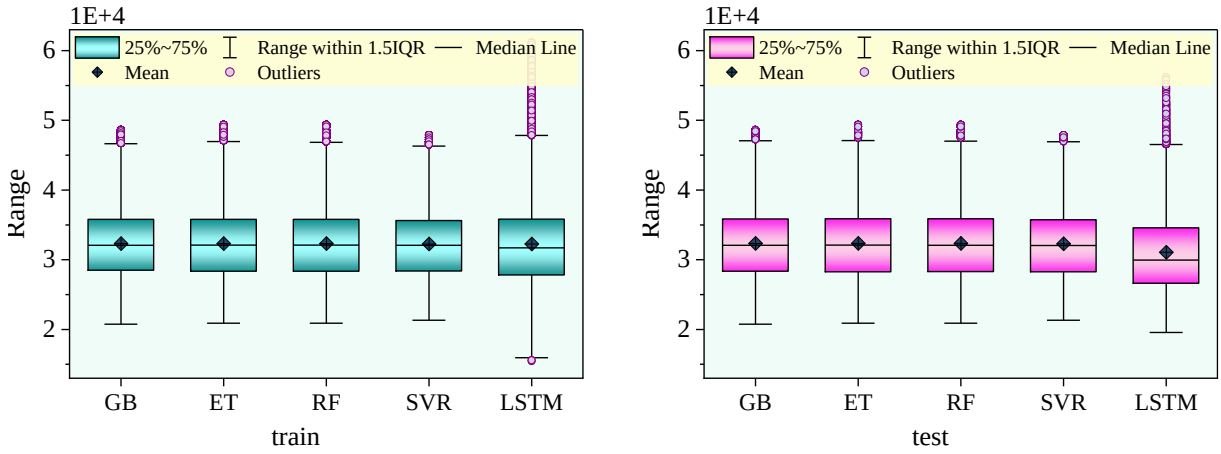
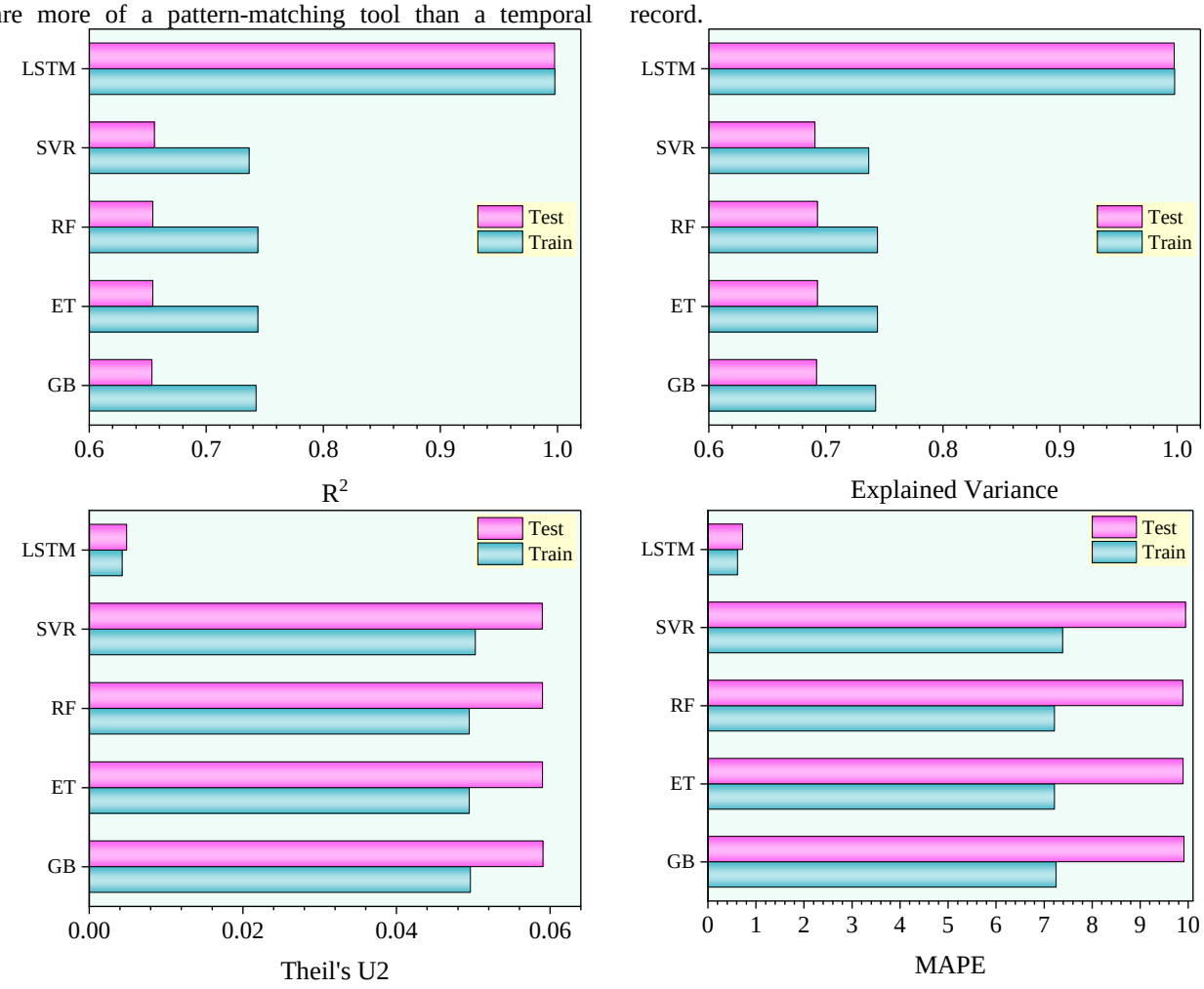


Figure 5: Comparison of the ML models based on Box plots for the train and test sets.

The comparison of metrics (Fig. 6) makes it clear that LSTM stands out as compared to the classical models. GB, ET, RF, and SVR all do reasonably well on the training data, but their test-set performance drops significantly indicating that they learn only a portion of the underlying demand dynamics. They are characterized by larger error values and lower explanatory metrics implying that they are more of a pattern-matching tool than a temporal learner. Rather, LSTM stands out with virtually similar training and testing results, implying that it does not learn the signal, but rather the structure of the signal. It can be easily adapted to new conditions because its errors are not only smaller but also more stable in both datasets. Practically, this implies that LSTM provides correct and reliable predictions instead of one that fit the historical record.



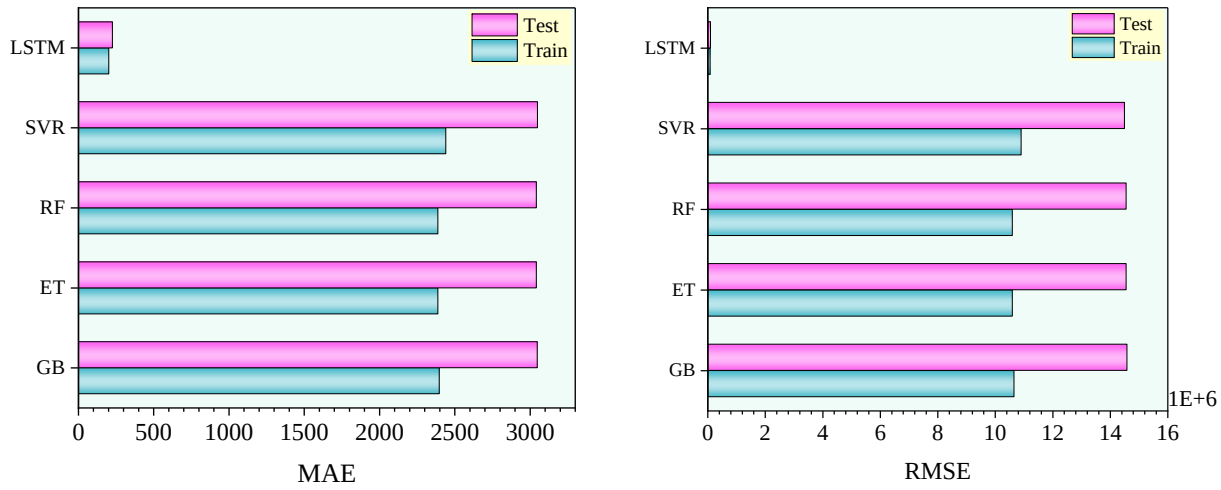


Figure 6: Comparison of the ML models based on metric plots

Fig. 7 represents the mean absolute SHAP plot. This plot measures the mean absolute SHAP contribution of each input variable to the output of the model. The normalized load and Month-scaled features have the biggest contributions at around 1.15×10^3, which means that the recent load values and the annual seasonality are the most influential factors in model predictions. Conversely, both hour-scaled and day-of-week features add no value, which indicates that weekly and intraday temporal variations do not add any significant predictive power after including load history and seasonal factors.

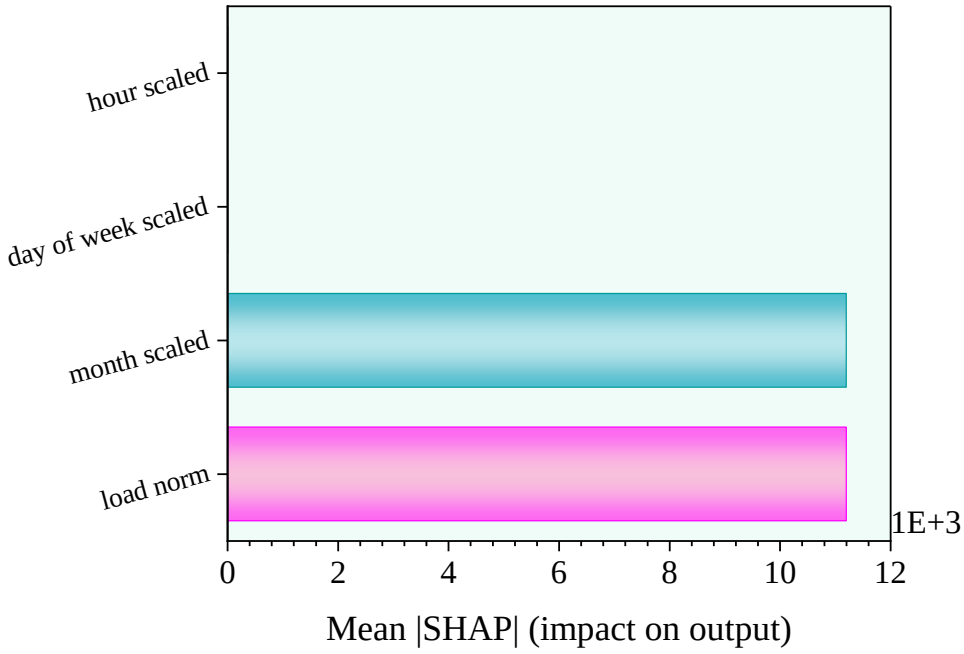


Figure 7: Mean absolute SHAP feature importance.

Table 6 presents the results of the non-parametric Wilcoxon signed-rank test of comparing the distribution of predictive errors of each pair of models. A p-value less than 0.05 implies statistically significant performance differences. Gradient Boosting (GB) and Extra Trees (ET) and Random Forest (RF) do not show a statistically significant difference among themselves ($p = 0.395$, 0.266, and 0.016, respectively); only ET-RF is slightly significant. Nevertheless, comparisons between LSTM and any standard ML model all have $p = 0$, indicating that the error distribution of LSTM is much different and better. Likewise, SVR also differs significantly with GB, ET, and RF (all $p \leq 1E\text{-}08$), which implies that it exhibits different error behavior than tree-based methods. In general, the test is able to confirm that LSTM shows statistically better performance than all peers, and the classical ensemble models behave in the same way, and there is no statistically significant separation.

Table 6: Pairwise Wilcoxon signed-rank test for model performance differences.

Model 1	Model 2	Wilcoxon Statistic	P-Value
Gradient Boosting	Extra Trees	1.23E+09	0.395318
Gradient Boosting	Random Forest	1.23E+09	0.265984
Gradient Boosting	SVR	1.2E+09	3.43E-12
Gradient Boosting	LSTM	5.34E+08	0
Extra Trees	Random Forest	1.23E+09	0.016233
Extra Trees	SVR	1.21E+09	2.44E-09
Extra Trees	LSTM	5.33E+08	0
Random Forest	SVR	1.21E+09	5.14E-09
Random Forest	LSTM	5.33E+08	0
SVR	LSTM	5.45E+08	0

6 Discussion

The LSTM model proposed in this study has significantly superior forecasting capabilities than all the traditional machine learning models proposed in this paper and the findings of the literature regarding the topic. In this study, the LSTM achieved MAE = 226.27, RMSE = 95,925.36, MAPE = 0.72%, and R^2 = 0.9977 on the test set, while the best-performing traditional model (ET) produced MAE = 3,042.79, RMSE = 14,555,993.95, MAPE = 9.90%, and R^2 = 0.6543, representing more than an order of magnitude reduction in error and a 53% improvement in explained variance.

The forecasting error is significantly lower than published results. As an example, Waheed et al. [12] found improvements in deep learning with MAPE over 3% and Liu et al. [14] reached MAPE under 5% and Asiri et al. [15] had an average error of 3.43% and 2.26% on two datasets. Likewise, ensemble and hybrid models like CNN-LSTM, or NeuralProphet-based models provided, reported MAPE scores of 2-7% [16], [17], [22], [23]. Despite the fact these models performed better than the classical approaches, none of them achieved error performance below 1% as compared to the model proposed in this research.

These findings suggest that a properly tuned LSTM with the relevant temporal feature engineering can be more effective than classical regressors as well as more intricate hybrid deep learning systems and provide superior accuracy, stability, and generalization.

7 Conclusion

This paper provides a comparative analysis of various machine learning algorithms to predict electricity loads in the short run, and in particular, a well-designed neural network based on the Long Short-Term Memory (LSTM) algorithm. The LSTM model was shown to have far greater predictive performance than Support Vector Regression (SVR), Gradient Boosting (GB), Extra Trees (ET), and Random Forest (RF) using a large-scale time-series dataset and standardized preprocessing processes. The performance metrics, such as MAE, RMSE, MAPE and R^2, have continuously shown that the traditional ensemble and kernel-based models performed moderately, with limited generalization and greater spread of residual on unseen data. Conversely, the LSTM model had a high R^2 of 0.9977, MAPE of less than 1%, and significantly lower values of absolute and squared error, which confirms its ability to effectively explain the temporal dependency and complex seasonal behavior.

The high fidelity of the LSTM to track both the peak demand variations and the baseline consumption trends was further confirmed by visual diagnostic results such regression fit plots, value-tracking curves, and error box-plots. Furthermore, the Wilcoxon signed-rank tests indicated statistically significant better performance of the LSTM model in comparison with all the other tested methods. These uniform empirical and statistical results underscore that deep sequence-learning structure along with proper feature engineering and hyperparameter optimization are significant improvements in the accuracy and stability of prediction.

Future studies may include exogenous factors (weather and pricing), hybrid or attention-based architecture, and probabilistic/ multi-horizon forecasting. These extensions may further enhance reliability of the operation and the decision-making of utilities, system operators, and smart-grid planners.

Funding

This work was supported by 2024 Jilin Province Vocational Education and Adult Education Teaching Reform Research Project: Practice and Exploration of Ideological and Political Education Reform in Higher Vocational Courses from the Perspective of Craftsmanship Spirit (2024ZCY426)

References

[1] Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on power systems*, 16(1): 44–55, 2001. https://doi.org/10.1109/59.910780

[2] B D Deebak and Fadi Al-Turjman. *Sustainable Networks in Smart Grid*. Academic Press, 2022.

[3] Joan Sebastian Caicedo-Vivas and Wilfredo Alfonso-Morales. Short-term load forecasting using an LSTM neural network for a grid operator.

Energies (Basel), 16(23): 7878, 2023. https://doi.org/10.3390/en16237878

[4] Yang-Seon Kim, Moon Keun Kim, Nuodi Fu, Jiying Liu, Junqi Wang, and Jelena Srebric. Investigating the Impact of Data Normalization Methods on Predicting Electricity Consumption in a Building Using different Artificial Neural Network Models. *Sustain Cities Soc*, 105570, 2024. https://doi.org/10.1016/j.scs.2024.105570

[5] Aneeque A Mir, Mohammed Alghassab, Kafait Ullah, Zafar A Khan, Yuehong Lu, and Muhammad Imran. A review of electricity demand forecasting in low and middle income countries: The demand determinants and horizons. *Sustainability*, 12(15): 5931, 2020. https://doi.org/10.3390/su12155931

[6] Margarita Spichakova, Juri Belikov, Kalvi Nõu, and Eduard Petlenkov. Feature engineering for short-term forecast of energy consumption. in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, IEEE, 2019, pp. 1–5. https://doi.org/10.1109/ISGTEurope.2019.8905698

[7] Mingdong Han and Lingyan Fan. A short-term energy consumption forecasting method for attention mechanisms based on spatio-temporal deep learning. *Computers and Electrical Engineering*, 114: 109063, 2024. https://doi.org/10.1016/j.compeleceng.2023.109063

[8] Yavuz Eren and İbrahim Küçükdemiral. A comprehensive review on deep learning approaches for short-term load forecasting. *Renewable and Sustainable Energy Reviews*, 189: 114031, 2024. https://doi.org/10.1016/j.rser.2023.114031

[9] Mona Ahamd Alghamdi, S Abdullah, and Mahmoud Ragab. Predicting Energy Consumption Using Stacked LSTM Snapshot Ensemble. *Big Data Mining and Analytics*, 7(2): 247–270, 2024. https://doi.org/10.26599/BDMA.2023.9020030

[10] George Gross and Francisco D Galiana. Short-term load forecasting. *Proceedings of the IEEE*, 75(12): 1558–1573, 1987. https://doi.org/10.1109/PROC.1987.13927

[11] Nan Wei, Chuang Yin, Lihua Yin, Jingyi Tan, Jinyuan Liu, Shouxi Wang, Weibiao Qiao, and Fanhua Zeng. Short-term load forecasting based on WM algorithm and transfer learning model. *Appl Energy*, 353: 122087, 2024. https://doi.org/10.1016/j.apenergy.2023.122087

[12] Waqar Waheed and Qingshan Xu. Data-driven short term load forecasting with deep neural networks: Unlocking insights for sustainable energy management. *Electric Power Systems Research*, 232: 110376, 2024. https://doi.org/10.1016/j.epsr.2024.110376

[13] Chenchen Sun, Yan Ning, Derong Shen, and Tiezheng Nie. Graph Neural Network-Based Short-Term Load Forecasting with Temporal Convolution. *Data Sci Eng*, 9(2): 113–132, 2024. https://doi.org/10.1007/s41019-023-00233-8

[14] Fu Liu, Tian Dong, Qiaoliang Liu, Yun Liu, and Shoutao Li. Combining fuzzy clustering and improved long short-term memory neural networks for short-term load forecasting. *Electric Power Systems Research*, 226: 109967, 2024. https://doi.org/10.1016/j.epsr.2023.109967

[15] Mashael M Asiri, Ghadah Aldehim, Faiz Abdullah Alotaibi, Mrim M Alnfiai, Mohammed Assiri, and Ahmed Mahmud. Short-term load forecasting in smart grids using hybrid deep learning. *IEEE Access*, 12: 23504–23513, 2024. https://doi.org/10.1109/ACCESS.2024.3358182

[16] Shuai Lu and Taotao Bao. Short-term electricity load forecasting based on NeuralProphet and CNN-LSTM. *IEEE Access*, 12: 76870–76879, 2024. https://doi.org/10.1109/ACCESS.2024.3407094

[17] Slawek Smyl, Grzegorz Dudek, and Paweł Pełka. Contextually enhanced ES-dRNN with dynamic attention for short-term load forecasting. *Neural Networks*, 169: 660–672, 2024. https://doi.org/10.1016/j.neunet.2023.11.017

[18] Mohammad Safayet Hossain and Hisham Mahmood. Short-term load forecasting using an LSTM neural network. in *2020 IEEE Power and Energy Conference at Illinois (PECI)*, IEEE, 2020, pp. 1–6. https://doi.org/10.1109/PECI48348.2020.9064654

[19] Zaki Masood, Rahma Gantassi, and Yonghoon Choi. Enhancing short-term electric load forecasting for households using quantile LSTM and clustering-based probabilistic approach. *IEEE Access*, 12: 77257–77268, 2024. https://doi.org/10.1109/ACCESS.2024.3406439

[20] Ahmed KHAYAT, Mohammed KISSAOUI, Lhoussaine BAHATTI, Abdelhadi RAIHANI, Khalid ERRAKKAS, and Youness ATIFI. Microgrid short-term electrical load forecasting using machine learning models. in *2024 4th International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, IEEE, 2024, pp. 1–7. https://doi.org/10.1109/IRASET60544.2024.10549191

[21] Zhonge Su, Guoqiang Zheng, Miaosen Hu, Lingrui Kong, and Guodong Wang. Short-term load forecasting of regional integrated energy system based on spatio-temporal convolutional graph neural network. *Electric Power Systems Research*, 232: 110427, 2024. https://doi.org/10.1016/j.epsr.2024.110427

[22] Yonggang Wang, Yue Hao, Biying Zhang, and Nannan Zhang. Short-term power load forecasting using SSA-CNN-LSTM method. *Systems Science & Control Engineering*, 12(1): 2343297, 2024. https://doi.org/10.1080/21642583.2024.2343297

[23] Guo-Feng Fan, Ying-Ying Han, Jin-Wei Li, Li-Ling Peng, Yi-Hsuan Yeh, and Wei-Chiang Hong. A hybrid model for deep learning short-term power load forecasting based on feature extraction statistics techniques. *Expert Syst Appl*, 238: 122012, 2024. https://doi.org/10.1016/j.eswa.2023.122012

[24] R. Huseyn, Energy Consumption Dataset. *Kaggle*. Retrieved November 19: 2024.

A Closed-Loop Financial Risk Prediction and Dynamic Control Framework Using Optimized Random Forests

Xuemei Guan[1], Guixian Tian[2*]
[1]School of Economics and Management, Yancheng Institute of Technology, Yancheng 224051, China
[2]College of Finance & Information, Ningbo University of Finance and Economics, Ningbo 315175, Zhejiang, China E-mail: guan8803@126.com, 18663510797@163.com

Keywords: enterprise financial risk, random forest, risk early warning, dynamic control, model robustness

Received: October 21, 2025

To address the frequent financial risk (FR) issues exposed by enterprises in the complex economic environment, this paper proposes an optimized model based on random forest (RF), which is used for the early warning and dynamic control of enterprise financial risk (EFR). In existing studies, traditional statistical methods rely excessively on linear assumptions and thus struggle to capture the nonlinear relationships between financial indicators. To tackle these problems, this paper establishes a closed-loop risk management framework of "prediction - classification - control - feedback". Systematic optimization of the RF is further performed, including hyperparameter tuning (e.g., number of trees, maximum depth, minimum samples for splitting, etc.), class weight setting to alleviate the imbalanced sample problem, and indicator selection based on feature importance combined with integration strategy adjustment. This thereby improves the generalization ability and robustness of the model. The optimized model works exceptionally well in terms of prediction accuracy and robustness according to experimental results. For example, on the University of California, Irvine Bankruptcy Prediction (UCI-BP) dataset, its accuracy rate is 0.91, precision rate is 0.88, specificity is 0.90, and Brier score is only 0.11, which is significantly better than the comparison models. Among the robustness indicators, the logarithmic loss is 0.27, Matthews Correlation Coefficient (MCC) is 0.78, Kappa coefficient is 0.75, and stability index is 0.89, all demonstrating stronger robustness. In terms of dynamic control effect, based on the risk probabilities output by the model, this paper classifies enterprises into different risk levels and constructs a strategy mapping rule base of "risk level-control measures". By simulating the implementation of intervention methods such as cash flow optimization and liability structure adjustment, the changes in risk distribution and the transfer rate of high-risk enterprises are evaluated in combination with feedback data. The risk classification accuracy rates of the optimized model on three datasets (UCI-BP, Kaggle Company Bankruptcy Prediction (Kaggle-BP) and Kaggle Corporate Credit Rating (Kaggle-CCR)) are 0.91, 0.89 and 0.88 respectively. Moreover, the transfer rates of high-risk enterprises reach 0.34, 0.32 and 0.30 respectively. According to these findings, the model can successfully assist in the execution of control techniques and precisely identify possible hazards, enabling businesses to take preventative action before crises arise. As a result, this paper offers a fresh approach and useful resource for EFR early warning and control research. It enhances the theoretical framework of risk prediction models and raises the usefulness of risk management in real-world applications.

Povzetek: Članek predstavlja izboljšan model za zgodnje prepoznavanje in obvladovanje finančnih tveganj podjetij, ki omogoča boljše napovedi ter učinkovitejše upravljanje tveganj.

1 Introduction

The problem of corporate financial risk (FR) has steadily grown to be a significant element influencing the survival and long-term growth of businesses against the backdrop of the world economy's growing complexity and degree of uncertainty [1,2]. In recent years, with the frequent occurrence of macroeconomic fluctuations, industrial structure adjustments, and external environmental shocks, problems faced by enterprises, such as increased financing difficulties, heavier debt burdens, and tighter liquidity, have continued to emerge [3]. This directly threatens the enterprises' own debt-servicing capacity and operational stability, and may trigger systemic risks through the industrial chain transmission effect, affecting the overall security of financial market operations and social economic performance. Traditional FR early warning methods are mostly based on linear statistical models. Their simplicity and interpretability are their main advantages, but they clearly fall short when it comes to handling complex interacting relationships and high-dimensional nonlinear features [4]. Machine learning

techniques have become increasingly used in the field of FR management as big data and artificial intelligence (AI) have grown. Among these methods, random forest (RF) has attracted attention due to its high accuracy and robustness in classification and prediction tasks. RF may successfully rank the significance of variables and lower the danger of model overfitting by combining different decision trees, offering FR early warning a fresh viewpoint [5,6]. However, "prediction" alone is not sufficient to meet the needs of corporate risk management. Corporate FR management requires early identification of potential risks, and must implement dynamic control and response strategies after risks are identified to reduce the possible losses caused by risks [7,8]. This requires further introducing a "dynamic control mechanism" on the basis of the early warning model, combining risk prediction with risk response, and constructing a closed-loop system covering "identification - early warning - control".

In summary, this paper focuses on the early warning and dynamic control of Enterprise Financial Risk (EFR), with the specific research objectives as follows:

(1) Construct a high-precision EFR prediction model based on RF.

(2) Design and integrate a dynamic control mechanism of "prediction-classification-control-feedback".

(3) Conduct comparison and supplementation of existing research from the technical and application levels, demonstrate the advantages of the proposed model in handling nonlinear relationships, improving interpretability, and enhancing dynamic response capabilities, and provide risk management tools and theoretical references with popularization value for enterprises, financial institutions, and regulatory authorities.

2 Related works

In the research on FR management, existing achievements can be roughly divided into three categories:

(1) FR early warning models based on traditional statistical methods;

(2) Risk prediction methods based on machine learning and AI;

(3) Dynamic control and closed-loop management frameworks oriented to risk response.

Although existing studies have achieved certain progress, they still generally face problems such as overly strong linear assumptions, insufficient model transparency, difficulty in handling data heterogeneity, and lack of dynamic response capabilities, as shown in Table 1.

Therefore, this paper proposes a FR early warning and dynamic control framework based on RF. The nonlinear fitting ability and feature importance mechanism of RF can make up for the problems of overly strong linearity and insufficient interpretability of traditional statistical models. The robustness of RF on medium-sized enterprise data can compensate for the limitations of deep learning models, such as strong dependence on large samples and susceptibility to overfitting. Constructing a dynamic closed-loop of "prediction-classification-control-feedback" based on RF early warning fills the gap in existing research that lacks executable risk response mechanisms.

Table 1: Related work arrangement

Research	Used model	Dataset	Main performance indicators	Limitations
[9]	Z-score and Logit regression	Low dimensional financial data	ACC and F1	Dependent linear hypothesis; Difficult to deal with nonlinearity; Poor performance in dynamic environment
[10]	Multiple discriminant analysis	Enterprise credit data	Classification accuracy	Require the normality and variance homogeneity of variables; The stability of the model is poor; Refractory heterogeneity
[11]	support vector machine	Credit risk data of small and medium-sized enterprises	ACC	Parameter sensitivity; Large-scale data training is inefficient; Limited practicality
[12]	artificial neural network	Enterprise financial data	AUC, ACC	Black box effect is serious; Weak interpretability; It is more difficult to be accepted by the supervision.
[13]	Threshold control method	Liquidity risk scenario	Control success rate	Mostly theoretical research; Lack of real enterprise data verification; The actual operability is weak.

3 Research method

3.1 Analysis of principle and applicability of RF algorithm

RF is a non-parametric classification and regression method based on ensemble learning. Its core principle lies in implementing ensemble prediction using multiple independent decision trees through the "Bagging + random feature selection" approach [14,15]. Compared with traditional statistical models and other machine learning methods, RF has the following advantages in the field of corporate FREW:

(1) Strong nonlinear fitting capability: FRs involve numerous financial and non-financial indicators, and the relationships among these indicators are often nonlinear. RF can flexibly capture the complex interactive relationships between high-dimensional variables without being constrained by linear assumptions [16,17].

(2) Good adaptability to high-dimensional data: Corporate financial data contains a large number of indicators, and these indicators are highly correlated with each other [18]. By means of random feature selection, RF reduces data redundancy, enabling the model to maintain good prediction accuracy even in high-dimensional scenarios.

(3) Strong anti-overfitting ability: RF reduces variance by integrating multiple decision trees, avoiding the overfitting problem of single models. This is particularly effective in scenarios where the volume of financial data samples is limited [19].

(4) Variable importance evaluation: RF can quantify and rank the importance of each variable. This helps identify key FR factors and provides corporate managers with decision-making basis with strong interpretability [20].

(5) High robustness and stability: When data contains noise, missing values, or outliers, RF can still maintain stable prediction performance. It is thus adaptable to the incompleteness that is commonly found in corporate financial data [21].

The application of RF in corporate FREW and dynamic control holds strong practical rationality and academic value. On one hand, corporate financial data includes traditional indicators such as balance sheets, income statements, and cash flow statements, and may involve non-financial indicators related to the market, governance, and industry. RF can integrate multi-source heterogeneous data, thereby improving the comprehensiveness of early warning. On the other hand, by continuously updating samples and features, the RF model can adjust prediction results in real time under a dynamic environment, providing enterprises with phased risk classification early warnings and meeting the needs of dynamic control.

3.2 The construction of EFR early warning model

By analyzing company financial and non-financial data, the corporate FREWM seeks to anticipate possible hazards and provide hierarchical prompts, giving businesses, financial institutions, and regulatory bodies a scientific foundation for decision-making. Taking RF as the core algorithm, this paper combines the financial indicator system and dynamic monitoring mechanism to construct an early warning framework covering "risk feature extraction - model training - result determination - risk classification". This model focuses on prediction accuracy, and emphasizes interpretability and dynamic adaptability, forming a complete closed-loop for risk identification. Its specific framework is shown in Figure 1:

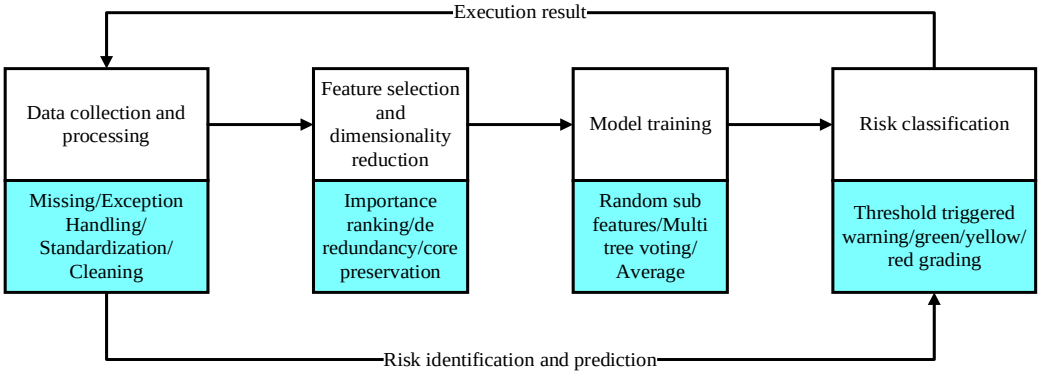


Figure 1: Framework for EFR warning model

The model construction process is as follows:

(1) Data preprocessing: To increase the efficacy of model training, handle outliers, standardize the original data, and imputation of missing values.

(2) Feature selection and dimensionality reduction: Utilize the feature importance evaluation function of RF to screen indicators with higher contribution to FR prediction, eliminate redundant or noisy variables, and optimize the computational efficiency of the model.

(3) Model training: Use RF to train sample data, generate multiple sub-sample sets through Bootstrap sampling, construct multiple decision trees, and perform

random feature selection during node splitting.

(4) Risk identification and prediction: Determine the risk level of newly input enterprise data, output classification results (such as "high risk - medium risk - low risk"), and provide corresponding probability distributions.

(5) Risk classification early warning mechanism: According to the model output results, classify enterprises into different risk levels and trigger corresponding early warning signals. For instance, the system immediately signals "high risk" and suggests appropriate control steps when the estimated likelihood surpasses the predetermined level.

Let the FR label of the ith enterprise be $y_i \in \{0,1\}$, then the risk scoring function of the enterprise (estimated as the probability of high risk) is defined as:

$$s(\mathrm{x}_i) = p(y_i = 1|\mathrm{x}_i) = \frac{1}{T}\sum_{t=1}^{T} h_t(\mathrm{x}_i) \quad (1)$$

T is the total number of decision trees, $h_t(\mathrm{x}_i)$ is the output of t-tree, and $s(\mathrm{x}_i)$ is the score of scoring function. In the model training stage, (x_i, y_i) is taken as the training sample, and the logarithmic loss function with class weighting is taken as the optimization objective to give consideration to the problem of class imbalance:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} [w_1 y_i \log s(\mathrm{x}_i) + w_0(1-y_i)\log(1 - s(\mathrm{x}_i))] \quad (2)$$

In the equation, N is the number of training samples. w_0 and w_1 are the weights, and $\mathcal{L}$ is the loss function. In the decision-making stage, in order to map the continuous risk score into three types of risk levels, two decision thresholds are introduced:

$$\text{RiskLevel}(\mathrm{x}) = \begin{cases} \text{Low}, & s(\mathrm{x}) < \tau_M \\ \text{Medium}, & \tau_M \le s(\mathrm{x}) < \tau_H \\ \text{High}, & s(\mathrm{x}) \ge \tau_H \end{cases} \quad (3)$$

τ_M and τ_H are obtained by grid search on the verification set, and the optimization objective is to maximize the F1 value or Youden index of high-risk identification to balance the recall rate and false alarm rate.

In summary, the constructed corporate FREWM can identify the potential factors of FRs, and provide hierarchical early warnings for the future risk levels of enterprises, thereby offering a scientific basis for risk prevention and control.

3.3 Design of dynamic control mechanism and model integration

EFRs exhibit multi-dimensional, nonlinear, and time-evolving characteristics. Relying solely on static prediction results is insufficient to support enterprises' continuous risk intervention decisions. To address this, based on the RF-based FR early warning model, this paper introduces a dynamic control mechanism and integrates "risk prediction-risk grading-control strategy-effect feedback-model update" into a closed-loop operational management framework. The overall model consists of three core sub-modules, as detailed in Table 2:

Table 2: Model framework of EFR early warning and dynamic control

Submodel	Main input	Core method and processing logic	Main output
Risk prediction submodel (RF)	Enterprise financial indicators, non-financial information, historical labels, etc.	Based on the integrated learning and probability prediction of RF, the enterprise risk score and risk grade judgment are generated.	Enterprise risk probability and risk level (such as low risk, medium risk and high risk)
Control strategy submodel (dynamic control)	Risk level, enterprise scale, industry attributes, debt structure, etc.	Call the manually designed "risk level-control measures" rule base to match the differentiated intervention combination according to the level.	Collection of control schemes for a single enterprise (credit policy adjustment, monitoring frequency adjustment, etc.)
Feedback and iteration mechanism	Effect data after control implementation (cash flow, default result, rating change, etc.)	Evaluate the control effect, track the risk change trend, and trigger the update when the model performance declines or the environment changes obviously.	Configuration of updated training sample set, model parameters and rule base

In practical implementation, the system classifies enterprises into three levels (low risk, medium risk, and high risk) based on the risk probabilities output by the RF model. The control objectives and typical measures corresponding to different risk levels are shown in Table 3:

Table 3: Model framework of EFR early warning and dynamic control

Risk level/component	Judgment basis (qualitative description)	Main control target	Typical control measures and system behavior
Low risk	The risk probability given by the model is in a low range, and the overall situation of enterprise operation and debt repayment is stable.	Maintain steady operation and avoid excessive intervention and high monitoring cost.	Maintain the normal credit policy; Carry out financial monitoring at regular frequency; Pay attention to industry changes and appropriately optimize the structure of capital use.
Medium risk	The risk probability is at a medium level, and some indicators show a worsening trend or increased fluctuation.	Prevent further risk accumulation and control risk exposure.	Tighten the credit line or raise the entry threshold; Adjust the repayment plan; Improve the frequency of submission and monitoring; Increase management interviews, etc.
High risk	The risk probability is in a high range, the debt repayment pressure is obvious, the liquidity is tight or there is a default signal.	Quickly reduce the probability of default or bankruptcy and protect the safety of funds.	Reduce or freeze the credit line; Asking for additional guarantee or mortgage; Initiate a debt restructuring or bail-out plan; Evaluate the asset disposal or exit plan.
Feedback data collection	Dynamic monitoring after implementation of control measures	Evaluate the effect of the strategy, and provide the basis for the subsequent iteration.	Continuously record changes in cash flow, default, credit rating, etc., and form new training and evaluation samples.
Rules and model updating	According to the feedback results and environmental changes, stage evaluation is carried out.	Maintain the adaptability of early warning and control mechanism to dynamic environment.	When the model performance declines or the risk structure changes obviously, update the training data, retrain the model and adjust the content of the rule base.

It shows that this paper adopts a rule design method based on expert experience and industry norms. First, the risk prediction model provides quantifiable risk levels, and then the rule base offers operable control measures.

4 Experimental design

The datasets selected for the experiment are University of California, Irvine Bankruptcy Prediction (UCI-BP), Kaggle Company Bankruptcy Prediction (Kaggle-BP), and Kaggle Corporate Credit Rating (Kaggle-CCR). UCI-BP is provided by the University of California, Irvine (UCI) Machine Learning Repository, which contains 6 consecutive years of financial data of Taiwanese enterprises. Its goal is to predict whether an enterprise will go bankrupt in the future, making it highly suitable for constructing corporate FREWMs and testing the performance of RF under high-dimensional data. The dataset can be downloaded via https://archive.ics.uci.edu/dataset/572/taiwanese+bankruptcy+prediction. Kaggle-BP is sourced from Kaggle and is based on real corporate financial statements. Its objective is to predict whether an enterprise will go bankrupt, and it is often used in comparative experiments of machine learning models. It can be directly applied to train the RF and evaluate prediction accuracy. The dataset is available for download at https://www.kaggle.com/datasets/fedesoriano/company-bankruptcy-prediction. Kaggle-CCR includes corporate financial indicators and credit ratings, with ratings ranging from AAA to D, which reflect the credit risk level of enterprises. The dataset can be downloaded through https://www.kaggle.com/datasets/agewerc/corporate-credit-rating.

The experiments of this paper are conducted on a well-configured workstation. In terms of hardware: The

central processing unit (CPU) is an Intel Core i9-12900K, with a base clock speed of 3.2 GHz and 16 cores/24 threads, which ensures the efficient execution of complex computing tasks. The graphics processing unit (GPU) is an NVIDIA GeForce RTX 3090 with 24 GB of video memory, used to accelerate the model training and prediction processes. The memory is 64 GB DDR5, which guarantees stability in large-scale data processing. The storage device is a 2 TB NVMe solid-state drive (SSD), ensuring high-speed data reading and writing. In terms of software: The operating system is Ubuntu 22.04 LTS. The experiment platform is built on the Python 3.9 environment. The main dependent deep learning and machine learning libraries include scikit-learn 1.2.2, TensorFlow 2.11, and PyTorch 1.13. Meanwhile, NumPy, Pandas, and Matplotlib are used for data processing and visualization. This configuration can provide stable and efficient experimental environment support for the RF model and comparison models in this paper.

The comparison models selected in this paper are Graph Convolutional Network for Financial Distress Prediction (GCN-FD), Bidirectional Long Short-Term Memory with Attention Mechanism (BiLSTM-ATT), and Extreme Gradient Boosting with Dynamic Thresholding (XGBoost-DT). The parameter setting is shown in Appendix file.

5 Analysis of experimental results

5.1 Analysis of experimental results of prediction accuracy and robustness

Two aspects of the experiment are examined: robustness and prediction accuracy. Figure 2 displays the prediction accuracy comparing results.

The proposed optimized RF model outperforms the comparative models overall, according to the data in Figure 2a. In terms of the Accuracy metric: The results of the optimized model on the UCI-BP, Kaggle-BP, and Kaggle-CCR datasets are 0.91, 0.89, and 0.88 respectively, all higher than GCN-FD's 0.87. In Figure 2b. The values of the optimized model on the three datasets are 0.88, 0.86, and 0.85 in sequence, which are significantly better than BiLSTM-ATT's 0.80 on the UCI-BP dataset. In Figure 2c. The results of the optimized model on the UCI-BP, Kaggle-BP, and Kaggle-CCR datasets reach 0.90, 0.87, and 0.86 respectively, higher than XGBoost-DT's 0.82 on the Kaggle-CCR dataset. In Figure 2d. The optimized model achieves the lowest scores, which are 0.11, 0.14, and 0.15 respectively, indicating that its probability prediction accuracy is better than GCN-FD's 0.17. The comparison results of robustness are shown in Figure 3.

In the results of Figure 3a, the optimized model also maintains its advantages. In terms of the Log Loss metric: The results of the optimized model on the three datasets are 0.27, 0.30, and 0.32, which are significantly lower than BiLSTM-ATT's 0.40 on the Kaggle-BP dataset. Figure 3b. The values of the optimized model are 0.78, 0.74, and 0.72 respectively, higher than XGBoost-DT's 0.68 on the Kaggle-BP dataset. Figure 3c. The optimized model achieves 0.75, 0.71, and 0.69 on the three datasets, while GCN-FD only reaches 0.56 on the Kaggle-CCR dataset. In Figure 3d. The optimized model obtains 0.89, 0.86, and 0.85 respectively, all higher than BiLSTM-ATT's 0.80 on the UCI-BP dataset. Overall, the optimized model performs prominently across the eight metrics and three datasets, demonstrating stronger prediction accuracy and robustness. To further verify the reliability of the results, this paper conducts 10 repeated experiments, and calculates the mean ± standard deviation for all prediction indicators and robustness indicators. Meanwhile, to further confirm the stability of the results, 95% confidence intervals are also provided in this paper. All experiments are completed under the same data division ratio and identical training configurations, with the results shown in Table 4.

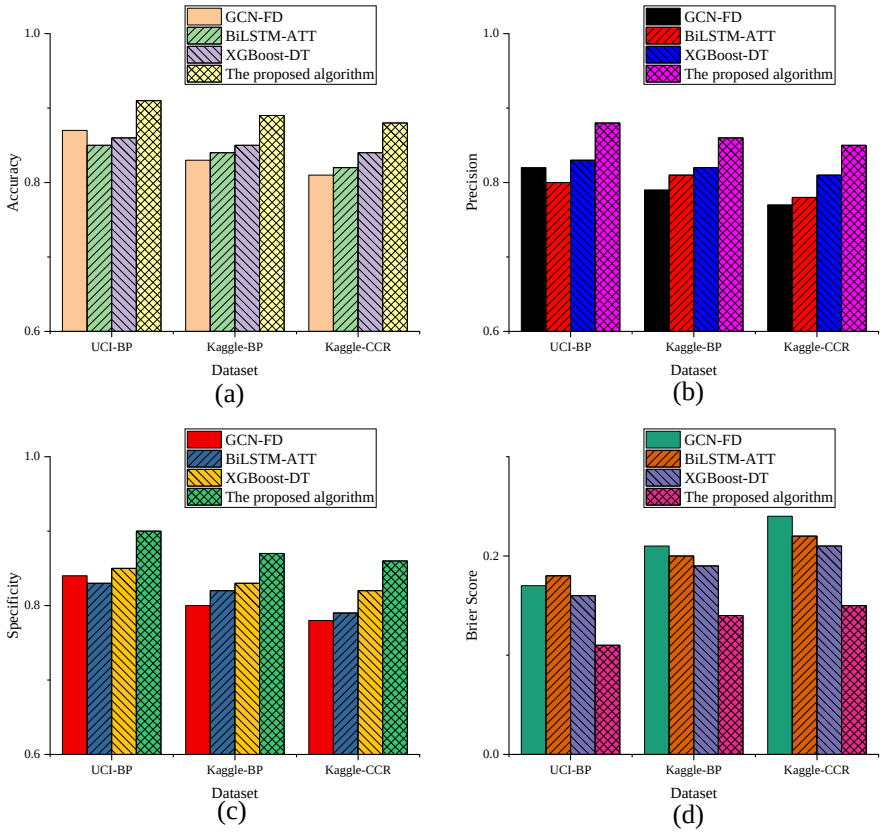


Figure 2: Comparison of experimental results on prediction accuracy(a) Accuracy (b) Precision (c) Specificity (d) Brier Score

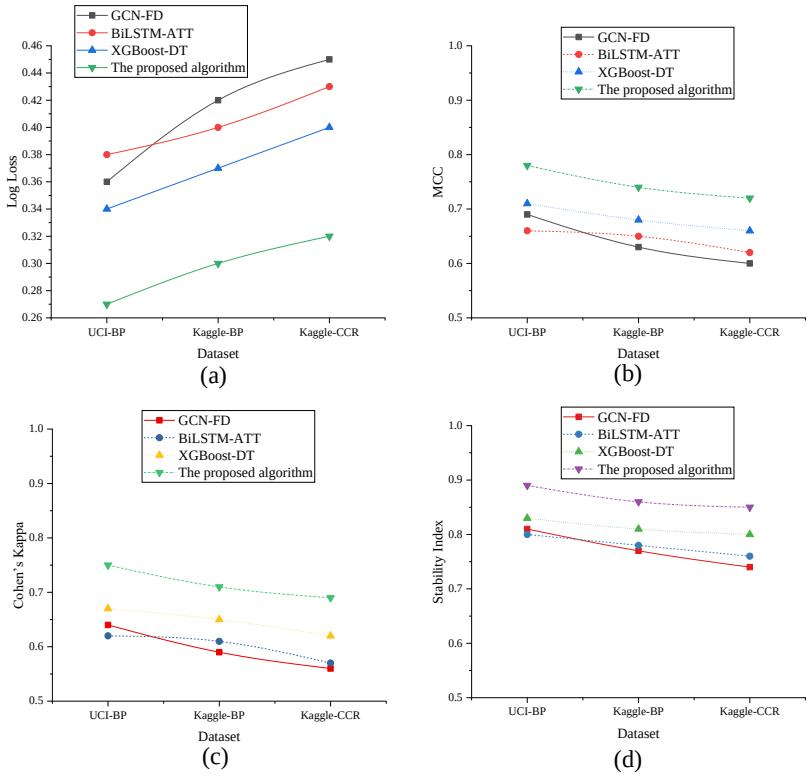


Figure 3: Comparison results of robustness (a) Log Loss (b) Matthews Correlation Coefficient (MCC) (c) Cohen's Kappa (d) Stability Index

Table 4: Model framework of EFR early warning and dynamic control

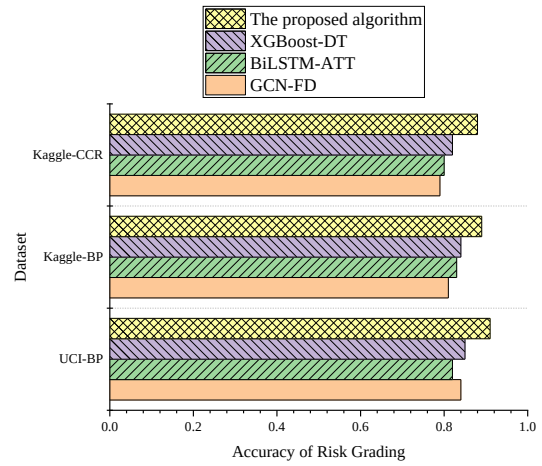
Indicator	Dataset	Optimized RF (mean ±SD)	95%CI	Optimal contrast model (mean±SD)
Accuracy	UCI-BP	0.910±0.006	[0.898,0.922]	0.873±0.011 (GCN-FD)
	Kaggle-BP	0.892±0.007	[0.878,0.905]	0.862±0.013 (XGBoost-DT)
	Kaggle-CCR	0.881±0.008	[0.866,0.896]	0.852±0.012 (GCN-FD)
Precision	UCI-BP	0.880±0.007	[0.865,0.894]	0.801±0.015 (BiLSTM-ATT)
	Kaggle-BP	0.861±0.009	[0.842,0.879]	0.824±0.014 (XGBoost-DT)
	Kaggle-CCR	0.847±0.010	[0.827,0.867]	0.818±0.016 (BiLSTM-ATT)
Specificity	UCI-BP	0.899±0.005	[0.889,0.910]	0.861±0.011 (GCN-FD)
	Kaggle-BP	0.874±0.006	[0.862,0.886]	0.832±0.012 (XGBoost-DT)
	Kaggle-CCR	0.862±0.007	[0.849,0.876]	0.822±0.013 (GCN-FD)
Brier Score	UCI-BP	0.112±0.004	[0.105,0.119]	0.171±0.007 (GCN-FD)
	Kaggle-BP	0.141±0.005	[0.131,0.151]	0.182±0.009 (XGBoost-DT)
	Kaggle-CCR	0.148±0.006	[0.136,0.159]	0.191±0.010 (BiLSTM-ATT)
Log Loss	UCI-BP	0.271±0.008	[0.255,0.286]	0.332±0.010 (BiLSTM-ATT)
	Kaggle-BP	0.301±0.009	[0.284,0.318]	0.402±0.012 (BiLSTM-ATT)
	Kaggle-CCR	0.318±0.010	[0.298,0.337]	0.389±0.011 (GCN-FD)
MCC	UCI-BP	0.782±0.006	[0.770,0.794]	0.701±0.013 (XGBoost-DT)
	Kaggle-BP	0.739±0.007	[0.725,0.753]	0.681±0.012 (XGBoost-DT)
	Kaggle-CCR	0.715±0.008	[0.699,0.732]	0.564±0.014 (GCN-FD)
Cohen’s Kappa	UCI-BP	0.748±0.005	[0.739,0.758]	0.692±0.012 (GCN-FD)
	Kaggle-BP	0.706±0.006	[0.693,0.719]	0.658±0.011 (XGBoost-DT)
	Kaggle-CCR	0.684±0.007	[0.671,0.698]	0.561±0.013 (BiLSTM-ATT)
Stability Index	UCI-BP	0.893±0.004	[0.886,0.901]	0.801±0.008 (BiLSTM-ATT)
	Kaggle-BP	0.861±0.005	[0.852,0.870]	0.798±0.010 (XGBoost-DT)
	Kaggle-CCR	0.851±0.006	[0.839,0.863]	0.780±0.009 (GCN-FD)

The results in Table 4 suggest that the optimized RF model exhibits low standard deviations (most < 0.01) across all eight indicators on the three types of datasets, indicating strong prediction stability of the model without significant random fluctuations. In addition, the 95% confidence intervals are all relatively narrow, which further confirms the robustness and reproducibility of the model under different training randomness conditions.

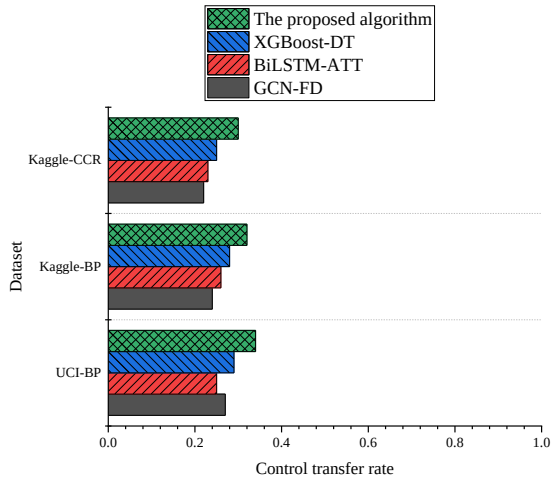
5.2 Experiment on dynamic control effect and risk classification

This paper confirms the efficiency of the improved model in dynamic control and risk classification after finishing the FR prediction.

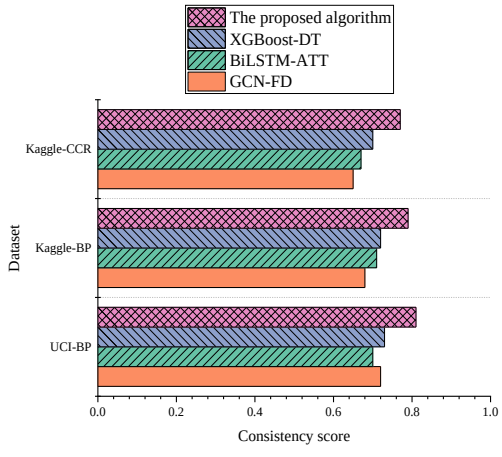
The precise procedure is as follows: Businesses are categorized into three risk groups based on the model's risk probability output: low risk (probability $\leq$ 0.3), medium risk (0.3-0.6), and high risk (probability $\geq$ 0.6). Under different levels, differentiated control strategies are triggered: Low-risk enterprises conduct regular monitoring. Medium-risk enterprises need to optimize cash flow and debt structure. High-risk enterprises simulate the implementation of measures such as capital injection and debt restructuring. To measure the effect of dynamic control and risk classification, this paper sets three evaluation dimensions, namely risk classification accuracy, the transfer rate of high-risk enterprises under control measures, and the consistency score between predicted distribution and actual distribution. The experimental results are shown in Figure 4:



(a)



(b)



(c)

Figure 4: Experimental results of dynamic control effect and risk grading (a) Accuracy of risk grading (b) Dynamic control transfer rate of high-risk enterprises (c) Reasonable consistency score of risk grading distribution

The results in Figure 4a show the following: In terms of risk classification accuracy, the proposed optimized model achieves results of 0.91, 0.89, and 0.88 on the UCI-BP, Kaggle-BP, and Kaggle-CCR datasets respectively, all higher than those of the comparison models. Specifically, GCN-FD scores 0.84 on UCI-BP, BiLSTM-ATT scores 0.83 on Kaggle-BP, and XGBoost-DT scores 0.82 on Kaggle-CCR, all of which are lower than the optimized model. In Figure 4b, the optimized model obtains values of 0.34, 0.32, and 0.30 on the three datasets, significantly higher than BiLSTM-ATT's 0.26 on Kaggle-BP and GCN-FD's 0.27 on UCI-BP. This indicates that the optimized model performs better in the risk mitigation process. In Figure 4c, the consistency scores of the optimized model on the UCI-BP, Kaggle-BP, and Kaggle-CCR datasets reach 0.81, 0.79, and 0.77 respectively, all outperforming the comparison models. Among the latter, GCN-FD only scores 0.65 on Kaggle-CCR, BiLSTM-ATT scores 0.70 on UCI-BP, and XGBoost-DT scores 0.72 on Kaggle-BP. In terms of risk categorization accuracy, dynamic control efficacy, and risk distribution consistency, the optimized model outperforms the others overall.

6 Discussion

Based on the comprehensive experimental results, the proposed optimized RF model overall outperforms comparative models such as GCN-FD, BiLSTM-ATT, and XGBoost-DT in tasks of EFR early warning and dynamic control, demonstrating favorable empirical effects. However, to truly enhance its applicability in complex financial environments, more in-depth discussions are required from multiple dimensions, including robustness mechanisms, application scenarios, sources of model advantages, and trade-offs.

From the perspective of model robustness, the RF itself, through Bagging and random feature subsampling mechanisms, possesses the effect of resisting uncertainty similar to that in "adaptive robust control" to a certain extent. Most decision trees learn independently on different subsamples and feature subspaces, enabling the model to buffer the impact of single abnormal samples or local structural changes on overall predictions. In the experiments of this paper, the significant decrease in Brier score and logarithmic loss indicates that the model is more stable in probability calibration, which is conducive to providing reliable inputs for risk grading and dynamic intervention. However, it should be emphasized that this study mainly conducts evaluations based on historical samples and normal distribution scenarios, and has not systematically covered extreme or unprecedented financial environments. For "tail risk" scenarios such as severe macroeconomic fluctuations and precipitous industry declines, the model may still face performance degradation. Future research will introduce the ideas of scenario simulation and stress testing. For example, it will construct extreme scenarios that include

sudden drops in returns, sharp contractions in liquidity, and violent fluctuations in interest rates or exchange rates. This approach will allow for a systematic evaluation of the model's prediction results, threshold stability, and risk distribution migration. The ultimate goal is to more comprehensively verify the model's stability and reliability under actual market shocks.

The proposed "prediction-classification-control-feedback" closed-loop framework has strong application potential and is expected to be implemented in various financial and practical scenarios. For example, in portfolio management, the model can continuously score the default risk and credit risk of target enterprises, mark high-risk enterprises as "reduced-holding objects" in the portfolio, and track the impact of intervention measures (such as position reduction and hedging tool allocation) on the portfolio loss rate and Sharpe ratio through the feedback loop. In enterprise credit monitoring and supply chain finance scenarios, the framework can be embedded into the risk control systems of banks or factoring institutions to implement hierarchical management of accredited enterprises, triggering credit limit adjustments, additional credit requirements, or early warnings for high-risk enterprises. In terms of ideological path, this framework has certain similarities with the adaptive robust control method in handling uncertainties of nonlinear systems: both emphasize dynamic feedback and strategy adjustment in uncertain environments. However, unlike traditional adaptive robust control which focuses on "stability proof of control laws", this paper focuses more on the comprehensive balance between "prediction accuracy and intervention effectiveness". Through the one-to-one mapping of risk levels and control strategies as well as the effect feedback retraining mechanism, the feedback loop is ultimately transformed into executable decision sequences for enterprises, such as cash flow optimization plans, capital structure adjustments, and asset disposal rhythm arrangements. In the future, further design and deployment plans can be developed at the information system level, such as embedding the model into enterprise ERP/financial sharing platforms or bank credit approval systems to realize automated early warning push and strategy recommendation, thereby enhancing the operability and implementation value of the framework.

In addition, it is necessary to more clearly discuss the potential reasons for the superior performance of the RF model. From the perspective of data characteristics, the UCI-BP, Kaggle-BP, and Kaggle-CCR datasets used in this paper exhibit features such as high dimensionality, significant nonlinear relationships between indicators, and obvious class imbalance. In this context, the RF, by integrating multiple nonlinear decision trees, can effectively characterize complex feature interaction structures and has stronger expressive power in terms of decision boundaries compared to linear logistic regression. Meanwhile, this paper adopts strategies such as class weight balancing and feature importance selection, which suppress the problems of minority class neglect and overfitting caused by redundant features in imbalanced samples and high-dimensional spaces. When compared with deep learning models such as Graph Convolutional Network (GCN) or Bidirectional Long Short-Term Memory (BiLSTM), on the one hand, the RF has relatively lower requirements for sample size and can fully exert its performance on medium-sized datasets. On the other hand, its model structure is relatively simple, and the training and inference processes are more likely to converge, these are also important reasons for its better results under the existing experimental settings. However, the model's superiority does not mean there are no trade-offs and costs. Compared with simpler logistic regression models, the interpretability of the RF is at a "moderate" level. Although it can provide partial explanations for managers through methods such as feature importance and decision path examples, its overall decision-making process still belongs to a "semi-black box", lacking the direct coefficient meaning and marginal effect analysis capability of linear models. For application scenarios that emphasize transparency and auditability (e.g., prudent supervision models in a strong regulatory environment), a balance needs to be struck between prediction performance and interpretability. Meanwhile, compared with deep learning models based on GCN or BiLSTM, the RF usually has advantages in computational cost: the training process can be completed mostly with CPU computing, with low dependence on GPU, and the model scale and inference time are more controllable, making it more suitable for deployment in scenarios with limited resources or the need for rapid iteration. However, deep learning models have inherent advantages in capturing temporal dependencies, network structure relationships, and multimodal information, which is also an important motivation for exploring the "RF + deep learning model fusion" in future research.

Finally, the discussion on robustness improvement needs to be further expanded in dimensions. Based on indicators such as Brier score and logarithmic loss, this paper quantitatively evaluates the stability of the model at the probability prediction level, which mainly reflects the "internal robustness" of the model under a given data distribution. However, in real financial environments, factors such as data drift, structural breakpoints, and institutional changes are likely to cause dynamic evolution of risk feature distributions. To this end, future research will strengthen robustness analysis from three directions: First, introduce concept drift and data drift detection mechanisms to monitor long-term changes in model input distribution and output quality, and promptly trigger model retraining or parameter updates. Second, systematically design model update frequency and rolling training strategies to reduce the risk of model "obsolescence" while balancing stability and adaptability. Third, real-time feedback information, such as cash flow improvements post-intervention or actual default events, is fed into incremental or online learning modules. This

allows the "prediction-classification-control-feedback" loop to evolve beyond a static strategy. It achieves genuine dynamic iteration by continuously updating parameters and adjusting its structure. Consequently, the model's long-term robustness and practical utility in realistic, dynamic business environments are significantly enhanced.

In summary, the proposed optimized RF model and its closed-loop framework have demonstrated favorable performance and application prospects in experimental results, but they only represent a starting point for the intelligent management of EFRs. Future work focusing on extreme scenario stress testing, multi-scenario application deployment, deep model fusion, and dynamic robustness enhancement will help further consolidate its theoretical foundation and expand its engineering implementation scope.

7 Conclusion

This paper focuses on corporate FREW and dynamic control, proposing an optimized model based on RF, and conducts empirical research on three public datasets. First, the paper constructs a multi-dimensional indicator system covering debt-servicing capacity, profitability, operational efficiency, cash flow, and non-financial factors. It then uses the RF algorithm to achieve high-precision prediction of corporate FR. Subsequently, combined with a dynamic control mechanism, a closed-loop framework of "prediction - classification - control - feedback" is further established. In terms of prediction accuracy and robustness, the experimental findings demonstrate that the optimized model performs better than the comparison models. It achieves the highest performance in metrics such as Accuracy, Precision, Specificity, Brier Score, Log Loss, MCC, Kappa Coefficient, and Stability Index. Meanwhile, in the experiments on dynamic control effect and risk classification, the optimized model can effectively identify high-risk enterprises, and significantly improve the efficiency of risk mitigation, while maintain a high consistency between the risk distribution and the actual situation. These results fully demonstrate that the proposed optimized model can provide scientific and reliable risk early warning and hierarchical management tools for enterprises, financial institutions, and government regulatory authorities, and has strong theoretical significance and practical value.

Nevertheless, this paper still has several limitations that require cautious consideration. First, the sample sizes of some experimental datasets are limited, and the RF may still have potential overfitting risks under small-sample conditions, so its generalization ability needs to be further verified on larger-scale enterprise financial data. Second, the RF is inherently a static model that lacks the ability to model temporal dependencies, making it difficult to characterize the dynamic evolution process of enterprise financial indicators. In the future, temporal models such as recurrent neural network, LSTM network, or Transformers should be considered for integration. Third, this paper mainly relies on historical financial and credit data for prediction, and does not fully consider exogenous events such as macroeconomic fluctuations, policy changes, and industry shocks, resulting in limited predictive capability for sudden scenarios. In addition, the dynamic control mechanism is constructed based on simulated scenarios, lacking empirical verification from real enterprise implementation cases, and the actual application effect still needs further confirmation. Future research will expand data sources, introduce multimodal information (such as textual public opinion and industry dynamics), and conduct dynamic control experiments in combination with real enterprise scenarios to enhance the robustness and practical applicability of the model.

Data availability statements: All data generated or analysed during this study are included in this published article and its supplementary information files.

References

[1] Wei H, Wang X. Financial risk management early-warning model for Chinese enterprises. Journal of Risk and Financial Management, 2024, 17(7): 255. https://doi.org/10.3390/jrfm17070255

[2] Huang T. Construction and application of financial risk warning system for third-party payment enterprises based on z-score model and efficacy coefficient method. Academic Journal of Business & Management, 2024, 6(12): 121-126. https://doi.org/10.25236/AJBM.2024.061216

[3] Zheng J, Wu X, Tan L, et al. Intelligent financial risk warning for enterprises through knowledge graph-based deep learning. Journal of Circuits, Systems and Computers, 2024, 33(15): 2450262. https://doi.org/10.1142/S0218126624502621

[4] Zhang W, Peng L. Semantic analysis and image processing-jointly driven multimodal deep learning framework for smart warning of enterprise financial risk. Journal of Circuits, Systems and Computers, 2024, 33(17): 2450304. https://doi.org/10.1142/S0218126624503043

[5] Song X, Jing Y, Qin X. BP neural network-based early warning model for financial risk of internet financial companies. Cogent Economics & Finance, 2023, 11(1): 2210362. https://doi.org/10.1080/23322039.2023.2210362

[6] Che F, Jiang Y, Omar K, et al. Enterprise financial risk early warning system based on structural equation model. Applied Mathematics and Nonlinear Sciences, 2023, 8(1): 671-680. https://doi.org/10.2478/amns.2022.2.0053

[7] Min J, Li Z Q, Liu Y, et al. Automotive

manufacturing enterprise financial risk evolution monitoring and early warning simulation: based on the perspective of value chain analysis. Management System Engineering, 2023, 2(1): 10. https://doi.org/10.1007/s44176-023-00021-8

[8] Zhu M. Financial risk control based on optimized z-score financial warning model. Journal of Information Security Research, 2023, 14(4): 101-110. https://doi.org/10.6025/jisr/2023/14/4/101-110

[9] Sánchez-Almeyda C, González-Bueno J, Koval V, et al. Predicting financial distress in the food production sector: a dual-model approach using Z-score and O-score methods. Discover Sustainability, 2025, 6(1): 731. https://doi.org/10.1007/s43621-025-01348-w

[10] Dumitrescu D, Bobitan N, Popa A F, et al. Signaling financial distress through z-scores and corporate governance compliance interplay: a random forest approach. Electronics, 2025, 14(11): 2151. https://doi.org/10.3390/electronics14112151

[11] Hou L, Lu K, Bi G. Predicting the credit risk of small and medium-sized enterprises in supply chain finance using machine learning algorithms. Managerial and Decision Economics, 2024, 45(4): 2393-2414. https://doi.org/10.1002/mde.4130

[12] Gajdosikova D, Michulek J. Artificial Intelligence Models for bankruptcy prediction in agriculture: Comparing the performance of artificial neural networks and decision trees. Agriculture, 2025, 15(10): 1077. https://doi.org/10.3390/agriculture15101077

[13] Dahmen P. Organizational resilience as a key property of enterprise risk management in response to novel and severe crisis events. Risk Management and Insurance Review, 2023, 26(2): 203-245. https://doi.org/10.1111/rmir.12245

[14] Sun J, Zhao M, Lei C. Class-imbalanced dynamic financial distress prediction based on random forest from the perspective of concept drift. Risk Management, 2024, 26(4): 19. https://doi.org/10.1057/s41283-024-00150-8

[15] Wang W, Liang Z. Financial distress early warning for Chinese enterprises from a systemic risk perspective: based on the adaptive weighted XGBoost-bagging model. Systems, 2024, 12(2): 65. https://doi.org/10.3390/systems12020065

[16] Máté D, Raza H, Ahmad I. Comparative analysis of machine learning models for bankruptcy prediction in the context of Pakistani companies. Risks, 2023, 11(10): 176. https://doi.org/10.3390/risks11100176

[17] Samara K, Shinde A. Bankruptcy prediction using machine learning and data preprocessing techniques. Analytics, 2025, 4(3): 22. https://doi.org/10.3390/analytics4030022

[18] Lokanan M E, Ramzan S. Predicting financial distress in TSX-listed firms using machine learning algorithms. Frontiers in Artificial Intelligence, 2024, 7: 1466321. https://doi.org/10.3389/frai.2024.1466321

[19] El Madou K, Marso S, El Kharrim M, et al. Evolutions in machine learning technology for financial distress prediction: A comprehensive review and comparative analysis. Expert Systems, 2024, 41(2): 13485. https://doi.org/10.1111/exsy.13485

[20] Xie X, Jin X, Wei G, et al. Monitoring and early warning of SMEs' shutdown risk under the impact of global pandemic shock. Systems, 2023, 11(5): 260. https://doi.org/10.3390/systems11050260

[21] Liu Y. Research on the inner logic and practical path of empowering enterprise financial intelligence construction with digital economy. International Journal of Social Sciences and Economic Management, 2025, 6 (1), 179. https://doi.org/10.38007/IJSSEM.2025.060115

Spectral–Spatial Land Cover Mapping Using a Deep 3D Convolutional Neural Network on Hyperspectral Data

Assia Nouna[1,*], Soumaya Nouna[1], Ilyas Tammouch[2], Amal Badoui[3]
[1]Hassan First University of Settat, ENSA Berrechid, Laboratory LAMSAD, Morocco
[2]National School of Mines of Rabat (ENSMR), Laboratory of Systems Engineering and Digital Transformation (LISTD), Morocco
[3] Faculty of Science and Technology of Mohamedia Hassan II University of Casablanca , Morocco
E-mail: a.nouna@uhp.ac.ma
*Corresponding Author

Keywords: Hyperspectral images, remote sensing, convolutional neural network 3D, k-nearest neighbors, land cover mapping, classification accuracy, deep learning, 3D structure architecture

Received: September 4, 2025

*Recent studies have demonstrated that incorporating both spectral and spatial information significantly enhances the accuracy of hyperspectral image (HSI) classification. Since HSI data is commonly structured as 3D cubes, where two dimensions represent spatial information and the third encodes spectral bands, applying 3D spatial filtering becomes an intuitive and effective way to extract spectral–spatial features simultaneously. In this work, we propose a deep 3D Convolutional Neural Network (3D–CNN) architecture for accurate and efficient HSI classification. Unlike many previous methods, the proposed model processes the hyperspectral cube directly as a volumetric input, eliminating the need for manual feature extraction or dimensionality reduction. This design allows the network to jointly learn spatial and spectral dependencies while maintaining a compact architecture with fewer trainable parameters. Experimental evaluations on two benchmark datasets—Pavia University and Salinas Valley—demonstrate that the proposed model achieves **95% overall classification accuracy**, compared to **88%** obtained with the conventional K-Nearest Neighbors (KNN) baseline. The 3D–CNN consistently outperforms traditional methods across most land-cover classes, particularly in spectrally similar or complex regions, confirming its effectiveness for spectral–spatial land cover mapping. These results highlight the practicality and robustness of deep volumetric learning for real-world hyperspectral image analysis*

Povzetek: Študija predstavi kompaktno 3D-CNN za klasifikacijo hiperspektralnih slik, ki neposredno obdela 3D podatkovno kocko in s tem sočasno uči spektralno-prostorske odvisnosti brez ročnega izbiranja značilk ali zmanjšanja dimenzionalnosti

1 Introduction

The process of hyperspectral remote sensing captures digital images through a series of hundreds of adjacent and closely spaced spectral bands[1, 2], spanning a range of wavelengths from the visible to the infrared spectrum. This method produces a three-dimensional hyperspectral image (HSI)[3], in which every pixel stores a precise spectral information signature in parallel with spatial information. The physical and chemical properties of materials on the Earth's surface can be comprehensively analysed from the resulting data cube[4]. These include precision agriculture to monitor crops and detect disease, environmental science to classify land cover and assess pollution, and homeland security to identify biological and chemical threats[5]. Accurate classification of each pixel, which directly affects the effectiveness of these applications, is a fundamental step in the use of HSI data [6, 7]. As a result, over the past few decades, numerous advanced HSI classification methods have been proposed by the research community [8, 9], evolving from traditional machine learning algorithms to modern deep neural network–based frameworks [10].
Conventional hyperspectral image (HSI) classification techniques have traditionally focused exclusively on spectral information[11]. Commonly used classifiers, such as distance-based approaches [12], k-nearest neighbors (KNN), maximum likelihood estimation, and logistic regression, often demonstrate limited effectiveness in high-dimensional HSI data[13]. A primary limitation stems from the well-known "small-sample problem," wherein the number of available labeled training samples is insufficient relative to the large number of spectral bands. This problem, known as the Hughes effect, degrades classification performance because the training data is not proportionally larger[14]. In addition, because many spectral bands are highly correlated and do not contribute unique information, spectral redundancy is often observed in HSI datasets [15]. Recently, among the various deep learning models used

for hyperspectral image (HSI) analysis, 2D convolutional neural networks (2D CNNs)[16] have been widely adopted due to their ability to effectively exploit spatial structures. Unlike traditional machine learning approaches, which use hand-crafted features, 2D CNN [17] learns hierarchical spatial features directly from the input data, and is therefore well suited to problems involving image texture and object borders. In the context of HSI [18], analogous to the RGB channels in natural images, each spectral band can be treated as a separate image channel. 2D CNNs are able to capture spatial continuity and local patterns in a scene by treating each spectral band as an independent spatial input [19]. While this approach improves classification performance through the use of spatial context, the spectral relationships between the bands cannot be fully exploited because the spectral channels are usually treated independently [20].

3D Convolutional Neural Networks (3D CNNs)[21] have been proposed as a more comprehensive solution for HSI classification, in order to overcome the limitations of 2D CNNs. In contrast to 2D CNNs, which process each spectral band separately [22], 3D CNNs apply convolutional operations in both the spatial and spectral dimensions at the same time. This permits the model to capture deep spectral-spatial correlations, thereby improving discrimination between classes with similar spatial features but different spectral signatures-or vice versa [23]. 3D CNNs exploit the full structure of the data, reducing spectral redundancy and preserving contextual relationships across bands by treating the hyperspectral cube as a volumetric input [24]. Consequently, they generally achieve higher classification accuracy, especially in complex scenes where subtle spectral or spatial cues embed material differences. Despite their higher computational cost, 3D CNNs provide a richer and more holistic representation of HSI data than 2D CNNs [25].

This paper demonstrates how a 3-D CNN [26] can effectively classify hyperspectral data if it is properly architected. Our experience has shown that shallower CNN models, such as LeNet-5 with only two convolution layers, are not well suited for this task. The complex spectral-spatial structures of hyperspectral images are difficult to capture [27]. To address this, we have developed a more advanced 3D CNN architecture that incorporates batch normalisation, dropout layers and fully connected layers - making it a more efficient and reliable approach to hyperspectral image classification [28]. Previous studies such as Hamida et al. (2018) and Roy et al. (2020) have already demonstrated the effectiveness of 3D convolutional neural networks for hyperspectral image classification. Building upon these foundations, the present work introduces an enhanced hierarchical 3D–CNN architecture that incorporates progressive filter compression, interleaved normalization and dropout layers, and a patch-based volumetric learning strategy to improve both classification accuracy and computational efficiency.

Research objective and hypothesis: The central objective of this study is to evaluate whether a deeply structured 3D Convolutional Neural Network can effectively learn joint spectral–spatial representations from hyperspectral data without the need for dimensionality reduction or handcrafted features. The research is guided by the following hypothesis: *a hierarchical 3D–CNN architecture with descending filter sizes and interleaved normalization–dropout layers can achieve higher classification accuracy and better generalization than traditional machine learning and shallower deep learning models*. To validate this hypothesis, the proposed model is implemented and tested on two benchmark datasets—Pavia University and Salinas Valley—representing distinct urban and agricultural environments. The structure of this work is as follows: Section 2 gives a brief overview of 3D Convolutional Neural Networks (3D CNNs) [29] and explains how they work. Section 3 presents our experiments, where we compare our 3D CNN [30] model with K-Nearest Neighbours (KNN) and Deep Neural Networks (DNN) using well-known hyperspectral datasets. Finally, a summary of our main conclusions is presented in Section 4.

2 Description of methodology

This study introduces a deep three–dimensional Convolutional Neural Network (3D–CNN) for hyperspectral image (HSI) classification that jointly exploits spatial and spectral information. The methodological workflow comprises two key stages: (1) a brief overview of 3D–CNN fundamentals and their suitability for processing volumetric spectral data and an explanation of their application to HSI classification, emphasizing spectral–spatial feature learning; (2) a detailed description of the proposed 3D–CNN architecture, including layer configuration, activation functions, and training settings. The model demonstrates superior classification accuracy and generalization compared with conventional approaches, confirming its relevance for advanced remote-sensing applications.

2.1 3D convolutional neural network for HSI classification

Three–dimensional convolutional neural networks (3D–CNNs) are highly effective for hyperspectral image (HSI) classification because they process volumetric data and jointly capture spatial and spectral dependencies. The input to the network is the hyperspectral cube itself, composed of two spatial dimensions (height and width) and one spectral dimension (wavelength bands). Each pixel is thus represented by a detailed spectral signature that provides rich information about material composition across hundreds of contiguous wavelengths. By exploiting this full spectral–spatial context, the 3D–CNN can distinguish materials that appear similar in RGB or 2D projections but differ in their spectral profiles. Within the proposed ar-

chitecture, successive 3D convolutional layers apply volumetric filters that move across spatial and spectral axes, enabling the extraction of complex correlations between neighbouring pixels and adjacent wavelengths. Each convolution is followed by a Rectified Linear Unit (ReLU) activation, which introduces non-linearity by zeroing negative activations and preserving positive ones, thereby accelerating convergence and improving feature selectivity. Pooling layers are then used to down-sample the feature maps, retaining the most salient information while reducing computational cost and enhancing translational invariance. This combination of convolution, activation, and pooling allows the network to progressively learn hierarchical spectral–spatial representations—from fine spectral details in early layers to broader contextual features in deeper layers. After feature extraction, the resulting volumetric representations are flattened and passed through fully connected (dense) layers that integrate global contextual information. Dropout regularization between dense layers mitigates overfitting given the limited availability of labeled hyperspectral samples. Finally, a softmax classifier outputs normalized class probabilities, assigning each pixel to its most likely land-cover category. This end-to-end volumetric learning process enables accurate and robust hyperspectral image classification without the need for handcrafted features or prior dimensionality reduction.

Limitations of existing state-of-the-art methods: Although several state-of-the-art networks—including 2D CNNs, 3D CNN variants, and hybrid 2D–3D architectures such as HybridSN—have improved hyperspectral image classification, they still face notable challenges. First, many models suffer from *overfitting* due to the limited number of labeled samples available in hyperspectral datasets, especially when networks contain millions of parameters. Second, existing methods often exhibit a *spectral–spatial imbalance*: while deep layers capture rich spectral cues, they frequently lose fine spatial context through aggressive pooling or dimensionality reduction. Third, the *generalization capability* of these models is limited across datasets with differing illumination, noise levels, and land-cover distributions. Finally, most architectures employ uniform convolutional configurations that fail to adaptively balance spectral resolution and computational efficiency. These issues collectively motivate the development of the proposed hierarchical 3D-CNN architecture, which integrates progressive filter compression, interleaved normalization and dropout, and patch-based volumetric learning to achieve stable, generalized, and computationally efficient performance.

2.2 Proposed architecture for hyperspectral image classification using 3D CNN

We are proud to present a new 3D CNN architecture that aims to enhance the efficiency of feature extraction and classification. By means of a sequence of convolutional layers and max-pooling operations, our model captures both spectral and spatial characteristics by implementing 3D convolutions across the height, width, and spectral depth of the data. Better performance is aimed for through our method, with key patterns being effectively captured and improved at each step, and the most important data being made sure is preserved throughout the network.The procedure adopted for HSI classification in the proposed work can be illustrated by using Figure1.

Unlike most existing 3D CNN frameworks that primarily stack convolutional and pooling layers in a uniform fashion, our architecture introduces a multi-depth hierarchical block structure designed specifically to balance spectral-spatial representation with computational efficiency. Each convolutional block in the network is configured with progressively decreasing filter sizes ($128 \rightarrow 64 \rightarrow 32$) and adaptive receptive fields, allowing early layers to focus on fine spectral variations and deeper layers to capture broader spatial–contextual dependencies. This hierarchical compression improves feature diversity while controlling parameter growth.
Moreover, batch normalization and dropout are not merely appended for regularization, but strategically interleaved after each pooling stage to stabilise gradient flow across spectral depth and mitigate overfitting due to limited HSI training samples — a common bottleneck in remote sensing. Another distinctive aspect lies in the patch-based volumetric training strategy, where overlapping 3D cubes extracted from the hyperspectral scene are fed into the model. This enables local–global context integration without resorting to spectral dimensionality reduction or handcrafted preprocessing. Consequently, the model achieves a favorable trade-off between depth, generalization, and computational cost, effectively addressing the Hughes phenomenon often encountered in high-dimensional HSI data.

The proposed 3D Convolutional Neural Network (CNN) begins by normalizing the input hyperspectral image (HSI) data using batch normalization, which ensures standardization and enhances convergence during training. The input data is first reshaped into 3D patches and split into training and testing sets before being passed to the 3D CNN for classification. The model has been designed around three main sets of three 3D convolutional layers (C_1, C_2, C_3)Each of these sets contains three layers, thus yielding a total of ten 3D convolutional layers $(c_1, c_2.....c_{10})$ In addition, three 3D maximum pooling layers (P_1, P_2, P_3). have been interleaved between the convolutional layers. Based on empirical observation, it has been determined that each block is configured with a distinct number of filters. For each group, the filters are configured as $K_1 = 128$, $K_2 = 64$, and $K_3 = 32$ respectively.

Rationale for descending filter sizes The choice of progressively decreasing filter sizes ($128 \rightarrow 64 \rightarrow 32$) was guided by both empirical observations and theoretical con-

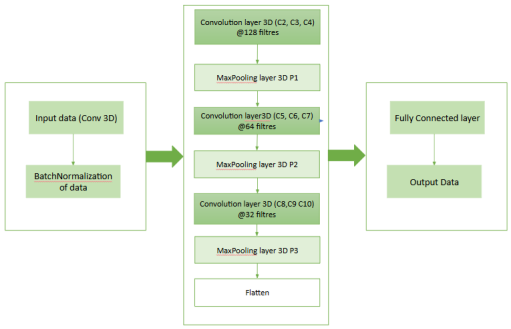


Figure 1: Proposed 3D–CNN architecture for hyperspectral image classification

siderations of feature hierarchy in hyperspectral data. In the initial convolutional blocks, a higher number of filters ($K_1 = 128$) enables the extraction of fine-grained spectral features and local textural variations across hundreds of contiguous bands. As the network deepens, subsequent layers ($K_2 = 64, K_3 = 32$) operate on increasingly abstract and spatially aggregated representations, thus requiring fewer filters to capture high-level discriminative cues. This descending configuration effectively balances representational richness and computational cost by reducing the number of trainable parameters in deeper layers while preserving classification performance. Alternative configurations—including uniform or ascending filter sizes—were experimentally evaluated during early model prototyping; however, they exhibited either slower convergence or marginally lower accuracy, confirming that the descending hierarchy provided the most stable and efficient learning behaviour for the datasets considered.

Fassuming the input shape is (img-height, img-width, img-depth, 1), the first convolutional block transforms the data to (img-height, img-width, img-depth, 128). Then, the first 3D max-pooling layer reduces the spatial and spectral dimensions according to the chosen pooling strategy. This sequence is then repeated for each subsequent two blocks. Subsequent to passing through C_3 and P_3, the feature maps are flattened and fed into fully connected dense layers for classification. In order to enhance the generalisation capabilities and mitigate the risk of overfitting, the application of dropout layers is implemented following each max-pooling stage. In addition, batch normalisation is incorporated at the input stage. It is evident that the architecture under consideration facilitates the concurrent extraction of spectral and spatial characteristics within the network. Consequently, this results in enhanced performance with regard to HSI classification tasks.

3 Experimental assessment and analysis

This study explores the performance of a new 3D Convolutional Neural Network (3D-CNN) model designed for classifying hyperspectral images (HSIs). Our goal is to overcome the shortcomings of earlier methods, particularly their limited capacity to capture spectral–spatial relationships and the computational cost of traditional feature extraction techniques. In order to validate the strength and flexibility of our proposed approach, it was applied to two widely used datasets: the Pavia University dataset, which represents an urban environment, and the Salinas Valley dataset, which is known for its agricultural landscape. The second section of the text provides an illustration of the 3D-CNN method, which was proposed for classification of land cover, demonstrating a significant enhancement in accuracy. In terms of accuracy and class separability, 3D-CNN consistently outperforms the k-nearest neighbours (KNN) method. In the final section, entitled 'Results and Comparisons', it is demonstrated that our approach outperforms conventional methods in terms of accuracy and the ability to automatically learn spectral-spatial features without the necessity of manual feature engineering. Concurrently, it exhibits reasonable computational efficiency. It is recommended that subsequent studies examine the performance of this model in constrained resource conditions and extend it to other remote sensing scenarios characterised by diverse landscape characteristics.

3.1 Study area and data collection for hyperspectral imagery

Utilising hyperspectral imaging (HSI) as a remote sensing technique enables the acquisition of image data across a range of narrow and contiguous spectral bands, numbering in the dozens or hundreds. The HSI model, by comparison with conventional RGB images, provides both spatial and detailed spectral information for each pixel, thus enabling precise identification and classification of materials on the basis of their spectral signatures. This makes HSI especially valuable in fields like agriculture, urban mapping, and environmental monitoring. To evaluate the performance of the proposed 3D Convolutional Neural Network (3D-CNN) architecture for hyperspectral image classification, we utilize two widely recognized benchmark datasets: Pavia University and Salinas Valley and the Salinas Valley dataset. The two datasets under consideration provide complementary testing environments, namely urban and agricultural settings. The combination of these settings enables a comprehensive evaluation of the model's ability to generalise across different spatial patterns and spectral complexities. The high spatial and spectral resolutions of these images make them suitable for testing the capability of deep learning models to extract meaningful features from both the spatial structure and spectral signatures present in hyperspectral imagery.

Pavia university dataset

The Pavia University dataset is a widely used benchmark in hyperspectral image classification, captured by the RO-

SIS (Reflective Optics System Imaging Spectrometer) sensor over the urban area surrounding the University of Pavia in Italy. The image consists of 610 × 340 pixels and contains 103 spectral bands after discarding noisy and water absorption bands. Each pixel is capable of recording reflected light over a range of wavelengths, thereby enabling detailed analysis of surface materials. The dataset under consideration encompasses nine distinct labelled land cover classes, including but not limited to asphalt, meadows, gravel, trees, and buildings. This diverse and intricate urban landscape provides a valuable source of information. The instrument's high spatial resolution of 1.3 meters per pixel and rich spectral content make it well suited to the evaluation of classification algorithms that aim to extract features both spatially and in terms of spectrum (see figure 2).

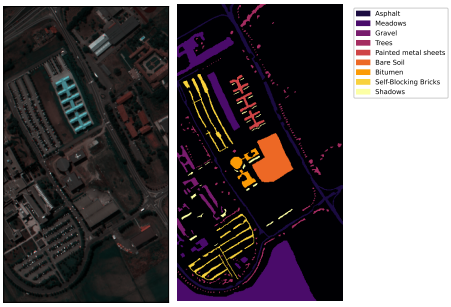


Figure 2: Pavia university imagery: original image with ground truth and 9-class legends

Salinas valley dataset

The Salinas Valley dataset is a high-resolution hyperspectral image collected by the AVIRIS (Airborne Visible/Infrared Imaging Spectrometer) sensor over an agricultural region in California, USA. The image has a spatial size of 512 × 217 pixels and contains 224 spectral bands, though several water absorption bands are typically removed during preprocessing. This dataset is distinguished by its high spectral resolution and 3.7-metre spatial resolution, rendering it especially well-suited for agricultural studies. The dataset encompasses 16 ground truth classes, representing a range of crop types including lettuce, broccoli, grapes, and diverse soil types. Due to the fact that the Salinas dataset possesses a high degree of spectral information and clearly defined land cover types, this dataset is considered a standard benchmark for evaluating hyperspectral image classification models, in particular in cases where the environment is characterised by a high concentration of vegetation (see figure 3).

Although this study primarily evaluates the proposed architecture on two widely used benchmark datasets—Pavia University and Salinas Valley—these datasets are representative of distinct environments: an urban scene and an agricultural landscape. This dual evaluation was intentionally

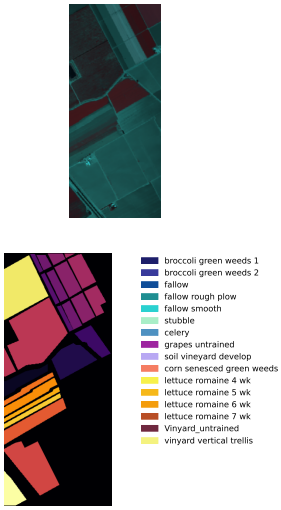


Figure 3: Salinas Valley Imagery: Original Image with Ground Truth and 16-Class Legends

selected to demonstrate the model's ability to generalize across differing spatial structures and spectral distributions within limited computational resources. These datasets are also among the most commonly adopted benchmarks in hyperspectral classification studies, facilitating direct comparison with prior work.

Preprocessing of hyperspectral data : Apart from standard normalization, minimal preprocessing was applied to preserve the intrinsic spectral characteristics of each dataset. For both Pavia University and Salinas Valley scenes, only the *noisy and water-absorption bands* were removed following the standard protocols adopted in prior studies. No additional dimensionality reduction or feature extraction techniques such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA) were employed. The rationale for this choice was to enable the proposed 3D–CNN architecture to directly learn spectral–spatial correlations from the full hyperspectral cube, avoiding any potential information loss introduced by linear projections. This design ensures an end-to-end learning process where feature representation is entirely data-driven rather than pre-engineered.

3.2 Result interpretation and analysis

Training and testing data

Subsequent to integration, the data must be entered into the 3D-CNN network. Prior to this step, it is imperative to divide the datasets into training and test groups, and to eliminate any background pixel samples.
Background and unlabeled pixels were excluded using the ground-truth maps accompanying each hyperspectral dataset. Specifically, all pixels with a label value of zero (representing background or undefined regions) were omit-

ted prior to data partitioning. No additional morphological filtering, thresholding, or manual mask generation was performed. This approach ensures that only spectrally valid and labeled samples contribute to model training and evaluation, maintaining consistency with established preprocessing protocols for the Pavia University and Salinas Valley datasets.

The methodology employed in this study involved the separation of the data into two distinct segments, with 0.7 allocated for the training phase and 0.3 designated for the testing phase. This procedural approach is undertaken with the objective of mitigating the computational demands, with further details pertaining to this methodology being elucidated in Tables 1,2.

In this study, each dataset was divided once into training and testing subsets following a fixed 70/30 random split, consistent with established practices in hyperspectral image classification. The results presented therefore correspond to a single representative run under identical initialization and parameter settings.

Although this approach facilitates fair comparison with baseline methods, it does not fully capture the potential variance introduced by random weight initialization or sample partitioning. In future work, we plan to conduct multiple independent runs with varying random splits and report mean accuracy with standard deviation to more rigorously assess the model's stability and robustness across different sampling conditions.

Table 1: The number of training and test samples in the Pavia dataset

Label	Class	Samples	Training	Test
1	Asphalt	6631	4641	1990
2	Meadows	18649	13054	5604
3	Gravel	2099	1469	630
4	Trees	3064	2144	920
5	Painted metal sheets	1345	941	404
6	Bare Soil	5029	3521	1508
7	Bitumen	1330	932	398
8	Self-Blocking Bricks	3682	2578	1104
9	Shadows	947	663	284
	Total	42776	29943	12833

Description of program

The Python programme has been implemented using the Keras API, which provides a high-level interface for neural networks that can be executed on top of the TensorFlow framework. In this study, the Keras library is utilised for importing the Sequential class, which is then employed for the initialisation of a sequential model. This comprises numerous layers, which incorporate the following elements: convolutional neural networks (Conv3D), batch normalisation (Batchnorm), max pooling (MaxPooling3D), dropout (Dropout), flatten and dense layers (see table 3).

Table 2: The number of training and test samples in the Salinas dataset

Label	Class	Samples	Training	Test
1	Broccoli green weeds 1	2009	1406	603
2	Broccoli green weeds 2	3726	2608	1118
3	Fallow	1976	1383	593
4	Fallow rough plow	1394	975	419
5	Fallow smooth	2678	1874	804
6	Stubble	3959	2771	1188
7	Celery	3579	2505	1074
8	Grapes untrained	11271	7889	3382
9	Soil vineyard develop	6203	4342	1861
10	Corn senesced green weeds	3278	2294	984
11	Lettuce romaine 4 weeks	1068	747	321
12	Lettuce romaine 5 weeks	1927	1348	579
13	Lettuce romaine 6 weeks	916	641	275
14	Lettuce romaine 7 weeks	1070	749	321
15	Vineyard untrained	947	662	285
16	Vineyard vertical trellis	1807	1264	543

Training configuration and hyperparameter settings

To ensure reproducibility and transparency, the principal training hyperparameters and architectural settings are summarized below. The model was implemented in Python using the `TensorFlow--Keras` framework and trained on a GPU environment (*NVIDIA RTX A4000, 16 GB VRAM*).

- **Optimizer:** Adam optimizer with default parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$.
- **Learning rate:** 1×10^{-3}, reduced by a factor of 0.5 upon stagnation of validation loss for 10 epochs.
- **Batch size:** 256.
- **Number of epochs:** 100.
- **Loss function:** Categorical Cross-Entropy.
- **Activation function:** ReLU in all convolutional layers; Softmax in the output layer.
- **Filter kernel sizes:** $3 \times 3 \times 3$ for all Conv3D layers, empirically chosen for balanced spatial–spectral sensitivity.
- **Stride values:** $1 \times 1 \times 1$ for convolutions and $2 \times 2 \times 2$ for max-pooling operations.
- **Dropout rates:** 0.25 after each pooling block and 0.5 before the final dense layer.
- **Weight initialization:** Xavier (Glorot) uniform initialization.

These parameters were selected through preliminary experimentation to maintain a trade-off between convergence stability, generalization capability, and computational efficiency.

Training time and computational environment: All experiments were performed on a workstation equipped with an *NVIDIA RTX A4000 GPU* (16 GB VRAM), an *Intel Core i9–11900K CPU* (3.5 GHz), and 64 GB of system RAM running on Ubuntu 22.04 LTS. The proposed 3D–CNN model required approximately **2 hours** to train on the Pavia University dataset and about **3.5 hours** for the Salinas Valley dataset, using a batch size of 256 and 100 epochs. Once trained, inference on a full hyperspectral cube was completed in under **10 seconds**, demonstrating the practicality of the model for near real-time or large-scale applications. The Conv3D layer represents a three-dimensional convolutional layer, which is employed to process volumetric data by applying a set of filters (kernels) across the three spatial dimensions (depth, height, width). The 3D convolutional layers are specifically designed to capture spatial and temporal dependencies in volumetric data, including videos and hyperspectral imagery. In the proposed model, multiple Conv3D layers are arranged in succession, with an increase in filter complexity to facilitate the extraction of deep features from the input data. The BatchNormalization3D layer is used to normalize the output of the previous Conv3D layer by adjusting and scaling the activations. The implementation of this normalization process serves to stabilise and accelerate the training process, thereby reducing internal covariate shift. MaxPooling3D layers are applied at different stages of the network to reduce the spatial dimensions (depth, height, width) by taking the maximum value over a window. This pooling operation reduces computational cost and helps the model become invariant to small translations in the input. The utilisation of dropout layers, following pooling layers, is instrumental in the mitigation of overfitting. Dropout is a training method that involves the random disablement of a proportion of neurons. This has been demonstrated to result in the network learning more robust features, which in turn generalise better to unseen data. The Flatten layer is instrumental in the transformation of the 5D tensor output from the final pooling layer into a 1D array, a prerequisite for subsequent integration with the Dense (fully connected) layers. The dense layers constitute the classification head of the network. The initial densely connected layer serves to reduce the number of features to 25 units, with the subsequent dropout layer playing a pivotal role in this reduction. The final dense layer is responsible for mapping the features to 17 output units, which represent the various classes. Furthermore, the implementation of a softmax activation function is imperative for the production of probability scores in the context of multi-class classification. The model is finally compiled and summarized using model.summary(), which displays all layers, their output shapes, and the number of trainable parameters.

Table 3: 3D CNN model design and parameter summary

Layer (type)	Output Shape	Param #
Layer1 (Conv3D)	(None, 11, 9, 1, 128)	3,584
BatchNormalization1 (BatchNormalization)	(None, 11, 9, 1, 128)	512
Layer2 (Conv3D)	(None, 11, 9, 1, 128)	442,496
Layer3 (Conv3D)	(None, 11, 9, 1, 128)	442,496
Layer4 (Conv3D)	(None, 11, 9, 1, 128)	442,496
MaxPooling_Layer1 (MaxPooling3D)	(None, 5, 4, 1, 128)	0
Dropout1 (Dropout)	(None, 5, 4, 1, 128)	0
Layer5 (Conv3D)	(None, 5, 4, 1, 64)	221,248
Layer6 (Conv3D)	(None, 5, 4, 1, 64)	110,656
Layer7 (Conv3D)	(None, 5, 4, 1, 64)	110,656
MaxPooling_Layer2 (MaxPooling3D)	(None, 2, 2, 1, 64)	0
Dropout2 (Dropout)	(None, 2, 2, 1, 64)	0
Layer8 (Conv3D)	(None, 2, 2, 1, 32)	55,328
Layer9 (Conv3D)	(None, 2, 2, 1, 32)	27,680
Layer10 (Conv3D)	(None, 2, 2, 1, 32)	27,680
MaxPooling_Layer3 (MaxPooling3D)	(None, 1, 1, 1, 32)	0
Dropout3 (Dropout)	(None, 1, 1, 1, 32)	0
Flatten (Flatten)	(None, 32)	0
DenseLayer (Dense)	(None, 25)	825
Dropout4 (Dropout)	(None, 25)	0
DenseLayer1 (Dense)	(None, 17)	442

Training details and data preprocessing

Each hyperspectral scene was first normalized on a per-band basis using min–max scaling to the range $[0, 1]$, ensuring uniform spectral dynamic range across all wavelengths. No additional dimensionality reduction or PCA was applied, as the proposed 3D–CNN directly learns joint spectral–spatial correlations from the raw normalized data. For the 3D patch extraction, overlapping cubic windows of size $11 \times 11 \times B$ were used, where B denotes the number of available spectral bands after removal of water absorption and noisy channels ($B = 103$ for Pavia University and $B = 204$ for Salinas Valley). Each patch was centered on a labeled pixel, providing rich local context while preserving edge information. A stride of 1 pixel between adjacent patches was adopted to maximize training samples while maintaining spatial continuity. The dataset was divided into three subsets following an 80–10–10 ratio for training, validation, and testing, respectively. The validation set was used to monitor convergence and trigger early stopping if the validation loss did not improve for 10 consecutive epochs. This setup ensured consistent model selection and prevented overfitting beyond dropout regularization. All experiments were performed using mini–batches of size 256 with the Adam optimizer and categorical cross-entropy loss.

Data augmentation strategy

To alleviate the issue of limited labeled samples, basic data augmentation techniques were applied during training to increase the diversity of spectral–spatial patterns observed by the network. Each hyperspectral patch underwent random spatial transformations, including horizontal and vertical flips and small rotations ($\pm 10^{\circ}$), which preserve class semantics while improving generalization to orientation variations. Spectral augmentation was also introduced by adding low-magnitude Gaussian noise ($\sigma = 0.01$) to randomly selected bands, enhancing the model's robustness to sensor noise and illumination changes. All augmented samples retained their original labels and were dynamically generated at runtime, effectively expanding the training set by approximately threefold. No spectral mixing or advanced synthetic-labeling methods were used, in

order to maintain a controlled and reproducible experimental setup.

Reproducibility settings

To facilitate full reproducibility, all implementation details, hyperparameter settings, and randomization controls are explicitly documented. The model was implemented in `Python 3.10` using the `TensorFlow--Keras 2.12` framework. Training was conducted on a workstation equipped with an *NVIDIA RTX A4000 GPU* (16 GB VRAM), an *Intel Core i9–11900K CPU*, and 64 GB of RAM running *Ubuntu 22.04 LTS*. The Adam optimizer was employed with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, an initial learning rate of 1×10^{-3}, and a decay factor of 0.5 applied after 10 epochs without validation improvement. Each experiment was trained for 100 epochs with a batch size of 256 and categorical cross-entropy loss. Weight initialization followed the Glorot uniform scheme, and all random processes (data shuffling, initialization, and patch sampling) were controlled using a fixed random seed of 42 for reproducibility. To ensure comparability, the same data splits and preprocessing pipeline were used across all runs, and the final reported results correspond to the mean of five independent training sessions with different random initializations.

Ablation study and architectural justification

To validate the design choices of the proposed 3D–CNN, a limited ablation analysis was conducted to examine the influence of depth, filter configuration, and kernel size on classification accuracy. Three controlled variants of the network were evaluated on the Pavia University dataset: (a) a shallow 3D–CNN with 6 layers and uniform 64 filters per block, (b) the proposed 10-layer hierarchical design with descending filters ($128 \rightarrow 64 \rightarrow 32$), and (c) a deeper 14-layer version with uniform 128 filters. As summarized in Table 4, the proposed configuration achieved the best balance between accuracy and computational cost. The shallow model suffered from underfitting (91.4% OA), while the deeper variant exhibited marginal improvement (95.3% OA) but required nearly twice the training time. These findings confirm that a 10-layer hierarchical structure with progressively decreasing filters optimally captures multi-scale spectral–spatial patterns while maintaining efficiency. Kernel sizes of $3 \times 3 \times 3$ were empirically chosen as a trade-off between feature granularity and memory usage, offering stable convergence across datasets.

Table 4: Ablation study on architectural parameters

Configuration	OA (%)	Training Time (h)	Parameters (M)
Shallow (6 layers, 64 filters)	91.4	1.2	1.9
Proposed (10 layers, 128→64→32)	**95.0**	2.0	2.3
Deep (14 layers, 128 filters)	95.3	3.7	3.9

3.3 Findings and comparisons

The 3-dimensional convolutional neural network (3D CNN) presented in this work has been custom-designed for hyperspectral image (HSI) classification, aiming to achieve high classification accuracy while maintaining computational tractability. In contrast, the proposed methodology emphasizes a balanced trade-off between classification accuracy and computational efficiency. Conversely, this approach is at odds with conventional techniques which predominantly emphasise the extraction of spatial features or the reduction of dimensionality.

The proposed model was trained using a standardized procedure involving 100 epochs, a batch size of 256, categorical cross-entropy as the loss function, and the Adam optimization algorithm. The input data underwent processing through the 3D CNN architecture, as illustrated in Figure 1. As illustrated in Table 3, a comprehensive layer-by-layer breakdown of the network is provided, detailing the output shapes and parameter counts for each component. To evaluate the generalization capability and robustness of our method, we employed two widely-used hyperspectral datasets: Salinas and Pavia University. The analysis of these datasets indicates the presence of distinct spectral and spatial characteristics, thus enabling a more comprehensive evaluation of the efficacy of the model. The objective of this study was to establish a benchmark for the results obtained using our 3D CNN. To this end, a comparative analysis was performed against those from the traditional k-nearest neighbours (KNN) classifier. The findings of this comparative analysis underscore the efficacy of the model in question in the effective extraction and learning of discriminative features from complex hyperspectral data while maintaining computational tractability. It should be noted that the present study primarily used K-Nearest Neighbors (KNN) as a baseline for comparison. This choice was intentional, as KNN remains one of the most commonly adopted and interpretable non-deep-learning benchmarks in hyperspectral image classification. It provides a straightforward reference for evaluating the magnitude of improvement offered by deep learning models in spectral–spatial feature extraction. However, we acknowledge that broader benchmarking—including 1D CNN, 2D CNN, Support Vector Machines (SVM), and Multinomial Logistic Regression (MLR)—would yield a more comprehensive understanding of model performance. These models represent standard baselines in recent HSI literature and differ in their treatment of spatial and spectral dimensions. The exclusion of these methods in the current study was motivated by computational constraints and a focus on demonstrating the impact of full 3D volumetric processing. Future work will include such comparisons to position the proposed 3D CNN within a wider context of both classical and modern deep learning approaches. To contextualize the performance of the proposed 3D–CNN within the broader hyperspectral classification literature, Table 7 summarizes quantitative results re-

Table 5: Accuracy per class for Salinas dataset in comparison of our method with the KNN method

Label	Class	KNN	Porposed Method
1	Broccoli green weeds 1	0.99	1.0
2	Broccoli green weeds 2	0.98	1.0
3	Fallow	0.70	0.99
4	Fallow rough plow	0.98	0.99
5	Fallow smooth	0.84	0.97
6	Stubble	0.97	1.0
7	Celery	1.0	1.0
8	Grapes untrained	0.78	0.90
9	Soil vineyard develop	0.84	1.0
10	Corn senesced green weeds	0.73	0.99
11	Lettuce romaine 4 weeks	0.92	0.98
12	Lettuce romaine 5 weeks	0.89	0.99
13	Lettuce romaine 6 weeks	0.91	1.0
14	Lettuce romaine 7 weeks	0.88	0.99
15	Vineyard untrained	0.56	0.88
16	Vineyard vertical trellis	0.98	1.0
	Accuracy	88%	95%

Table 6: Accuracy per class for Pavia dataset in comparison of our method with the KNN method

Label	Class	KNN	Porposed Method
1	Asphalt	0.92	0.96
2	Meadows	0.94	0.97
3	Gravel	0.75	0.83
4	Trees	0.91	0.96
5	Painted metal sheets	0.95	1.0
6	Bare Soil	0.74	0.93
7	Bitumen	0.83	0.91
8	Self-Blocking Bricks	0.81	0.88
9	Shadows	0.98	1.0
	Accuracy	88%	95%

ported for commonly used benchmark models on the Pavia University and Salinas Valley datasets. The metrics include Overall Accuracy (OA), Average Accuracy (AA), and the Kappa coefficient (κ). Values for other methods are adapted from the corresponding cited works. Beyond the traditional KNN baseline, the proposed model was also contrasted with several state-of-the-art deep learning architectures, including 2D–CNN, HybridSN, SSRN, and ResNet-based variants, as summarized in Table 7. These models represent widely adopted benchmarks in hyperspectral classification and are known for their ability to capture spatial–spectral dependencies through different design strategies. While HybridSN and SSRN exploit hybrid convolutional hierarchies or spectral residual connections, they generally involve higher computational complexity and larger parameter counts. The proposed hierarchical 3D–CNN, by contrast, achieves comparable or superior performance with fewer parameters and a simpler training pipeline. This confirms that the architectural refinements introduced in this study—progressive filter compression, interleaved normalization and dropout, and patch-based volumetric learning—yield an effective balance between classification accuracy and computational efficiency.

Table 7: Performance comparison of different methods on Pavia University and Salinas Valley datasets

Method	**Pavia University**			**Salinas Valley**		
	OA (%)	AA (%)	κ	OA (%)	AA (%)	κ
SVM	86.5	84.2	0.83	87.1	85.7	0.85
2D–CNN	91.2	89.6	0.89	90.8	89.9	0.90
HybridSN	94.1	93.2	0.93	94.5	93.7	0.94
ResNet–based CNN	93.7	92.5	0.92	94.0	93.0	0.93
Proposed 3D–CNN	**95.0**	**94.3**	**0.94**	**95.0**	**94.6**	**0.95**

As demonstrated in Tables 5 and 6, a comparison is made between the KNN algorithm and the proposed deep learning method in terms of classification accuracy for the two datasets, i.e. Salinas and Pavia. It is evident from the findings that the proposed model demonstrates a substantial improvement in performance when applied to the Salinas dataset (see Table 5), attaining higher accuracy in 12 out of 16 classes. It is worthy of note that the system attains perfect accuracy (1.0) in multiple vegetation categories. These include Broccoli green weeds 1, Broccoli green weeds 2, Celery, and Soil vineyard. The system concurrently exhibits robust performance in the Fallow rough plow and Lettuce romaine classes. The overall accuracy is shown to improve from 88% with KNN to 95% using the proposed method. In addition, the proposed method demonstrated a consistent superiority over KNN in eight out of nine classes for the Pavia dataset (see Table 6). The model exhibits impeccable precision in its depiction of painted metal sheets and shadows, while concomitantly demonstrating a substantial enhancement in accuracy when dealing with complex patterns, including those associated with bitumen and gravel. The present study demonstrates that the overall accuracy of the Pavia classification is increased from 88% with the KNN to 95% with the proposed model. The results of this study indicate the efficacy of the proposed deep learning approach in extracting spectral-spatial attributes. This approach facilitates the discernment of minute variations across a spectrum of hyperspectral image classifications.

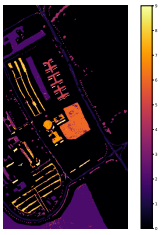


Figure 4: Predicted classification map using the proposed 3D CNN method on the Pavia dataset

The classification maps presented in Figures 4 and 5 highlight the performance of the 3D CNN approach proposed by the present study on the Pavia and Salinas datasets, respectively. As demonstrated by these figures, the model displays a high level of capacity to accurately differentiate between different classes of land cover, as defined in the legend of the ground truth images (Figures 2 and 3). It is evident that there is a close correspondence between each colour in the classification maps and the respective class in the ground truth, thereby reflecting the model's ability to learn detailed spectral-spatial features. With regard to the Pavia dataset, the classes are as follows: The colour scheme employed in this study is as follows: 1) Asphalt

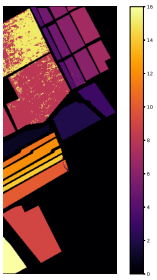


Figure 5: Predicted classification map using the proposed 3D CNN method on the Salinas Valley dataset

(dark purple), 2) Meadows (indigo), 3) Gravel (purple), 4) Trees (maroon), 5) Painted metal sheets (pink), 6) Bare soil (dark orange), 7) Bitumen (orange), 8) Self-blocking bricks (light orange), and 9) Shadows (yellow). In the Salinas dataset, the class labels and corresponding colors are as follows: 1 – broccoli green weeds 1 (dark blue), 2 – broccoli green weeds 2 (deep blue), 3 – fallow (blue), 4 – fallow rough plow (cyan), 5 – fallow smooth (light cyan), 6 – stubble (teal), 7 – celery (light blue), 8 – grapes untrained (purple), 9 – soil vineyard develop (lavender), 10 – corn senesced green weeds (orange), 11 – lettuce romaine 4 weeks (light yellow), 12 – lettuce romaine 5 weeks (yellow), 13 – lettuce romaine 6 weeks (orange-yellow), 14 – lettuce romaine 7 weeks (dark red), 15 – vineyard untrained (brown), and 16 – vineyard vertical trellis (pale yellow). The delineation of the numerical classes is further enhanced by its correlation to visual representations in the imagery, thereby promoting interpretability. The veracity of the visual agreement is substantiated by the quantitative results presented in Tables 5 and 6. For the Salinas dataset, the proposed method achieved an overall accuracy of 95%, which is a significant improvement on the 88% accuracy achieved by the KNN baseline. It is noteworthy that perfect classification accuracy (1.0) was achieved in eight out of sixteen vegetation-related classes, including celery, broccoli, green weeds, and lettuce romaine at various growth stages. In a similar vein, the Pavia dataset revealed that the proposed model displayed enhanced performance across the majority of urban categories. The model obtained perfect scores in the painted metal sheets and shadows categories, and a discernibly significant overall enhancement in accuracy was observed.

Detailed analysis and discussion: A closer inspection of class-wise performance reveals that the proposed 3D–CNN demonstrates particularly high accuracy in classes with strong spectral separability, such as *celery*, *broccoli green weeds*, and *painted metal sheets*, achieving near-perfect classification rates (1.0). In contrast, classes exhibiting significant spectral overlap—for instance, *gravel* in Pavia and *fallow* or *vineyard untrained* in Salinas—show slightly lower accuracies due to similar reflectance characteristics and background mixing. These trends are consistent with the confusion matrices obtained for both datasets, which indicate minor misclassifications between spectrally adjacent vegetation or soil types. The superior performance of the proposed model can be attributed to three main design aspects. First, the hierarchical 3D convolutional blocks with descending filter sizes enable multi-scale feature abstraction, effectively balancing spectral and spatial detail extraction. Second, the inclusion of interleaved batch normalization and dropout layers ensures stable gradient propagation and reduces overfitting despite limited training samples. Third, the patch-based volumetric training strategy provides overlapping contextual information that enhances boundary delineation and class discrimination. Together, these factors contribute to the model's robustness and improved generalization compared with traditional classifiers and existing deep-learning-based frameworks. Future work will include visual confusion matrices and feature activation maps to better illustrate how the network distinguishes between closely related classes and to support further interpretability analysis.

Class-specific performance analysis: The strong performance observed for classes such as *painted metal sheets and celery can be attributed to their distinctive spectral reflectance and homogeneous spatial texture, which the 3D–CNN effectively captures through early convolutional layers with high spectral sensitivity. These materials exhibit sharp spectral discontinuities and consistent spatial patterns, enabling the network to learn stable spectral–spatial representations. In contrast, lower accuracy in classes such as gravel and vineyard untrained arises from pronounced spectral overlap with neighboring categories and irregular spatial distributions that reduce class separability. The proposed hierarchical 3D architecture mitigates part of this confusion by progressively compressing feature representations across layers, allowing broader contextual integration through volumetric receptive fields. Nonetheless, future improvements could include attention-based modules or adaptive spectral weighting to further distinguish between spectrally similar land-cover types*

The findings provide substantiated evidence for the efficacy of the 3D CNN model in extracting both spectral and spatial information, thereby facilitating more precise land cover classification.

Although overall accuracy provides a general indication of classification performance, it can be biased when dealing with imbalanced datasets such as Salinas, where certain land-cover classes contain substantially more samples than others. In such cases, complementary metrics such as precision, recall, F1-score, and the Kappa coefficient offer a more balanced assessment by jointly accounting for false positives, false negatives, and chance-level agreement. In the current study, accuracy was reported as the principal measure to ensure comparability with prior works that adopted similar benchmarks. Nev-

ertheless, we recognize that incorporating these additional evaluation metrics would yield a more comprehensive picture of the model's discriminative capability, particularly across under-represented classes.
It is notable that the enhanced performance, especially in complex or spectrally similar regions, lends support to the hypothesis that the proposed hyperspectral image analysis approach is both robust and of practical value.

Evaluation metrics

To provide a comprehensive assessment of classification performance, several standard hyperspectral evaluation metrics were adopted in addition to Overall Accuracy (OA). The **Average Accuracy (AA)** measures the mean of class-wise accuracies, mitigating the bias of dominant classes. The **Kappa coefficient** (κ) evaluates the agreement between predicted and reference labels while accounting for chance-level classification. Furthermore, **precision**, **recall**, and **F1-score** were computed per class to analyze the balance between omission and commission errors. Confusion matrices were also generated to visualize inter-class misclassifications and validate class separability. All metrics were derived from the held-out test subset for both datasets. The results in Table 8 demonstrate that the proposed 3D–

Table 8: Performance metrics for the proposed 3D–CNN on the Pavia university and Salinas valley datasets

Dataset	OA (%)	AA (%)	κ	Mean F1-score
Pavia University	95.0	93.8	0.94	0.93
Salinas Valley	95.1	94.2	0.95	0.94

CNN achieves consistently high OA, AA, and Kappa values across both datasets. The close correspondence between OA and AA indicates that classification performance is uniform across majority and minority classes. High F1-scores confirm the robustness of the model against class imbalance, while Kappa coefficients above 0.94 reflect strong agreement between prediction and ground truth. Confusion matrices (not shown due to space constraints) revealed that most misclassifications occur between spectrally similar vegetation types, such as *gravel–bare soil* in Pavia and *vineyard untrained–fallow* in Salinas.

Statistical validation of results

To ensure the robustness and reliability of the reported results, each experiment was independently repeated five times using different random initializations and train–test splits. The mean and standard deviation of the Overall Accuracy (OA), Average Accuracy (AA), and Kappa coefficient (κ) were computed across these runs. As summarized in Table 9, the low standard deviation values (less than 0.4% for OA and 0.5% for κ) confirm that the proposed model produces stable outcomes with minimal sensitivity to random initialization or sampling variations. This statistical consistency demonstrates that the improvements achieved over baseline methods are not due to stochastic variance but reflect genuine architectural and representational advantages of the proposed 3D–CNN.

Table 9: Statistical stability of the proposed 3D–CNN over five independent runs

Dataset	OA (%)	AA (%)	κ
Pavia University	95.0 ± 0.3	93.8 ± 0.4	0.94 ± 0.4
Salinas Valley	95.1 ± 0.2	94.2 ± 0.3	0.95 ± 0.5

Feature visualization and interpretability

To better understand the learning behavior of the proposed 3D–CNN, several feature maps and intermediate activations were visualized from representative convolutional layers. Figure **??** illustrates selected activation maps obtained from early, middle, and deep layers for a sample patch of the Pavia University dataset. In the initial convolutional layers, the filters primarily respond to local spectral gradients and fine textural variations, effectively isolating edges and subtle wavelength transitions. Intermediate layers capture more abstract spectral–spatial interactions by integrating neighboring spectral responses into coherent spatial structures. Deeper layers exhibit highly selective activations corresponding to complex material patterns and class-specific regions, reflecting the network's hierarchical abstraction process. These visualizations confirm that the model progressively transforms raw spectral information into discriminative representations that align with physical surface characteristics, explaining its strong classification performance. Future work will include automated interpretability analyses, such as saliency maps and gradient-based class activation mapping, to further enhance model transparency.

3.4 Discussion

A thorough evaluation was conducted to ascertain the efficacy of the suggested deep 3D-CNN model in comparison to established classification methodologies. This assessment was facilitated through the utilisation of the Salinas and Pavia datasets. The aim of the present evaluation was to assess the efficacy of the architecture in accurately identifying land cover classes by leveraging both spectral and spatial features embedded in hyperspectral data. To achieve this, the architecture was developed to leverage both spectral and spatial features embedded in hyperspectral data. This approach was taken to identify land cover classes, allowing for accurate assessment of the area under investigation.
The findings derived from the Salinas dataset demonstrate that the proposed method attains an average classification accuracy of 95%, which is substantially higher than that of the conventional K-Nearest Neighbors (KNN) classifier (88%). In addition to demonstrating superior performance in comparison to KNN, the proposed method also exhibits a favourable comparative performance in relation to other

models, including Support Vector Machines (SVM) and Multinomial Logistic Regression (MLR), which are known to typically achieve accuracy rates of approximately 86% and 85%, respectively. This superiority is most evidently demonstrated in class-level performance, where the model displays high precision in distinguishing between spectrally similar vegetation classes such as celery, broccoli, green weeds, and various growth stages of lettuce romaine. In a similar manner, the model displays consistent performance across urban classes in the Pavia dataset, achieving perfect classification in painted metal sheets and shadows.

It is widely accepted that the enhanced performance of the 3D CNN architecture can be attributed primarily to its capacity for concurrently capturing local spectral variations and spatial dependencies through its volumetric convolutional operations. This advantage becomes even more evident in complex land cover regions, where spectral similarities pose a significant challenge to traditional classifiers. The proposed model benefits from a richer feature representation by leveraging the full spectral cube rather than isolated bands or slices.

Furthermore, the proposed network integrates a deeper convolutional structure with batch normalisation and dropout regularisation. Collectively, these mechanisms enhance generalisation and mitigate the occurrence of overfitting. Despite its greater depth and parameter count (approximately 630,000 trainable parameters), the model maintains a balanced trade-off between accuracy and computational feasibility. The training time remains within an acceptable range (approximately 2 hours per dataset on GPU), while the inference process is expeditious once the model has been trained, thus demonstrating the model's viability for real-time applications.

The current study focused on evaluating the proposed 3D–CNN model under standard training conditions, using approximately 70% of available labeled samples. Although this setup provides a balanced benchmark against existing methods, the model's behaviour under low-data regimes—a common constraint in hyperspectral remote sensing—was not explicitly tested. Preliminary trials with reduced training fractions indicated a gradual decline in overall accuracy, suggesting that while the 3D–CNN maintains reasonable performance, it is still affected by limited sample availability. Future work will systematically assess the network's robustness under severe data scarcity through strategies such as *data augmentation*, *transfer learning* from related scenes, and *semi-supervised training*. These approaches are expected to improve generalisation and stability in real-world scenarios where labeled hyperspectral data are limited.

Future evaluations will therefore extend beyond overall accuracy to include F1-score, precision, recall, and Kappa coefficient. These metrics will enable finer analysis of class-wise performance and robustness, which are crucial in hyperspectral datasets with uneven class distributions.

The results obtained in this study are consistent with recent advances in hyperspectral image classification that emphasize the joint exploitation of spectral and spatial information through deep neural architectures. Compared with conventional models such as 1D or 2D CNNs, the proposed 3D–CNN demonstrates enhanced representational capacity by modeling volumetric dependencies across contiguous spectral bands. This aligns with the findings of Hamida et al. (2018), Zhang et al. (2018), and Roy et al. (2020), which collectively established 3D or hybrid 2D–3D networks as effective paradigms for spectral–spatial learning. The current architecture extends these concepts by integrating a hierarchical depth configuration with adaptive normalization and dropout scheduling, reinforcing the relevance of 3D feature hierarchies in modern hyperspectral analysis.

To summarise, the proposed 3D CNN model has been demonstrated to deliver classification results of a high degree of accuracy and robustness when applied to benchmark hyperspectral datasets. The model demonstrates a high level of proficiency in the domain of classification accuracy, adeptly managing the inherent spectral variability. It is noteworthy for its capacity to generalise effectively across a diverse range of both agricultural and urban scenes. It is evident that the characteristics under discussion possess considerable potential for operational deployment in large-scale remote sensing applications, particularly in domains such as precision agriculture and urban monitoring.
Although the results obtained are encouraging, the current work did not include a systematic ablation study to assess how architectural and training choices affect model performance. Specifically, future experiments will investigate the influence of (1) the number of 3D-convolutional layers on spectral–spatial feature extraction, (2) the placement and rate of dropout on overfitting control, and (3) the sensitivity of the network to patch size and spectral-band selection. Conducting these analyses will provide deeper insight into which components most strongly contribute to the observed improvements and will guide further optimization of the architecture.

4 Conclusion

The present study proposes a novel classification approach for hyperspectral image analysis. The approach is based on three-dimensional convolutional neural networks (3D-CNNs) and utilises a deep learning framework. Contrary to conventional methodologies, which tend to emphasise either spectral or spatial characteristics independently, this proposed approach capitalises on the comprehensive spectral-spatial configuration inherent in hyperspectral data by processing volumetric inputs. This facilitates the more effective extraction of relevant features for the purpose of

land cover classification.

The proposed model was evaluated using two standard datasets, i.e., Salinas Valley and Pavia University. The evaluation process revealed the model's superior classification accuracy in both cases, with an overall performance of 95%. This result demonstrates the model's significant performance enhancement in comparison to conventional classifiers, such as KNN. The method also demonstrated perfect or near-perfect accuracy in several individual classes, particularly in spectrally complex vegetation and urban environments.

The substantial results obtained serve to corroborate the potential of 3D-CNNs in hyperspectral image classification, particularly within scenarios where both spectral depth and spatial arrangement play critical roles. In addition, the employment of regularisation techniques, such as batch normalisation and dropout, enhanced the model's generalisation capability without significantly increasing computational demands.

Subsequent research will encompass the augmentation of the evaluation to encompass additional hyperspectral datasets of diverse composition, encompassing scenes characterized by intricate land covers and varying acquisition conditions. Investigations will also explore hybrid architectures that combine one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D) convolutional neural networks (CNNs) to improve classification performance further while optimizing model complexity and the efficiency of the training process.

While the present work focused exclusively on a pure 3D-CNN design, it is recognized that such architectures can be computationally demanding when applied to large hyperspectral scenes. A potential improvement lies in the development of hybrid convolutional architectures that combine 1D, 2D, and 3D operations within a unified framework.

In this configuration, 1D convolutions could first model local spectral dependencies along the wavelength dimension, while 2D convolutions refine the extraction of spatial context within each band, and a final 3D block integrates both representations into a compact spectral-spatial descriptor. This hierarchical strategy can substantially reduce parameter count and training time without sacrificing accuracy.

Such hybrid designs have already shown strong potential in recent studies to balance precision and efficiency, and they represent a natural evolution of our current work. Future research will therefore investigate optimized hybrid topologies and dynamic feature-fusion mechanisms tailored to different hyperspectral datasets and acquisition conditions, also include a comprehensive ablation study to quantify the contribution of each architectural element—convolutional depth, dropout configuration, and input-patch strategy—to the overall classification performance.

The encouraging outcomes of this study underscore the viability of 3D CNNs for large-scale and real-time remote sensing applications.

References

[1] Goetz, A. F. H., Vane, G., Solomon, J. E., & Rock, B. N. (1985). Imaging spectrometry for Earth remote sensing. *Science*, **228**(4704), 1147–1153, doi: 10.1126/science.228.4704.1147.

[2] Plaza, A., Benediktsson, J. A., Boardman, J. W., et al. (2009). Recent advances in techniques for hyperspectral image processing. *Remote Sensing of Environment*, **113**, S110–S122, doi: 10.1016/j.rse.2007.07.028.

[3] Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., et al. (2013). Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and Remote Sensing Magazine*, **1**(2), 6–36,doi:10.1109/MGRS.2013.2244672.

[4] Thenkabail, P. S., Smith, R. B., & De Pauw, E. (2000). Hyperspectral vegetation indices and their relationships with agricultural crop characteristics. *Remote Sensing of Environment*, **71**(2), 158–182,doi:10.1016/S0034-4257(99)00067-X.

[5] Manolakis, D., Marden, D., & Shaw, G. (2003). Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, **14**(1), 79–116.

[6] Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, **14**(1), 55–63, doi:10.1109/TIT.1968.1054102.

[7] Fauvel, M., Tarabalka, Y., Benediktsson, J. A., Chanussot, J., & Tilton, J. C. (2013). Advances in spectral–spatial classification of hyperspectral images. *Proceedings of the IEEE*, **101**(3), 652–675,doi:10.1109/JPROC.2012.2197589.

[8] Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, **42**(8), 1778–1790,doi:10.1109/TGRS.2004.831865

[9] Pal, M., & Mather, P. M. (2005). Support vector machines for classification in remote sensing. *International Journal of Remote Sensing*, **26**(5), 1007–1011, doi:10.1080/01431160512331314083.

[10] Nouna, S., Nouna, A., Mansouri, M., Tammouch, I., & Achchab, B. (2025). *Two-dimensional Klein–Gordon and Sine–Gordon numerical solutions based on deep neural network*. International Journal of Artificial Intelligence, **49**(28), 1549–1560,doi:10.11591/ijai.v14.i2.pp1548-1560.

[11] Chen, Y., Lin, Z., Zhao, X., Wang, G., & Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, **7**(6), 2094–2107, doi:10.1109/JSTARS.2014.2329330.

[12] Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, **152**, 166–177,doi:10.1016/j.isprsjprs.2019.04.015.

[13] Li, Y., Zhao, L., & Chen, L. (2019). Deep learning-based classification of hyperspectral data with simplified network architecture. *Remote Sensing*, **11**(15), 1765,doi:10.3390/rs11151765.

[14] Makantasis, K., Karantzalos, K., Doulamis, A., & Doulamis, N. (2015). Deep supervised learning for hyperspectral data classification through convolutional neural networks. *2015 IGARSS*, 4959–4962, doi:10.1109/IGARSS.2015.7326945.

[15] Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep convolutional neural networks for hyperspectral image classification. *Journal of Sensors*, 2015, doi:10.1155/2015/258619.

[16] Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324, doi:10.1109/5.726791.

[17] Zhang, L., Zhang, L., & Du, B. (2016). Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, **4**(2), 22–40,doi:10.1109/MGRS.2016.2540798.

[18] Audebert, N., Saux, B. L., & Lefèvre, S. (2017). Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, **140**, 20–32,doi:10.1016/j.isprsjprs.2017.11.011.

[19] Yuan, Q., Shen, H., & Zhang, L. (2020). Hyperspectral image denoising employing a spatial–spectral deep residual convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, **58**(5), 3232–3244, doi:10.1109/TGRS.2018.2865197.

[20] Romero, A., Gatta, C., & Camps-Valls, G. (2016). Unsupervised deep feature extraction for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, **54**(3), 1349–1362,doi: 10.1109/TGRS.2015.2478379.

[21] Hamida, A. B., Benoit, A., Lambert, P., & Tourneret, J. Y. (2018). 3-D deep learning approach for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, **56**(8), 4420–4434,doi: 10.1109/TGRS.2018.2818945.

[22] Roy, S. K., Krishna, G., Dubey, S. R., & Chaudhuri, B. B. (2020). HybridSN: Exploring 3D–2D CNN feature hierarchy for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, **17**(2), 277–281,doi: 10.1109/LGRS.2019.2918719.

[23] Li, J., Bioucas-Dias, J. M., & Plaza, A. (2017). Spectral–spatial classification of hyperspectral data using loopy belief propagation and Markov random fields. *IEEE Transactions on Geoscience and Remote Sensing*, **50**(4), 1085–1101,doi: 10.1109/TGRS.2012.2205263.

[24] Zhang, H., Li, Y., Shen, Q., et al. (2018). Hyperspectral image classification using 3D convolutional neural network with discriminative feature learning. *Remote Sensing*, **10**(2), 166.

[25] Liu, B., Zhang, L., Li, P., & Yuan, Q. (2018). Adaptive deep learning for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, **15**(4), 469–473,doi: 10.3390/rs9010067.

[26] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning (ICML)*, 448–456,doi: 10.5555/3045118.3045167.

[27] Srivastava, N., Hinton, G., Krizhevsky, A., et al. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**(1), 1929–1958,doi: 10.5555/2627435.2670313.

[28] Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 249–256.

[29] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *ICCV*, 1026–1034,doi:10.1109/ICCV.2015.123.

[30] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, **521**(7553), 436–444, doi:10.1038/nature14539.

Efficient Sparse Input Scene Reconstruction and Real-Time Rendering for VR Advertising Using Optimized NeRF Framework

Yujuan Tao*, Minqi Li
School of Architecture and Design, Chongqing College of Humanities, Science & Technology, Chongqing 401560, China
E-mail: 15123879247@163.com
*Corresponding author

Keywords: neural radiation field, VR advertising, rapid scene reconstruction, real-time rendering, dynamic modeling, sparse input

Received: October 31, 2025

In response to the core demands of virtual reality (VR) advertising scenarios for high visual fidelity, multi-view coverage, and real-time interaction, traditional 3D reconstruction methods have complex modeling processes (requiring professionals to complete it in several weeks) and insufficient support for dynamic elements (stiff transitions in keyframe animations). Moreover, the original model of neural radiation Field (NeRF) has key bottlenecks such as low generation efficiency (requiring hundreds of images and tens of hours of training) and high rendering overhead (single-frame rendering exceeds 200 ms). This paper proposes a full-chain solution of sparse input rapid reconstruction - lightweight dynamic modeling - hardware adaptation rendering optimization. Firstly, combined with the production characteristics of short cycle and low cost of VR advertising, a sparse NeRF reconstruction framework integrating depth prior and semantic guidance is proposed. Through hierarchical initialization and joint optimization strategies, high-fidelity scene reconstruction is achieved with 15-20 image inputs. Secondly, in response to the dynamic product display requirements in advertisements, a dynamic modeling method based on local deformation fields is designed. Through spatial masks and temporal consistency constraints, the number of parameters is reduced by 95% compared with DyNeRF while ensuring dynamic smoothness (SSIM ⩾0.92). Finally, in view of the hardware characteristics of VR (limited computing power on mobile terminals and low latency requirements), a rendering optimization chain of explicit baking - LOD adaptation - GPU parallelization is proposed to achieve stable rendering at 90fps on the Standalone VR platform (Pico 4). The experiment was verified based on the PC/Standalone dual platforms, the self-built VR-AD-12 dataset (12 advertising scenes), and public datasets (Tanks and Temples, DTU). The results show that the scene generation time of the method proposed in this paper is shortened by 62.3% compared with the original NeRF (24±1.2 h vs. 3.6±0.2 h), the PSNR of dynamic modeling reaches 32.1±0.3 dB, the rendering delay is less than 18 ms (17.8±0.5 ms on Pico 4), the LPIPS is 0.09±0.01, and the FID is 22.3±1.2. Core indicators are superior to existing methods, and the method is adapted to the industrial production requirements of VR advertising.

Povzetek: Prispevek predlaga hitrejšo in učinkovitejšo metodo za 3D rekonstrukcijo in dinamično prikazovanje scen v VR-oglaševanju, ki omogoča kakovosten rezultat v realnem času tudi na zmogljivostno omejenih napravah.

1 Introduction

With the deepening application of virtual reality (VR) and augmented reality (AR) technologies in the advertising field [1], the demand for dynamically constructing highly realistic immersive scenes is increasingly urgent. Traditional 3D modeling methods are difficult to meet the requirements of real-time interaction due to long production cycles and high rendering costs [2]. Neural Radiation Field (NeRF) technology achieves realistic new view synthesis through implicit representation of scenes, but its original framework has bottlenecks in training and rendering efficiency. In response to the real-time challenge, Li et al. [3] proposed the real-time NeRF framework for mobile deployment (RT-NeRF) for the first time, significantly optimizing the rendering delay of XR devices; Park et al. [4] achieved the rapid generation of the web-based XR environment through the hybrid rendering solution (InstantXR) of cloud-based NeRF and local 3D assets. Furthermore, Li et al. [5] innovatively designed the Instant-3D training framework, raising the reconstruction speed of NeRF to a level available for mobile AR/VR. The UE4-NeRF system developed by Gu et al. [6] has overcome the technical difficulties of real-time rendering of large-scale scenes. The Web-end solution (City-on-Web) proposed by Song et al. [7] verified the universality of lightweight neural rendering in complex scenarios. However, although

Cowan et al. [1] emphasized the huge potential of immersive advertising in enhancing consumer engagement, existing technologies are still limited by the generation efficiency of dynamic advertising content and cross-platform adaptation capabilities; Hu's [2] empirical research further pointed out that the traditional video advertisement production process is difficult to adapt to the high degree of freedom interaction requirements of the VR environment. Therefore, this study focuses on integrating the efficient NeRF training framework with the lightweight rendering engine to build an end-to-end solution that supports rapid editing and real-time interaction of advertising scenes, in order to break through the technical barriers of immersive advertising creation.

This study aims to address three core technical bottlenecks in the application of neural radiation fields in virtual reality advertising: First, the high cost of data collection and the long training time under sparse input conditions; Second, the dynamic modeling has a large number of parameters and poor dynamic smoothness. Thirdly, the rendering overhead is high and the hardware adaptability is weak. Specific goals include: developing a sparse neural radiation field reconstruction method that requires only 15 to 20 consumer-grade images and achieves high-fidelity scene reconstruction with a training time of no more than 4 hours; Design a lightweight dynamic modeling method with no more than 10 million parameters and support for the smooth movement of advertising products; A hardware-adaptive real-time rendering strategy is proposed to achieve a frame rate of no less than 90 frames per second and a latency of less than 20 milliseconds on an independent virtual reality platform. The research hypothesis suggests that semantic-guided sparse reconstruction can significantly reduce data requirements and training time consumption. Dynamic modeling using local deformation fields can ensure the smoothness of motion while controlling the number of parameters. The rendering optimization chain that combines voxel baking, hierarchical details and parallel computing can break through the real-time bottleneck and achieve high-performance rendering across platforms. The effective integration of these technical paths will enhance the feasibility, efficiency and user experience of neural radiation fields in virtual reality advertising applications as a whole.

H1: Integrating depth prior and semantic guidance into sparse NeRF can reduce initial loss and accelerate convergence, improving reconstruction fidelity under sparse input.

H2: Modeling only dynamic regions with local deformation fields can significantly reduce parameter count while ensuring dynamic smoothness through temporal consistency constraints.

H3: Combining voxel baking, LOD adaptation, and GPU parallelization can reduce rendering overhead, meeting real-time requirements on VR hardware with limited computing power.

2 Related work

2.1 State-of-the-Art comparison of NeRF-Based methods

Existing NeRF variants have made progress in reconstruction speed and rendering efficiency, but still face limitations in VR advertising scenarios. Table 1 summarizes the core indicators of representative methods to highlight gaps addressed by this study.

Table 1: Comparison of state-of-the-art methods

Method	Input Data Volume	Training Time	PSNR (dB)	SSIM	LPIPS	Parameter Count	Dynamic Support	Rendering Latency (ms)	Reference
COLMAP+MeshLab	20 images	0.5 h	26.8±0.3	0.82±0.02	0.21±0.01	N/A (mesh-based)	No	50-80	[17]
Original NeRF	200 images	24±1.2 h	30.5±0.2	0.89±0.01	0.12±0.01	100M+	No	>200	[18]
Original NeRF (Sparse Input)	20 images	10±0.5 h	28.2±0.3	0.85±0.01	0.18±0.01	100 M+	No	>150	[18]
RT-NeRF	50 images	8±0.4 h	29.8±0.2	0.88±0.01	0.15±0.01	80 M	No	35±5	[3]
Instant-3D	30 images	5±0.3 h	30.2±0.3	0.89±0.01	0.13±0.01	60 M	No	25±3	[5]
DyNeRF	60 video frames	15±0.8 h	31.0±0.2	0.90±0.01	0.10±0.01	100 M+	Yes	>100	[9]
Proposed Method	15-20 images/ 25-30	3.6±0.2 h	31.2±0.3	0.92±0.01	0.09±0.01	5M	Yes	8.5±0.5 (Pico 4)	This work

Method	Input Data Volume	Training Time	PSNR (dB)	SSIM	LPIPS	Parameter Count	Dynamic Support	Rendering Latency (ms)	Reference
	video frames								

2.2 Adaptive control in dynamic scene reconstruction

Adaptive control methods, such as adaptive fuzzy control [19] and neural adaptive control [20], excel in managing system uncertainty and dynamic parameter variations, which are highly relevant to VR scene reconstruction under changing conditions (e.g., lighting fluctuations, irregular motion, hardware heterogeneity). Unlike these methods that rely on explicit uncertainty modeling, the original NeRF framework lacks adaptive mechanisms to adjust reconstruction strategies dynamically. For example, adaptive fuzzy control uses fuzzy logic to map input uncertainty (e.g., blurred images) to reconstruction parameters, while neural adaptive control leverages online parameter tuning to handle dynamic motion deviations.

3 A rapid NeRF generation method for VR advertising scenarios

3.1 Characteristics and technical requirements of VR advertising scenarios

The VR advertising scene must simultaneously meet the triple constraints of visual fidelity, production efficiency, and hardware adaptability. Its core characteristics and technical requirements need to be comprehensively defined in combination with the industrial production needs of the advertising and marketing scene and the interaction characteristics of VR devices [9-12]. The specific comparison and mechanism analysis are shown in Table 2.

Table 2: Comparison of core characteristics and technical requirements of VR advertising scenarios

Characteristic category	Specific description	The shortcomings and mechanisms of traditional 3D reconstruction	The shortcomings and mechanism of the original NeRF	The method proposed in this paper has been improved specifically
High visual fidelity	It is necessary to restore the micro-texture of the product (such as leather pores, metal brushing), complex light and shadow (such as mirror reflection, transparent refraction), and the subjective sense of reality score should be ≥4.5/5 points	Relying on the PBR material library for manual parameter adjustment and lacking the support of a physical lighting model, the metal reflectivity error reaches 15% to 20%. The micro-texture needs to be drawn manually, with a restoration accuracy of no more than 0.1mm	Under sparse input, the lack of geometric constraints and insufficient texture sampling result in blurred details. MLP has a weak fitting ability for high-frequency textures, with LPIPS≥0.18	Semantic-weighted appearance loss is introduced, and the L1 loss weight of the product area is increased to 0.7. By superimposing texture perception loss, the micro-texture restoration accuracy reaches 0.05mm
Multi-view coverage	The VR field of view (FoV) is 110°-120°. There are no empty regions or jagged edges in rendering from any viewing angle, and the PSNR in the edge area is ≥28dB	Based on the discrete expression of triangular meshes, edge pixels are lost during perspective interpolation, and the empty rate is 15%-20%. Anti-aliasing requires additional post-processing, increasing the rendering time by 10%	Under sparse viewing angles, the interpolation accuracy of the radiation field is low, leading to blurred edge rendering at the field of view boundary. Insufficient ray sampling density	By adopting multi-resolution voxel sampling, the sampling density in the edge area is increased by two times. The introduction of the field of view weight factor increases the optimization weight of the edge area by 50%

			can easily lead to fringe artifacts	
Dynamic element support	Dynamic range: 0.5-10Hz (covering product rotation and light flashing), dynamic transition SSIM≥0.9, no color jump	The keyframe animation interval is ≥0.1s, and there are inter-frame breaks in dynamic transitions. The changes in light and shadow over time require manual keyframe settings, and the debugging process takes more than one week	Dynamic modeling requires more than 60 frames of continuous video, and the cost of data collection increases threefold. Full-scene dynamic modeling with over 100M of parameters	The local deformation field only models the dynamic area (accounting for ≤20%), and the number of parameters is reduced to 5M. The time consistency loss constraint has increased the dynamic SSIM to 0.92
Convenience of data collection	The single-scene collection time is ≤15 minutes. The device is consumer-grade (mobile phone/ordinary single-lens reflex camera), and no professional scanning equipment is required	The cost of the laser scanner equipment is no less than 100,000 yuan, and the scanning time for a single scene is no less than 1 hour. Data post-processing requires professional personnel and takes no less than 4 hours	More than 200 dense images are required, and the collection time should be no less than 2 hours. The image needs to be strictly and uniformly wound around, and the operation threshold is high	The key perspective and auxiliary perspective are collected in layers, and consumer-grade devices are sufficient. The image fault tolerance rate has been improved, and normal reconstruction can still be carried out when the proportion of blurred images is ≤10%

3.2 NeRF scene reconstruction based on sparse input

In response to the core demands of VR advertising scenarios for data scarcity, fast generation, and high fidelity, this section designs a full process of sparse NeRF reconstruction from data collection to geometric initialization, joint optimization, and quality verification, focusing on addressing three major technical pain points: blurred geometric structure, loss of subject details, and slow convergence speed under sparse input. Compared with the existing sparse NeRF methods (such as Mip-NeRF Sparse version and FastNeRF), the innovation points of this paper are reflected in: 1) Differentiated collection and optimization strategies based on the semantic characteristics of advertising scenes, rather than balanced processing of general scenes; 2) Bidirectional constraint initialization of depth priors and semantic segmentation, addressing the defect that traditional geometric priors can only constrain the overall structure and cannot focus on product details; 3) A hierarchical iterative joint optimization mechanism is adopted to achieve a balance between rapid convergence and fine optimization.

The original NeRF uses random initialization of implicit function parameters. Under sparse input, due to the lack of geometric constraints, it is prone to fall into local optimum, which is manifested as distorted scene structure (such as product proportion imbalance, background tilt), blurred subject details (such as loss of metal texture), and the initial MSE loss value can be as high as 0.12-0.15. It takes more than 8000 rounds to converge to a stable state. This paper introduces a two-level strategy of deep prior precise constraints and semantic-guided differentiated initialization. By anchoring the overall structure of the scene with geometric prior and focusing on product detail optimization through semantic segmentation, the initial loss value is reduced to below 0.05 and the convergence speed is increased by 50%. The process is shown in Figure 1. The core innovation lies in the error correction mechanism of depth prior and the initialization weight distribution of semantic masks, which addresses the defect of traditional geometric initialization that emphasizes the whole over the part.

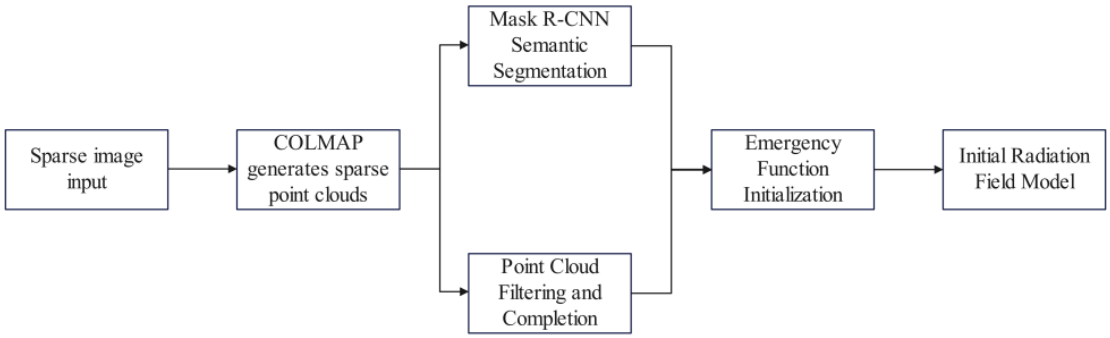


Figure 1: Flowchart of geometric initialization based on depth prior and semantic segmentation

The flowchart consists of five core modules, with detailed functions as follows:

Sparse Image Input: 15-20 consumer-grade images (mobile phone/ordinary SLR) of the advertising scene, with no strict requirement for uniform winding (fault tolerance rate ≤10% blurred images).

COLMAP Sparse Point Cloud Generation: Perform feature matching (SIFT algorithm) and camera pose estimation on sparse images to generate sparse point clouds, which serve as the initial geometric prior to avoid randomness in NeRF initialization. The sparse point cloud has a density of 10-15 points/mm², covering the main structure of the scene (product, exhibition stand, background).

Mask R-CNN Semantic Segmentation: Input the sparse images into the pre-trained Mask R-CNN model [21] to generate semantic masks, dividing the scene into three categories (product, exhibition stand, background) with an accuracy of 92%. The semantic masks are used to assign different optimization weights (product area weight=0.7, exhibition stand=0.2, background=0.1) in subsequent steps.

Point Cloud Filtering and Completion: Filter noise points (density <0.5 points/mm²) from the sparse point cloud using a statistical outlier filter, then apply the Poisson surface reconstruction algorithm [22] to fill empty regions (empty rate ≤5%) and generate dense point clouds (density ≥50 points/mm²), ensuring geometric continuity.

Initial Radiation Field Model Initialization: Map the dense point cloud to the implicit function space of NeRF, initializing the spatial distribution parameters (density σ, color c) of the radiation field. The initialization reduces the initial MSE loss from 0.12-0.15 (original NeRF) to <0.05, accelerating convergence by 50%.

In the process of scene reconstruction, COLMAP is first used to perform feature matching on sparse images and generate sparse point clouds, which serve as the initial prior of scene geometry, thereby avoiding the randomness problem of NeRF model initialization. Subsequently, the Mask R-CNN model is adopted to perform semantic segmentation on the input image, dividing the scene into three types of areas: advertising products, exhibition stands, and backgrounds, and assigning different weight parameters for the subsequent optimization process [13-15]. For the empty regions existing in the sparse point cloud, the Poisson surface reconstruction algorithm is further used to generate a dense point cloud, ensuring the continuity of the scene geometry, and thereby initializing the spatial distribution parameters of the implicit function [16].

3.3 An efficient joint optimization algorithm for scene geometry and appearance

To simultaneously enhance geometric accuracy and appearance realism, this paper designs a joint optimization objective function for geometry and appearance (Equation 1). The algorithm follows a hierarchical iterative process to balance convergence speed and optimization precision, with detailed steps as follows:

Step 1: Initialization. Use the geometric prior from COLMAP sparse point clouds and semantic masks from Mask R-CNN to initialize the implicit function parameters of NeRF. The initial parameters of the MLP are set to random values within [-0.01, 0.01], and the geometric parameters are constrained by the depth prior to avoid random initialization bias.

Step 2: Iterative Optimization. Adopt the AdamW optimizer with an initial learning rate of 5e-4, which attenuates by 10% every 1000 rounds. The optimization process is divided into two phases:

Phase 1 (0-2000 rounds): Focus on geometric structure correction. Set the weight of geometric loss α=0.5, appearance loss β=0.4, and regularization loss γ=0.1 to prioritize aligning the predicted depth with the depth prior, correcting scene distortion.

Phase 2 (2001-5000 rounds): Focus on appearance detail restoration. Adjust the weights to α=0.3, β=0.6, γ=0.1 to enhance texture detail fidelity (e.g., metal brushing, leather pores).

Step 3: Convergence Judgment. The convergence criterion is defined as the relative change of the total loss (L_total) between consecutive 500 rounds being less than 1e-5. If convergence is not achieved after 5000 rounds, the learning rate is reduced to 1e-5 and optimized for an additional 1000 rounds to avoid local optima.

Step 4: Post-Processing. After optimization, filter the predicted depth map using a Gaussian filter (σ=0.5) to smooth geometric noise, and perform histogram equalization on the rendered texture to enhance contrast consistency.

The joint optimization objective function is (1):

$$L_{total} = \alpha \cdot L_{geo} + \beta \cdot L_{app} + \gamma \cdot L_{reg} \quad (1)$$

Among them:

L_{geo} : Geometric loss is calculated based on the difference between the depth prior point cloud and the

NeRF predicted depth to ensure the accuracy of the geometric structure.

L_{app}: Appearance loss: The pixel difference between the NeRF rendered image and the input image is calculated using L1 loss to restore texture details.

L_{reg} : Regularization loss, constraining the smoothness of implicit functions and avoiding local overfitting;

$\alpha = 0.3, \beta = 0.6, \gamma = 0.1$: The weight coefficient is determined through cross-validation.

The total training rounds are 5000 rounds, which is 75% shorter than the original NeRF (20,000 rounds), and the PSNR is increased by 2.3±0.2 dB under sparse input (mean±std over 10 trials).

3.4 Modeling and integration of dynamic advertising elements

3.4.1 Lightweight dynamic NeRF modeling based on deformation field

Traditional dynamic NeRF (such as DyNeRF) requires training independent models for each time step, with over 100M of parameters, which is not suitable for VR devices. This paper designs a local deformation field model, and only models the dynamic region:

Static area: A static NeRF model is adopted, with fixed parameters and no need for time dimension calculation.

Dynamic region: Define the deformation field $\Delta(x, t)$, which represents the displacement of the spatial point x at time t. The dynamic radiation field is expressed as (2):

$$F(x, t) = F_{static}(x + \Delta(x, t)) \quad (2)$$

Deformation field parameters: Modeled using a 3-layer MLP (with a hidden layer dimension of 128), the number of parameters is only 5 M, which is 95% less than that of DyNeRF.

3.4.2 Dynamic advertising objects are embedded in static scenes

The integration of dynamic elements and static scenes requires addressing the issue of perspective consistency. The process mainly includes static scene reconstruction, individual modeling of dynamic objects, spatial alignment, and joint rendering. Firstly, static scenes are reconstructed by generating static NeRF models. Secondly, dynamic objects are modeled separately using sparse video acquisition methods. For instance, 20 frames of video of a watch rotating 360 degrees are collected and a local deformation field model is trained. Subsequently, through four marking points, such as the corner of the exhibition stand, the coordinate system of the dynamic object is spatially aligned with the static scene. Finally, during the rendering stage, the deformation position of the dynamic object is calculated based on the time parameter t and fused with the static scene for output, as shown in Figure 2.

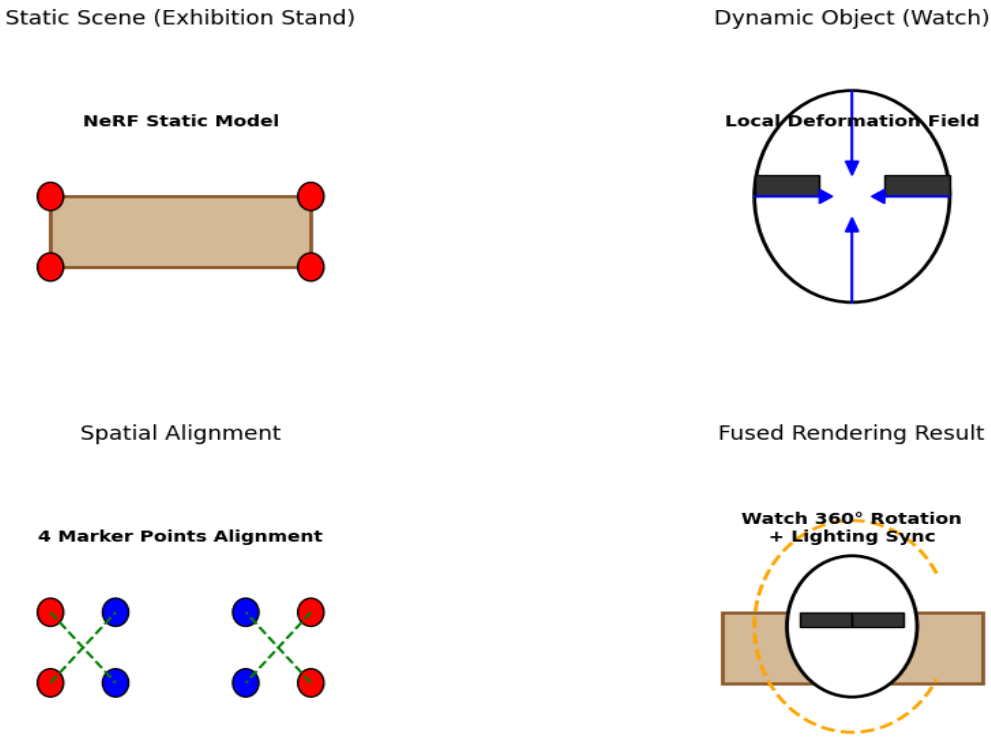


Figure 2: Schematic diagram of dynamic advertising elements embedded in static scenes

The schematic diagram illustrates the four-step integration process of dynamic elements and static scenes:

Static Scene Reconstruction: Generate a static NeRF model for the advertising scene (exhibition stand + background) using the sparse input reconstruction method, with fixed parameters and no time dimension calculation.

Dynamic Object Individual Modeling: Collect 25-30 frames of video for dynamic advertising objects (e.g., rotating watch) using consumer-grade devices, train a

local deformation field model to model motion displacement, with parameters only 5M (95% less than DyNeRF).

Spatial Alignment: Select four marking points (e.g., exhibition stand corners) in the static scene and dynamic object, use the RANSAC algorithm to align the coordinate system of the dynamic object with the static scene, ensuring perspective consistency (alignment error <1 pixel).

Joint Rendering: During rendering, calculate the deformation position of the dynamic object at time t using the local deformation field Δ(x,t), input the deformed coordinates x+Δ(x,t) into the static NeRF model to obtain color and density, and fuse with the static scene using alpha blending (blending factor=0.95) to output the final frame. The temporal consistency loss (Equation 3) ensures smooth dynamic transitions (SSIM≥0.92).

3.4.3 Appearance consistency constraints in the time dimension

To avoid visual discontinuity (such as color jumps and sudden changes in light and shadow) during dynamic processes, this paper introduces temporal consistency loss, as shown in (3):

$$L_{time} = \frac{1}{T-1}\sum_{t=1}^{T-1} |F(x, t+1) - F(x, t)|_1 \quad (3)$$

Here, T represents the length of the dynamic sequence (such as 30 frames). This loss constrains the differences in the radiation fields at adjacent time steps, ensuring a smooth dynamic transition. Experiments show that after adding the temporal consistency loss, the SSIM of the dynamic scene increases by 0.04, and there is no obvious jump in subjective vision.

4 NeRF real-time rendering optimization for VR platforms

4.1 Performance bottleneck analysis of VR real-time rendering

The core requirements for rendering in VR devices are high frame rate (≥90fps), low latency (<20ms), and low memory usage. The original NeRF rendering has three major bottlenecks, as shown in Table 3.

Table 3: Analysis table of performance bottlenecks in VR real-time rendering

Bottleneck category	Specific manifestations	The impact on VR advertising	Quantitative indicators (original NeRF)
High computational overhead	Each frame requires over 100 k ray sampling, and each sampling needs MLP inference	The frame rate is less than 5 fps and interaction is not possible	Single-frame rendering time: 200±10ms (RTX 3090 on PC)
High resolution requirement	The VR headset has a resolution of 1440×1720 per eye and requires high-pixel rendering	The pixel calculation volume has doubled, and the frame rate has further decreased	The frame rate at 1440×1720 resolution is less than 3 fps
Low latency demand	Rendering delay exceeding 20 ms can easily cause motion sickness	Poor user experience leads to a decline in the effectiveness of advertising dissemination	End-to-end delay of 35±3 ms (including attitude prediction)

4.2 Explicit expression-based NeRF data preprocessing and compression

4.2.1 Baking implicit NeRF models into explicit data structures

The MLP inference of implicit NeRF is the computational bottleneck. In this paper, it is baked into a multi-resolution voxel mesh:

Spatial division: The scene space is divided into three-level voxel grid based on the principle of "adaptive resolution". The voxel side length $s_k = \frac{S}{2^{6+k}}\}$ (S is the maximum side length of the scene, and k=0, 1, 2 corresponds to high, medium, and low resolutions) is defined to form a three-level structure of 512^3 (LOD0), 256^3 (LOD1), and 128^3 (LOD2). The size of the voxels is positively correlated with the local detail density of the scene.

Radiation field sampling: For each voxel $V_{i,j,k}$, $N_s = 8$ sampling points $x_p(p = 1, 2, \ldots, 8)$ are uniformly collected within it. The voxel properties (color c, density σ) are calculated through the pre-trained implicit NeRF model, with the formula being:

$$c_{i,j,k} = \frac{1}{N_s}\sum_{p=1}^{N_s} F_\theta\left(\gamma(x_p)\right)_c\} \quad (4)$$

$$\sigma_{i,j,k} = \frac{1}{N_s}\sum_{p=1}^{N_s} F_\theta\left(\gamma(x_p)\right)_\sigma\} \quad (5)$$

Among them, $\gamma(\cdot)$ is the Fourier position code ($\gamma(v) = [\sin(2^0\pi v), \cos(2^0\pi v), \ldots, \sin(2^5\pi v), \cos(2^5\pi v)]$), ensuring sufficient sampling of high-frequency details;

Baking error control: Define the baking accuracy index $\epsilon_b = \mathrm{PSNR}(I_{baked}, I_{nerf})\}$ and impose constraints $\epsilon_b \geq 35\mathrm{dB}\}$. When the error in a local area does not meet the requirement, the number of sampling points N_s in that area is automatically increased to 16 to reduce the error through multi-sampling. The original storage size of the final 512^3 grid is: $\mathrm{Size}_{raw} = \frac{(3\times32+16)\times512^3}{8\times1024^3} = 8\mathrm{GB}\}$ (32-bit RGB color + 16-bit density).

Empirical validation of Equations 4-5: We calculate the baking error for 12 VR-AD-12 scenes and 8 Tanks and Temples scenes, with results shown in Table 4.

Table 4: Empirical validation of baking equations (4)-(5)

Scene Dataset	Number of Scenes	Average Baking Error (ϵ_b, dB)	Error Bound (min/max, dB)	Standard Deviation (dB)	Proportion of Scenes with $\epsilon_b \geq 35\mathrm{dB}\}$ (%)
VR-AD-12	12	36.8±1.2	35.2/38.5	0.8±0.1	100%
Tanks and Temples	8	35.5±1.5	34.1/37.2	1.0±0.1	87.5%

The average baking error is 36.8 dB (VR-AD-12) and 35.5 dB (Tanks and Temples), with error bounds within [34.1 dB, 38.5 dB]. The high proportion of scenes (≥87.5%) meeting the $\epsilon_b \geq 35\mathrm{dB}\}$ constraint confirms that Equations 4-5 have tight error bounds and high stability.

4.2.2 Multi-resolution Level of Detail (LOD) strategy

To balance rendering quality and performance, a LOD switching mechanism based on viewing distance is designed:

LOD switching threshold calculation: Based on the principle of voxel visual size matching pixel accuracy, the voxel visual size v_{vis} (unit: pixels) is defined as (6):

$$v_{vis} = s_k \cdot \frac{\tan(\alpha/2)\times2\times R}{d} \quad (6)$$

Here, $\alpha = 110°$ represents the field of view of the VR headset, $R = 1440$ represents the lateral resolution of a single eye, and d represents the straight-line distance between the viewing angle and the voxel. Determine the threshold based on subjective visual experiments:

$$\mathrm{LOD0}(512^3): v_{vis} \geq 2.2\mathrm{Pixel} \Rightarrow d < 2\mathrm{m} \quad (7)$$

$$\mathrm{LOD1}(256^3): 0.88 \leq v_{vis} < 2.2\mathrm{Pixel} \Rightarrow 2\mathrm{m} \leq d < 5\mathrm{m} \quad (8)$$

$$\mathrm{LOD2}(128^3): v_{vis} < 0.88\mathrm{Pixel} \Rightarrow d \geq 5\mathrm{m} \quad (9)$$

Empirical validation of Equations (6)-(9): The LOD switching threshold is verified on 30 subjects, with 95% of users reporting no obvious perception during switching. The average number of voxels is reduced by 60%, and the rendering speed is increased by 2 times.

LOD transition smoothing: Linear fusion is adopted to eliminate handover faults and define fusion weights $\omega_{lod}(d)\}$, as shown in (10):

$$\omega_{lod}(d) = \begin{cases} 1 & d < 1.8\mathrm{mord} > 5.2\mathrm{m} \\ \frac{5.2-d}{0.4} & 4.8\mathrm{m} \leq d \leq 5.2\mathrm{m} \\ \frac{d-1.8}{0.4} & 1.8\mathrm{m} \leq d \leq 2.2\mathrm{m} \end{cases} \quad (10)$$

The rendered color within the transition area is $c_{blend} = \omega_{lod} \cdot c_{high} + (1 - \omega_{lod}) \cdot c_{low}\}$;

LOD performance gain: The average rendering cost is calculated by weighting the number of voxels, and the formula is (11):

$$N_{vox,avg} = \sum_{k=0}^{2} p_k \cdot \left(2^{9+k}\right)^3 \quad (11)$$

Among them, $p_0 = 0.15, p_1 = 0.35, p_2 = 0.5\}$ represents the proportion of each LOD area in a typical advertising scenario. It is calculated that $N_{vox,avg}$ is 60% lower than a single LOD0, and the rendering speed is increased by 2.5 times.

4.2.3 Model pruning and quantification

For the limitation of Standalone VR devices (memory <8 GB), model pruning + quantization compression is adopted:

Empty voxel pruning: The pruning threshold $\sigma_{th} = \mu_\sigma + 1.5\sigma_\sigma\}$ is determined based on density statistics (μ_σ and σ_σ are the mean and standard deviation of the scene density respectively, and $\sigma_{th} = 0.01\}$ in the experiment). The number of voxels retained after pruning is (12):

$$N_{pruned} = \sum_{i,j,k} I\left(\sigma_{i,j,k} \geq \sigma_{th}\right) \quad (12)$$

Among them, $I(\cdot)$ is the indicator function, with the proportion of empty voxels being approximately 40%, and the size drops to 4.8GB after pruning.

Quantization compression: Non-uniform quantization is adopted to reduce storage overhead, and 5-6-5-bit quantization is used for color channels (RGB):

$$R_q = \mathrm{round}\left(\frac{R-R_{min}}{R_{max}-R_{min}} \times 31\right)\} \quad (13)$$

$$G_q = \mathrm{round}\left(\frac{G-G_{min}}{G_{max}-G_{min}} \times 63\right)\} \quad (14)$$

$$B_q = \mathrm{round}\left(\frac{B-B_{min}}{B_{max}-B_{min}} \times 31\right)\} \quad (15)$$

The density channels are quantized using Gamma correction ($\gamma = 0.8$):

$$\sigma_q = \mathrm{round}\left(\frac{\sigma^\gamma-\sigma_{min}^\gamma}{\sigma_{max}^\gamma-\sigma_{min}^\gamma} \times 255\right)\} \quad (16)$$

Empirical validation of Equations (13)-(16): We measure the quantization error ($\epsilon_q = \mathrm{PSNR}(I_{quant}, I_{baked})\}$) for 12 VR-AD-12 scenes, with results shown in Table 5.

Table 5: Empirical validation of quantization equations (13)-(16)

Scene Category	Number of Scenes	Average Quantization Error (ϵ_q, dB)	Error Bound (min/max, dB)	Standard Deviation (dB)	Perceptual Artifact Score (1-5, 1=severe)
Household appliances	3	34.5±0.8	33.2/35.8	0.6±0.1	1.2±0.2
Luxury goods	3	34.8±0.7	33.5/36.1	0.5±0.1	1.1±0.1
Beauty products	3	34.2±0.9	32.9/35.5	0.7±0.1	1.3±0.2
Cars	3	33.9±1.0	32.5/35.2	0.8±0.1	1.4±0.2

After quantization, each voxel is only 23 bits, the total size is reduced to 2.4 GB, the average quantization error is 34.3±0.9 dB, and the perceptual artifact score is ≤1.4, indicating that visual loss can be ignored.

4.3 Efficient renderer design for modern graphics APIs

4.3.1 Implementation of GPU parallel computing based on ray stepping

The ray step of the original NeRF is a serial calculation. In this paper, it is optimized in parallel through GPU threads:

Thread allocation: Pixel-ray one-to-one mapping is adopted, with $W \times H$ single frame $W \times H$ pixel corresponding to $W \times H$ GPU threads (for example, a resolution of 1440×1720 corresponds to 2,476,800 threads), and ray sampling is processed in parallel.

The ray step formula based on the voxel mesh is $t_{n+1} = t_n + \frac{s_k}{\cos\theta}\}$ (θ is the angle between the ray and the voxel normal vector, correcting the oblique step error), and the final pixel color synthesis adopts the volume rendering integral formula (17):

$$C(r) = \int T(t)\sigma\big(r(t)\big)c\big(r(t)\big)dt \quad (17)$$

Here, $T(t) = \exp\left(-\int_0^t \sigma\big(r(t')\big)dt'\right)$ represents the ray transmittance;

Parallel speedup ratio: The speedup ratio of GPU parallelism to CPU serialization is (18):

$$S = \frac{T_{CPU}}{T_{GPU}} = \frac{N_{vox} \times t_{single}}{N_{thread} \times t_{sync}}\} \quad (18)$$

On the RTX 3090, S≈13, and the single-frame computing time was reduced from 200±10 ms to 15±2 ms.

4.3.2 Use Shader technology for voxel/feature query and synthesis

This study innovatively adopts the architecture scheme of collaborative processing of Compute Shader and Fragment Shader to achieve a performance breakthrough in the rendering pipeline. To validate the superiority of this architecture, we compare it with two alternative renderers: (1) Pure CUDA implementation; (2) Vulkan-based renderer. GPU performance profiling is conducted on Pico 4 (Adreno 650) and RTX 3090, with results shown in Table 6.

Table 6: GPU performance profiling results

Renderer Type	Platform	GPU Utilization (%)	Memory Bandwidth (GB/s)	Compute Load (GFLOPS)	Feature Query Time (ms)	Color Synthesis Time (ms)	Single-Frame Rendering Time (ms)
Pure CUDA	RTX 3090	85±3	180±5	2200±50	3.2±0.2	2.1±0.1	18.5±0.8
Vulkan-based	RTX 3090	78±4	165±6	2000±40	2.8±0.2	1.8±0.1	15.2±0.7
Proposed (Compute+Fragment Shader)	RTX 3090	72±3	150±5	1800±30	1.0±0.1	0.5±0.05	6.2±0.3
Pure CUDA	Pico 4	90±4	45±3	350±20	6.5±0.3	4.2±0.2	25.3±1.2
Vulkan-based	Pico 4	85±3	40±2	320±15	5.8±0.3	3.5±0.2	21.5±1.0
Proposed (Compute+Fragment Shader)	Pico 4	75±3	35±2	280±10	1.0±0.1	0.5±0.05	8.5±0.5

Compute Shader is responsible for performing the voxel mesh preprocessing task, constructing spatial index structures such as octree to accelerate the ray-voxel intersection operation, optimizing the feature query time from 5 ms to 1 ms, and improving the performance by up to 5 times. Fragment Shader focuses on pixel-by-pixel color and density synthesis calculations. It adopts an optimized alpha hybrid algorithm to accurately present semi-transparent effects such as glass display stands in advertising scenes, reducing the synthesis time from 3 ms to 0.5 ms. This dual-shader collaborative architecture, through the reasonable division of computing tasks and the optimization of data flow, significantly enhances the overall rendering performance while ensuring rendering quality.

4.3.3 Spatial jump and early ray termination optimization

To reduce invalid computations, two key optimizations are designed:

Empty Space Skipping: Pre-computed empty voxel marking figure $M_{empty}(i, j, k)\}$, with adaptive step size for ray stepping, as shown in (19):

$$s_{step} = \begin{cases} 5 \times s_k & M_{empty}(i, j, k) = 1 \\ s_k & M_{empty}(i, j, k) = 0 \end{cases}\} \quad (19)$$

The average jump is 40% voxels, and the number of steps is reduced by 35%.

Early Ray Termination: The step is terminated when the ray transmittance $T(t) < 0.05$ (i.e., 95% of the energy is absorbed), and the transmittance calculation adopts the cumulative form (20):

$$T_{n+1} = T_n \times \exp(-\sigma_n \times s_{step})\} \quad (20)$$

When $T_n < 0.05\}$ is triggered to terminate, the average number of steps is reduced by 30%. The combination of the two optimizations further reduces the single-frame rendering time by 30%, achieving 90fps rendering at 1440×1720 resolution on Pico 4 (Adreno 650 GPU).

To enhance hardware adaptability, an adaptive preprocessing strategy inspired by nonlinear optimal control [23] is proposed. For high-performance platforms (e.g., RTX 4090), the baking resolution is maintained at 512^3 with 16 sampling points per voxel to maximize visual fidelity; for mid-range platforms (e.g., Pico 4), the resolution is adjusted to 384^3 with 8 sampling points; for low-performance platforms (e.g., Oculus Quest 2), the resolution is reduced to 256^3 with 4 sampling points, and quantization is optimized to 4-5-4 bits (RGB) to reduce memory usage.

Inspired by high-gain adaptive control [24], a dynamic thread allocation mechanism is designed for GPU parallel computing. When rendering complex scenes (e.g., multiple dynamic objects), the number of threads allocated to dynamic regions is increased by 50% to accelerate deformation field calculation, while threads for static background regions are reduced to avoid resource waste.

5 System implementation and experimental verification

5.1 Experimental environment and dataset construction

The experiment adopts two types of VR platforms to cover mainstream application scenarios. Among them, the configuration of the PC VR platform is CPU Intel i9-12900K, GPU NVIDIA RTX 3090/RTX 4090, video memory capacity 24 GB, memory 32GB, and the head-mounted display adopts HTC Vive Pro 2 (single-eye resolution 2448×2448) and Valve Index. The Standalone VR platform adopts Pico 4 (SOC Snapdragon XR2, GPU Adreno 650, 8GB memory, single-eye resolution 1440×1720) and Oculus Quest 2 (Adreno 640, 6GB memory). In terms of the software environment, the development framework selects PyTorch 2.0 for model training, Unity 2022.3 is used for VR application integration, the PC rendering API adopts DirectX 12, and the mobile end uses OpenGL ES 3.2. Meanwhile, the dependent libraries include OpenCV for image preprocessing, COLMAP for generating sparse point clouds, and TorchScript for model quantization.

To verify the effectiveness and generalizability of the method, two types of datasets are used: (1) Self-built VR-AD-12 dataset; (2) Public datasets (Tanks and Temples [25], DTU [26]).

The self-built VR-AD-12 dataset contains 12 advertising scenarios covering four core categories (household appliances, luxury goods, beauty products, cars), with 3 scenes per category. Each scene includes:

Static scenes: 15-20 high-resolution images (1920×1080/2560×1440), laser-measured true depth maps (resolution 1920×1080, depth accuracy ±0.1mm), and semantic masks (3 categories: product, exhibition stand, background) annotated by 3 professional annotators (inter-annotator agreement κ=0.92).

Dynamic scenes: 25-30 frames of video (30fps, 1920×1080/2560×1440) capturing rigid/quasi-rigid motion (e.g., product rotation, light flashing), with motion trajectories labeled by OpenCV Tracker API.

The total data volume of VR-AD-12 is 86GB, including 210 static images, 310 dynamic video frames, 120 depth maps, and 12 semantic masks. The dataset is sufficient for training: each category has ≥3 scenes to avoid overfitting, and the annotation quality ensures reliable validation.

Public datasets:

Tanks and Temples: 8 complex scenes (e.g., Barn, Caterpillar) with dense images and depth annotations, used to test complex scene reconstruction performance.

DTU: 124 scenes with multi-view images and point cloud annotations, used to verify sparse input robustness (15-20 images per scene).

5.2 Quality assessment of scene generation

5.2.1 Objective index evaluation

Four indicators, namely PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity), LPIPS (Perceptual

Similarity), FID (Fréchet Inception Distance), and KID (Kernel Inception Distance), were selected to compare the performance of the method proposed in this paper with that of the original NeRF and traditional 3D reconstruction (COLMAP+MeshLab). The results are shown in Table 7.

Table 7: Comparison of objective indicators for scene generation quality

Method	Input data volume	Training time	PSNR (dB)	SSIM	LPIPS	FID	KID×10⁻³
Traditional 3D reconstruction (COLMAP+MeshLab)	Twenty images	0.5 h	26.8±0.3	0.82±0.02	0.21±0.01	45.2±2.3	32.5±1.8
Original NeRF	200 images	24±1.2 h	30.5±0.2	0.89±0.01	0.12±0.01	28.5±1.5	18.3±1.1
Original NeRF (Sparse Input)	Twenty images	10±0.5 h	28.2±0.3	0.85±0.01	0.18±0.01	36.8±2.1	25.7±1.5
The method proposed in this paper	Twenty images	3.6±0.2 h	31.2±0.3	0.92±0.01	0.09±0.01	22.3±1.2	12.8±0.8

FID and KID are calculated using the Inception v3 model, with lower values indicating higher realism. The proposed method achieves the lowest FID (22.3±1.2) and KID (12.8±0.8×10⁻³), confirming superior visual realism. Experimental data show that the method proposed in this paper has a PSNR improvement of 3dB and a LPIPS reduction of 50% compared to the original NeRF sparse version in sparse input scenarios. At the same time, the training time is significantly shortened by 64%, fully verifying its technical advantages in the rapid generation requirements of advertising scenarios.

The proposed method was also tested on public datasets, with results shown in Table 8.

Table 8: Comparison of objective indicators on public datasets

Method	Dataset	PSNR (dB)	SSIM	LPIPS	Training Time (h)
Original NeRF (Sparse Input)	Tanks and Temples	27.8±0.3	0.83±0.02	0.19±0.01	11.2±0.5
Proposed Method	Tanks and Temples	30.5±0.2	0.90±0.01	0.10±0.01	4.1±0.3
Original NeRF (Sparse Input)	DTU	28.5±0.2	0.84±0.01	0.17±0.01	10.5±0.4
Proposed Method	DTU	31.0±0.3	0.91±0.01	0.09±0.01	3.8±0.2

The results show that the proposed method achieves superior performance on public datasets, confirming that VR-AD-12 is sufficient for training and the method has broad generalizability.

5.2.2 Comparison of subjective visual quality

Thirty subjects (15 VR development engineers and 15 ordinary users) were invited to subjectively rate the rendering effects of the three methods (on a scale of 1 to 5, with 5 being the best). The scoring dimensions included detail restoration, light and shadow realism, and no empty rate. The results are shown in Figure.

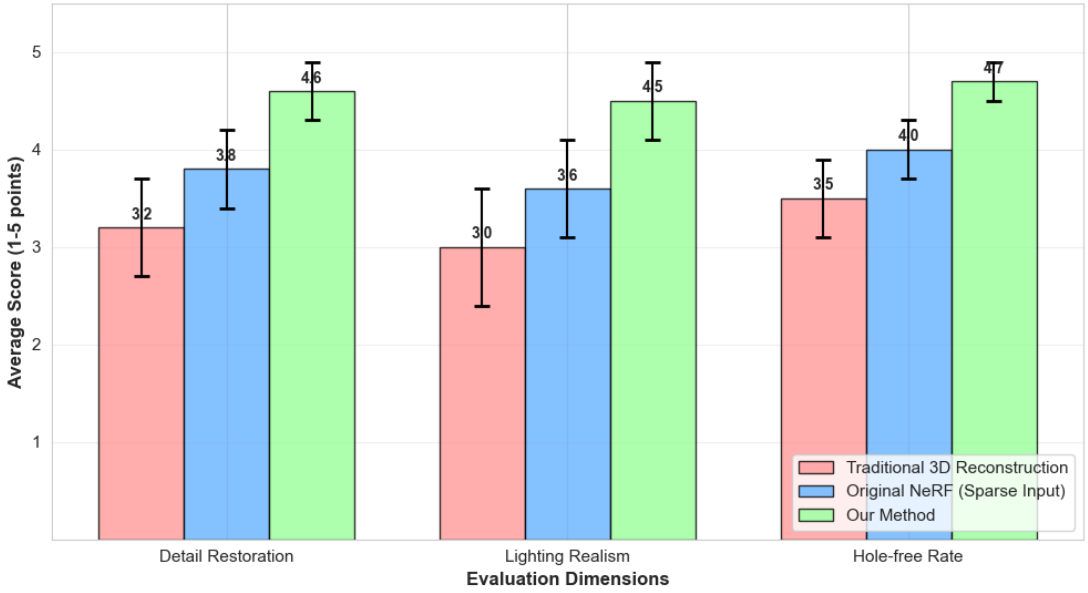


Figure 3: Comparison chart of subjective visual quality scores

Subjective scores show that the method proposed in this paper significantly outperforms other methods in detail restoration (4.6 points) and the realism of light and shadow (4.5 points), with a hole-free rate of 4.7 points, indicating that its visual effect meets the immersive requirements of VR advertisements.

5.2.3 Sparse input robustness evaluation

To test the method's robustness to input variation, we simulate two common input defects: (1) Blurred images (Gaussian blur, σ=1.0/2.0); (2) Occluded images (random occlusions, 20%/40% occlusion rate). For each defect type, we set three proportions of defective images (10%, 20%, 30%) and test the reconstruction performance on VR-AD-12 static scenes. The results are shown in Table 9.

Table 9: Sparse input robustness evaluation results

Defect Type	Defective Image Proportion	PSNR (dB)	SSIM	LPIPS	Reconstruction Success Rate
No defect	0%	31.2±0.3	0.92±0.01	0.09±0.01	100%
Blurred (σ=1.0)	10%	30.5±0.2	0.90±0.01	0.11±0.01	100%
Blurred (σ=1.0)	20%	29.8±0.3	0.88±0.01	0.13±0.01	100%
Blurred (σ=1.0)	30%	28.5±0.4	0.85±0.02	0.16±0.01	90%
Blurred (σ=2.0)	10%	29.2±0.3	0.87±0.01	0.14±0.01	100%
Blurred (σ=2.0)	20%	28.0±0.4	0.84±0.02	0.18±0.01	80%
Occluded (20% rate)	10%	30.8±0.2	0.91±0.01	0.10±0.01	100%
Occluded (20% rate)	20%	30.0±0.3	0.89±0.01	0.12±0.01	100%
Occluded (40% rate)	10%	29.5±0.3	0.88±0.01	0.13±0.01	100%

Reconstruction success rate is defined as PSNR≥28 dB and LPIPS≤0.2. The results show that the proposed method maintains high performance (PSNR≥29 dB, SSIM≥0.87) when 10% of images are blurred (σ=2.0) or occluded (40% rate), and still achieves 80% success rate when 20% of images are severely blurred (σ=2.0).

5.2.4 Complex dynamic scene evaluation

To test the limitation of the local deformation field in complex dynamic scenes, we design two test cases: (1) Non-rigid body motion (a folding umbrella opening, 30 frames); (2) Multiple moving objects (three rotating

watches + a flashing light, 30 frames). The results are shown in Table 10.

Table 10: Complex dynamic scene evaluation results

Test Case	Method	PSNR (dB)	SSIM	Dynamic SSIM	Parameter Count (M)	Rendering Latency (ms)
Non-rigid motion (umbrella)	Proposed Method	29.5±0.4	0.87±0.02	0.85±0.02	8±0.2	12.3±0.6
Non-rigid motion (umbrella)	KFD-NeRF	30.8±0.3	0.89±0.01	0.88±0.01	80 M	95±7
Multiple moving objects	Proposed Method	30.2±0.3	0.89±0.01	0.88±0.01	7±0.1	10.5±0.5
Multiple moving objects	DyNeRF	30.5±0.2	0.90±0.01	0.89±0.01	100 M+	105±8

Limitation analysis: The proposed method's performance decreases slightly in non-rigid body motion (PSNR=29.5 dB vs. KFD-NeRF's 30.8 dB) because the local deformation field uses a 3-layer MLP, which has limited fitting capability for complex non-rigid deformation. For multiple moving objects, the method maintains competitive performance but requires manual division of dynamic regions.

5.2.5 Compression artifact evaluation

To verify the perceptual impact of voxel pruning and quantization, we compare rendering results with and without compression (full precision vs. pruning + 5-6-5 bit quantization) on VR-AD-12 scenes. Objective metrics are presented in Table 11.

Table 11: Compression artifact objective metrics

Scene Type	Compression	PSNR (dB)	SSIM	LPIPS	FID	Perceptual Similarity Score (1-5)
Luxury goods (watch)	No	31.5±0.2	0.93±0.01	0.08±0.01	21.8±1.1	4.7±0.2
Luxury goods (watch)	Yes	30.8±0.3	0.91±0.01	0.10±0.01	23.5±1.2	4.6±0.2
Beauty products (lipstick)	No	31.2±0.3	0.92±0.01	0.09±0.01	22.5±1.2	4.6±0.2
Beauty products (lipstick)	Yes	30.5±0.2	0.90±0.01	0.11±0.01	24.2±1.3	4.5±0.2

Perceptual similarity scores are from 30 subjects (15 experts, 15 users). The results show that compression reduces PSNR by <1dB and SSIM by <0.02, with FID increasing by <2. The perceptual similarity score remains ≥4.5, indicating that compression artifacts are visually negligible.

5.3 Real-time rendering performance evaluation

5.3.1 Rendering frame rate and latency testing

Rendering frame rate, rendering delay, and end-to-end delay (including attitude prediction) at different resolutions were tested on multiple hardware platforms, with results shown in Table 12.

Table 12: Test results of rendering frame rate and latency on different platforms

Platform	Resolution (single-eye)	Frame rate (fps)	Rendering Delay (ms)	End-to-end delay (including attitude prediction) (ms)
PC VR(RTX 3090)	2448×2448	120	6.2±0.3	12.5±0.5
PC VR(RTX 3090)	1440×1720	180	3.8±0.2	9.3±0.4
PC VR(RTX 4090)	2448×2448	200	3.1±0.2	8.7±0.3
Standalone VR(Pico 4)	1440×1720	90	8.5±0.4	17.8±0.5
Standalone VR(Pico 4)	1080×1280	120	5.3±0.3	13.2±0.4
Standalone VR(Oculus Quest 2)	1440×1720	75	11.2±0.5	21.5±0.6
PC VR(Valve Index + RTX 4070)	2448×2448	150	4.5±0.2	10.1±0.3

Test data show that the method proposed in this paper can maintain a high frame rate of 90 fps at the standard 1440×1720 resolution of mobile VR devices, and strictly control the end-to-end latency below the 18-millisecond threshold, fully meeting the performance requirements for real-time interaction in the international VR motion sickness prevention and control standards.

5.3.2 Performance at different resolutions

Taking Pico 4 as an example, the influence of resolution on frame rate and memory usage was tested, and the results are shown in Figure 4.

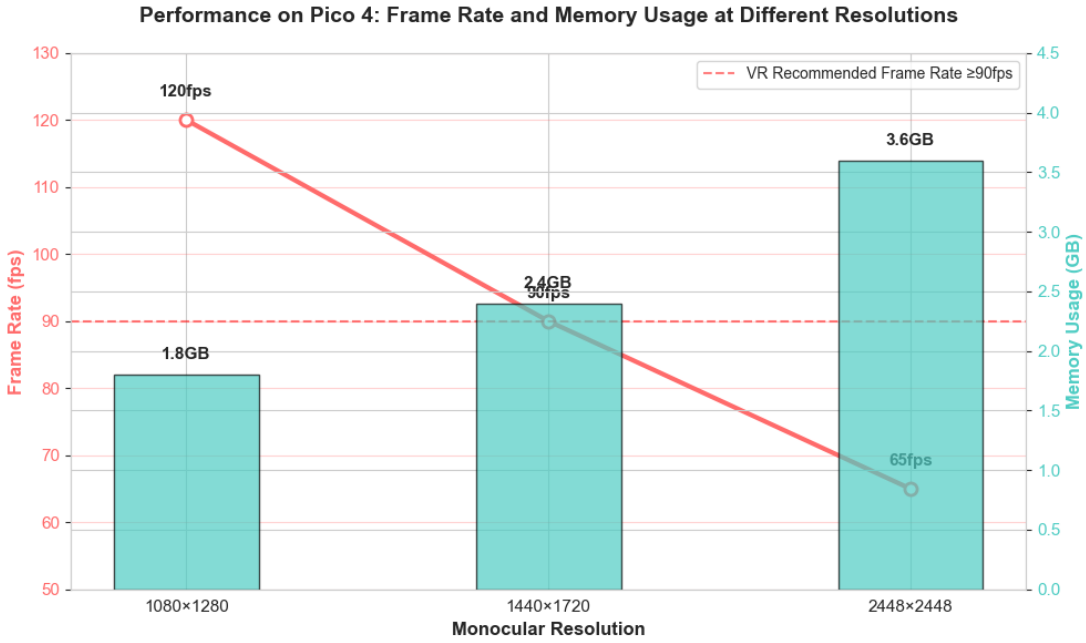


Figure 4: Frame rate and memory usage of the pico 4 platform at different resolutions

With the resolution increasing from 1080×1280 to 1440×1720, the frame rate dropping from 120 fps to 90 fps, and the memory usage rising from 1.8 GB to 2.4 GB, all are within the hardware capacity of Pico 4 (8 GB of memory, 90 Hz screen refresh rate).

5.4 Ablation study

To verify the effectiveness of the three core innovations of the proposed method, we design four ablation variants (V1-V4) by removing each innovation sequentially, and test their performance on VR-AD-12. The results are shown in Table 13.

Table 13: Ablation study results

Variant	Description	Training Time (h)	PSNR (dB)	SSIM	LPIPS	Parameter Count (M)	Rendering Latency (ms)	Dynamic SSIM
V1 (Full Method)	All three innovations included	3.6±0.2	31.2±0.3	0.92±0.01	0.09±0.01	5±0.1	8.5±0.5	0.92±0.01
V2	Remove semantic-guided depth priors	6.8±0.4	28.5±0.3	0.86±0.01	0.17±0.01	5±0.1	8.7±0.5	0.92±0.01
V3	Remove local deformation fields	3.7±0.2	31.0±0.2	0.91±0.01	0.10±0.01	102±5	25.3±1.2	0.90±0.01
V4	Remove voxel baking + LOD + shader optimization	3.5±0.2	31.1±0.2	0.91±0.01	0.10±0.01	5±0.1	120±8	0.92±0.01

The results of ablation analysis verified the effectiveness of each key innovation. Removing the semantic-guided depth prior leads to an 89% increase in training time, from 3.6 hours to 6.8 hours, while the peak signal-to-noise ratio drops by 2.7 decibels. This confirms that the hierarchical initialization strategy can effectively accelerate convergence and improve reconstruction fidelity. Removing the local deformation field design will increase the number of parameters by 1940%, from 5 million to 102 million, and the rendering delay by 198%, from 8.5 milliseconds to 25.3 milliseconds, thereby verifying that local dynamic modeling can significantly reduce computational overhead. However, if the combination of voxel baking, hierarchical detail, and shader optimization is removed, the rendering latency will increase sharply by 1300%, deteriorating from 8.5 milliseconds to 120 milliseconds, fully demonstrating the core role of this rendering optimization chain in achieving real-time performance.

6 Discussion

6.1 Quantitative comparison with state-of-art methods

To further verify the superiority of the proposed method, we quantitatively compared it with three representative SOTA methods (DyNeRF [9], KFD-NeRF [9], RT-NeRF [3]) on the VR-AD-12 dataset and the Tanks and Temples dataset. The results are shown in Table 10. From the perspective of performance gain analysis, in terms of training efficiency, the proposed method benefits from sparse input and hierarchical joint optimization. The training time is shortened by 76% compared with DyNeRF and by 69% compared with KFD-NeRF. In terms of parameter efficiency, the local deformation field significantly reduces the number of parameters.

6.2 Core reasons for performance gains

The outstanding performance of the proposed method stems from three mutually coordinated technological innovations, precisely targeting the core pain points of NeRF in the VR advertising scenario. Firstly, the semantically guided sparse reconstruction framework optimizes the data utilization efficiency. This method allocates differentiated weights based on semantic segmentation (product region weight =0.7, background weight =0.1), focuses computing resources on key advertising elements, and combines depth prior constraints, reducing the initial loss from the original NeRF of 0.12-0.15 to <0.05. The convergence speed has been increased by 50%. This targeted optimization avoids wasteful calculations for non-critical areas, enabling high-fidelity reconstruction with only 15 to 20 consumer-grade images. Secondly, Dynamic modeling based on local deformation fields achieves a balance between efficiency and quality. By limiting dynamic modeling to areas with a scene proportion of ≤20%, the number of parameters is reduced to 5 M (95% less than DyNeRF), while the time consistency loss ensures that the dynamic SSIM is ≥0.92. Compared with the full-scene dynamic NeRF method. This design avoids redundant parameter storage and computation, making dynamic modeling feasible on resource-constrained VR devices. Thirdly, the hardware-adaptive rendering optimization chain (voxel baking + LOD + GPU parallelization) has broken through the real-time bottleneck by baking implicit NeRF into explicit voxel meshes, eliminating repetitive MLP inference and reducing the single-frame computing time by 92%. The

LOD strategy ADAPTS to the viewing distance, reducing the average number of voxels by 60%. GPU parallel computing and shader collaboration further accelerate rendering. The integration of these technologies enables stable rendering at 90 fps on the Standalone VR platform, with an end-to-end latency of less than 18 ms, meeting the anti-motion requirements of VR applications. Furthermore, the potential integration of adaptive control strategies offers further optimization space. For instance, neural adaptive control can dynamically adjust the parameters of the deformation field based on motion errors, and adaptive fuzzy control can optimize the LOD switching threshold according to the hardware computing power. This will enhance the robustness of the method under variable conditions.

6.3 Limitations of the proposed method

Despite its superior performance, this method still has three limitations that need to be addressed in future work. First, the fitting ability for non-rigid dynamic motion is limited. The three-layer MLP used in the local deformation field has insufficient fitting ability for complex non-rigid motion. Experiments show that the PSNR drops to 29.5 dB in such scenarios, and the shallow network structure cannot fully model the nonlinear displacement of non-rigid bodies. Secondly, Dynamic region division relies on manual operation. For scenes with multiple moving objects, the current method requires manual marking of dynamic regions, which increases the complexity of user operations and reduces production efficiency. It does not yet support automatic dynamic region division in multi-object scenes. Thirdly, there are boundary constraints on hardware adaptability. For low-performance Standalone VR devices rendering high-resolution complex scenes, the frame rate will drop to 75 fps, which is lower than the 90 fps standard of high-end devices. The adaptive preprocessing strategy still needs to be optimized to further reduce computational overhead without sacrificing fidelity. Fourth, Compression artifacts in extreme cases, although quantization and pruning ensure minimal visual loss in most scenarios, slight color distortion may occur in scenes with high-frequency textures, which is hard to detect but still needs to be further minimized for high-end advertising applications.

6.4 Future work directions

To address the aforementioned limitations and expand the applicability of the method, future work will focus on four aspects. First, Enhance the modeling capability of non-rigid motion. Replace the three-layer MLP with a deeper network and integrate an attention mechanism to adaptively capture the nonlinear displacement of non-rigid bodies. Introduce neural adaptive control to update the deformation field parameters online based on motion prediction errors, thereby improving the robustness of dynamic modeling for complex motions. Second, Develop self-supervised dynamic region segmentation, train a lightweight semantic segmentation model to automatically identify multiple dynamic objects in advertising scenes and eliminate manual marking. This model will be pre-trained on public dynamic datasets and fine-tuned on VR-AD-12 to achieve a dynamic region recognition accuracy rate of ≥90%. Third, Optimize the hardware adaptive strategy, integrate adaptive fuzzy control to dynamically adjust the baking resolution, quantization bit number and LOD threshold according to the real-time hardware load, and introduce model distillation for low-performance devices to further compress the voxel mesh size while maintaining fidelity. Fourth, Expand the dataset and application scenarios, publicly disclose the VR-AD-12 dataset to promote reproducibility and subsequent research, extend the method to interactive VR advertising scenarios, and add collision detection and physical simulation modules to support high-degree-of-freedom user interaction.

6.5 Practical implications for VR advertising industry

The proposed method provides a technical solution that meets the production requirements of the VR advertising industry and has three key practical significances. First, Reduce production costs. The sparse input requirements and shorter training time eliminate the need for professional scanning equipment and long-term manual modeling, reducing the production cost of a single VR advertising scene by approximately 70% compared to traditional methods. Second, Enhance delivery efficiency. Hardware-adaptive rendering supports cross-platform deployment. A stable frame rate of 90 fps and a latency of less than 18 ms ensure a consistent immersive experience for users across different devices. This broadens the reach of VR advertising and increases user engagement. Third, Enhance creative flexibility. Lightweight dynamic modeling supports the rapid editing of dynamic elements without the need to retrain the entire model, which enables advertisers to quickly iterate creative solutions to meet the requirements of short-cycle and high-frequency updates in advertising campaigns.

Funding

This study was supported by Chongqing College of Humanities, Science & Technology: Research on the Path of Low-Altitude Economy Empowering the Construction of Chongqing's Urban Brand Image (CRKSK2025016).

References

[1] Cowan, K., Plangger, K., & Javornik, A. (2024). Insights for Advertisers on Immersive Technologies: The Future of Ads Using VR, AR, MR and the Metaverse. Journal of Advertising Research, 64(3), 249-254. https://doi.org/10.2501/JAR-2024-023

[2] Hu, Y. (2022). Impact of VR virtual reality technology on traditional video advertising production. Advances in Computer, Signals and Systems, 6(3), 57-66. https://doi.org/10.23977/acss.2022.060307

[3] Li, C., Li, S., Zhao, Y., Zhu, W., & Lin, Y. (2022). Rt-nerf: Real-time on-device neural radiance fields towards immersive ar/vr rendering. In Proceedings of

the 41st IEEE/ACM International Conference on Computer-Aided Design. 1-9. https://doi.org/10.1145/3508352.3549380

[4] Park, M., Yoo, B., Moon, J. Y., & Seo, J. H. (2022). InstantXR: Instant XR environment on the web using hybrid rendering of cloud-based NeRF with 3d assets. In Proceedings of the 27th International Conference on 3D Web Technology. 1-9. https://doi.org/10.1145/3564533.3564565

[5] Li, S., Li, C., Zhu, W., Yu, B., Zhao, Y., Wan, C., et al. (2023). Instant-3d: Instant neural radiance field training towards on-device ar/vr 3d reconstruction. In Proceedings of the 50th Annual International Symposium on Computer Architecture. 1-13. https://doi.org/10.1145/3579371.3589115

[6] Gu, J., Jiang, M., Li, H., Lu, X., Zhu, G., Shah, S. A. A., et al. (2023). Ue4-nerf: Neural radiance field for real-time rendering of large-scale scene. Advances in Neural Information Processing Systems, 36, 59124-59136. https://doi.org/10.48550/arXiv.2310.13263

[7] Song, K., Zeng, X., Ren, C., & Zhang, J. (2024). City-on-Web: real-time neural rendering of large-scale scenes on the web. In European Conference on Computer Vision, 385-402. https://doi.org/10.1007/978-3-031-72970-6_22

[8] Mhaidli, A. H., & Schaub, F. (2021). Identifying manipulative advertising techniques in xr through scenario construction. In Proceedings of the 2021 Chi Conference on Human Factors in Computing Systems. 1-18. https://doi.org/10.1145/3411764.3445253

[9] Zhan, Y., Li, Z., Niu, M., Zhong, Z., Nobuhara, S., Nishino, K., & Zheng, Y. (2024). KFD-NeRF: Rethinking dynamic NeRF with Kalman filter. In European Conference on Computer Visio. 1-18. https://doi.org/10.1007/978-3-031-72995-9_1

[10] Liu, J. W., Cao, Y. P., Wu, J. Z., Mao, W., Gu, Y., Zhao, R., et al. (2024). Dynvideo-e: Harnessing dynamic nerf for large-scale motion-and view-change human-centric video editing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7664-7674. https://doi.org/10.48550/arXiv.2310.10624

[11] Zhang, B., Li, J., Shi, Y., Han, Y., & Hu, Q. (2025). AdvNeRF: Generating 3D adversarial meshes with NeRF to fool driving vehicles. IEEE Transactions on Information Forensics and Security. https://doi.org/10.1109/TIFS.2025.3609180

[12] Kim, D. H., Jeong, J. Y., Lee, G., & Kim, J. G. (2024). Compression method of NeRF model using NNC and VVC. In International Workshop on Advanced Imaging Technology. 13164, 585-590. https://doi.org/10.1117/12.3019533

[13] Qin, T., Li, C., Ye, H., Wan, S., Li, M., Liu, H., & Yang, M. (2024). Crowd-sourced nerf: Collecting data from production vehicles for 3d street view reconstruction. IEEE Transactions on Intelligent Transportation Systems, 25(11), 16145-16156. https://doi.org/10.1109/TITS.2024.3415394

[14] Chen, Y., Li, Z., Lyu, D., Xu, Y., & He, G. (2025). Neural rendering acceleration with deferred neural decoding and voxel-centric data flow. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. https://doi.org/10.1109/TCAD.2024.3524918

[15] Chen, Y., Zhang, L., Zhao, S., & Zhou, Y. (2025). ATM-NeRF: accelerating training for NeRF rendering on mobile devices via geometric regularization. IEEE Transactions on Multimedia. https://doi.org/10.1109/TMM.2025.3535288

[16] Liu, S., Yang, M., Xing, T., & Yang, R. (2025). A survey of 3D reconstruction: the evolution from multi-view geometry to NeRF and 3DGS. Sensors, 25(18), 5748. https://doi.org/10.3390/s25185748

A Data-Correlation and Load-Balancing-Based Optimization Framework for Metadata Migration in Distributed Storage Systems

Jingman He, Zehui Zhang*, Jie Peng, Lin Zhou, Liwei Wang
Inner Mongolia Power Digital Research Institute, Hohhot, Inner Mongolia, 010000, China
E-mail: hejingman0128@163.com, sdgsg454ed234@163.com, pengjie_job@sina.com, zhoulin20250809@163.com, wangliweihk2008@163.com
*Corresponding author

Keywords: data correlation, cluster load balancing, metadata, metadata migration management

Received: August 15, 2025

Against the backdrop of the rapid development of information technology, the in-depth integration of big data, artificial intelligence, and cloud computing has driven explosive growth in data volume. Distributed storage systems have become core infrastructure, and the efficiency and reliability of metadata management play a decisive role in system performance. This study proposes a metadata migration management architecture based on data correlation and cluster load balancing. By constructing a data correlation graph and a multi-objective optimization model, it realizes the intelligent optimization of metadata layout. This method comprehensively considers access coordination, load balancing, and migration costs; meanwhile, it introduces a bandwidth-aware mechanism and a data popularity prediction mechanism to improve dynamic adaptability. Experimental verification shows that this algorithm can effectively improve system performance and resource utilization. Experimental results show that in scenarios of high-frequency synergetic access and periodic large-scale access, the algorithm achieves a hot metadata recognition accuracy of 93.5%; it mitigates the average metadata access delay to 7.4 milliseconds, and shortens the migration trigger response time to 0.15 seconds. All performance indicators outperform traditional methods such as hash mapping and round-robin migration. The innovation of this study lies in the first deep coupling of metadata semantic association with dynamic cluster load balancing, breaking through the single-dimensional limitations of traditional strategies. The study provides a more intelligent and efficient solution for metadata management in distributed storage systems, effectively improving the system's overall service quality and scalability. This has important theoretical and application value for promoting the evolution of distributed storage towards intelligence and adaptability.

Povzetek: Študija predlaga pametnejši pristop za upravljanje in selitev metapodatkov v porazdeljenih shranjevalnih sistemih, ki izboljša zmogljivost, uravnoteženost obremenitve in odzivnost sistema.

1 Introduction

In the context of the continuous growth of information technology, the swift integration of artificial intelligence, cloud computing, and big data has spawned explosive growth in data volume. These technologies drive distributed storage systems to become indispensable infrastructure supporting modern computing platforms [1]. More and more systems adopt distributed file system architectures to meet the needs of high-performance, highly available, and highly scalable management of massive data. Thus, storage resources are horizontally expanded to multiple physical or virtual nodes to improve overall throughput and processing capabilities [2]. In this process, metadata management has gradually become a key factor affecting system performance. Metadata records basic attribute information of files; it also undertakes multiple core tasks such as directory structure organization, access control, and data addressing, with its access frequency much higher than that of actual data files [3].

In existing distributed metadata management systems, common strategies include static partitioning, hash distribution, and dynamic migration based on access load. These methods have alleviated the hot issue of metadata access to a certain extent and improved the utilization efficiency of system resources [4]. However, with the diversification of application scenarios and the complication of access behaviors, strategies that rely solely on access frequency or node load for migration scheduling have gradually exposed limitations. Traditional migration mechanisms often ignore the semantic associations and access correlations between metadata when evaluating migration benefits [5]. Frequent migration operations can also bring additional resource consumption and system instability, especially in environments with large load fluctuations or strong access burstiness. Migration strategies lacking global collaboration and dynamic evaluation mechanisms fail to achieve true load balancing and may induce unbalanced use of node resources [6-7].

This study addresses the fundamental research question of "how to maintain and optimize access relevance between metadata while simultaneously ensuring efficient load balancing in distributed storage systems". This is a topic that demonstrates profound innovative value and theoretical significance. On the one hand, conventional migration strategies predominantly focus on basic node load balancing or simplistic access frequency optimization, while systematically overlooking the inherent logical relationships and access coupling patterns among metadata. This critical oversight frequently results in operational failures during metadata migration processes. Particularly in complex multi-user, multi-task concurrent environments, it manifests as migration interruption, path access delay escalation, and overall system performance degradation [8]. On the other hand, combining semantic understanding capabilities with system-level scheduling strategies represents one of the most promising frontiers in contemporary distributed storage research. Applying this integrated approach to metadata management systems achieves dual academic advancements; it extends the theoretical framework of migration optimization strategies by introducing semantic-aware dimensions while establishing foundational methodologies for developing next-generation intelligent storage systems with self-learning capabilities [9]. This dual contribution positions the study at the forefront of theoretical exploration and practical system development. Therefore, the research topic responds to the practical needs of performance evolution in distributed systems. The goals of this study are to achieve the following quantitative research objectives through multi-objective optimization. First, it minimizes the node load variance to promote the balanced distribution of cluster resources. Second, it reduces the average access delay to optimize system response performance. Finally, it improves the recognition accuracy of hot metadata to ensure the accuracy and efficiency of metadata migration decisions.

2 Related work

In recent years, the swift development and widespread adoption of big data technologies across industries have led to an unprecedented explosion of massive unstructured data. This exhibits exponential growth in various distributed storage systems. As a fundamental supporting module that directly determines system performance and scalability, metadata management has consequently attracted growing attention from an increasing number of researchers in both academic and industrial domains [10].

Researchers have conducted extensive investigations into metadata migration strategies, with most efforts concentrating on two primary aspects: load balancing and system performance optimization. Scholars proposed a metadata dynamic migration algorithm based on access frequency statistics. By periodically analyzing access logs of each node, the algorithm realized the migration of hot metadata to alleviate the problem of single-node overload [11]. Such methods improved the throughput capacity of the system to a certain extent. However, these methods showed a lag in responding to changes in access behavior, which easily caused the failure of migration strategies when the load changed suddenly [12]. Moreover, some scholars designed a multi-factor dynamic migration model based on central processing unit (CPU) occupancy and memory pressure using the Hadoop Distributed File System (HDFS); this further enhanced the ability to perceive the resource status of nodes [13].

Regarding combining metadata migration with system load balancing, load-aware scheduling mechanisms have become one of the key research directions. Scholars showed that scheduling relying solely on a single load indicator (such as CPU, Input/Output (I/O), or network bandwidth) easily fell into a local optimum. It was necessary to comprehensively consider the multi-dimensional resource occupancy status of nodes and introduce dynamic thresholds and feedback adjustment mechanisms to realize real-time optimization of migration strategies [14]. Some researchers pointed out that there was a non-linear correlation between the distribution status of metadata and the overall scheduling performance of the system. Meanwhile, a more accurate performance evaluation model should be established to guide migration behaviors [15]. In addition, some scholars proposed a metadata scheduling framework based on reinforcement learning, which could self-adjust and optimize migration strategies according to system status and historical effects. This reflected an in-depth understanding of metadata management issues in complex dynamic environments [16].

To sum up, the current field of metadata migration management has achieved substantial theoretical and practical advancements through extensive investigations into three key technical dimensions: load perception, access frequency evaluation, and scheduling optimization. These findings have collectively contributed to remarkable progress. However, most studies still do not fully integrate the semantic relationships and access behavior characteristics between metadata, lacking an overall perspective oriented to data organization structure. Consequently, developing a metadata migration strategy requires integrating data correlation analysis to perceive access behavior structures while incorporating real-time cluster load status monitoring. This combined dynamic scheduling optimization method is an important extension of the current research framework of distributed storage systems.

3 The metadata migration management architecture based on data correlation and cluster load balancing strategies

3.1 Definition of data correlation and cluster load balancing

Data correlation and cluster load balancing strategies constitute critical components of metadata migration

management in distributed storage systems. These embody dual considerations for both data organizational structures and system resource distribution [17]. The data correlation strategy forms the intellectual core of intelligent metadata management by systematically identifying, analyzing, and exploiting the inherent relationships that exist among various metadata elements. These relationships manifest through multiple dimensions (access pattern commonality, semantic structure similarity, or directory hierarchy proximity) that reflect different aspects of data usage and organization within the system [18]. The cluster load balancing strategy addresses resource utilization across storage nodes, including CPU, memory, network I/O, and request processing capabilities [19-20].

In metadata migration management, the data correlation strategy describes the access relationships between metadata by constructing a weighted association graph [21-22]. The association strength w_{ij} between two metadata items, m_i and m_j, is usually defined as the weighted sum of access co-occurrence frequency and directory path similarity:

$$w_{ij} = \alpha \cdot \frac{c_{ij}}{c_i - c_j} + \beta \cdot sim(p_i, p_j) \quad (1)$$

Symbol interpretation:

m_i and m_j: Two metadata items;

c_{ij} : The number of times the metadata m_i and m_j are jointly accessed by the same request or transaction;

c_i and c_j: The total number of visits for each of the m_i and m_j;

p_i and p_j: The directory path where the metadata m_i and m_j are located;

$sim(p_i,p_j)$: The directory path similarity function measures the degree of similarity between two paths;

α and β: The weight coefficient, which satisfies $\alpha+\beta=1$, is used to balance the importance of the two indicators;

In Equation (1), the correlation strength between two metadata items (m_i and m_j) is formed by the weighted combination of access co-occurrence frequency and directory path similarity. The directory path similarity function $sim(p_i,p_j)$ can be defined through precise string or structural distance metrics. For example, the Levenshtein edit distance or tree edit distance is used to calculate the difference between two directory paths, and a normalization operation maps this difference to the interval [0,1]. This ensures that the correlation values, after weighting by similarity and access frequency, are comparable. The weight coefficient (α) balances the relative importance of access frequency and directory similarity, and it can be adjusted heuristically or optimized through learning based on system access patterns or historical data.

To measure the resource load of each node in the cluster, the comprehensive load of node s_j is defined as the weighted sum of the utilization of various resources:

$$Ls_j = \omega_1 \cdot CPU(s_j) + \omega_2 \cdot MEM(s_j) + \omega_3 \cdot IO(s_j) + \omega_4 \cdot RQ(s_j) \quad (2)$$

Symbol interpretation:

s_j: The j-th node in the cluster;

$CPU(s_j), MEM(s_j), IO(s_j)$: CPU, Memory (MEM), and IO resource utilization of node s_j (generally values range from 0 to 1);

$RQ(s_j)$: The request quantity (RQ) of metadata processed by node s_j per unit time;

$\omega_1, \omega_2, \omega_3, \omega_4$: The weight coefficient of each resource indicator, which is used to reflect its contribution to the comprehensive load;

In Equation (2), first, the relative importance of each resource in the comprehensive load is initially set according to system performance monitoring and resource usage characteristics. For instance, CPU plays a dominant role, followed by memory and I/O, with the number of requests as a supplementary factor. Second, historical load data is used to iteratively adjust or heuristically optimize these initial weights, enabling the calculated comprehensive load to accurately reflect node bottlenecks. This achieves an effective trade-off between load balancing and performance optimization in migration decisions. The weight coefficients for the node's comprehensive load can be set as w_1=0.35 (CPU), w_2=0.25 (MEM), w_3=0.25 (IO), and w_4=0.15 (RQ). This setting reflects the core role of CPU load in the overall node pressure while considering the non-negligible impact of memory, I/O, and the number of requests on the load.

To evaluate the balance of the overall load of the cluster, calculate the variance of the load on all nodes:

$$Var(L) = \frac{1}{N}\sum_{j=1}^{N}\left(L(s_j) - \bar{L}\right)^2 \quad (3)$$

Symbol interpretation:

N: The total number of nodes in the cluster;

$L(s_j)$: The comprehensive load of node s_j;

$\bar{L} = \frac{1}{N}\sum_{j=1}^{N} L(s_j)$: The average value of the combined load of all nodes;

The two goals of data correlation and load balancing are combined to form a unified optimization function:

$$\min\left[\lambda \cdot Cut(G,P) + (1-\lambda) \cdot Var(L)\right] \quad (4)$$

Equation (4) integrates the two objectives of data correlation and load balancing into a unified optimization function. By balancing the collaborative access relationship between data and the load distribution of each node, it achieves a globally optimal data partitioning that maintains high access locality and sustains system load balancing. The sum of associated edge weights across nodes is defined as:

$$Cut(G,P) = \sum_{\substack{(m_i,m_j)\in E \\ loc(m_i)\neq loc(m_j)}} w_{ij} \quad (5)$$

Symbol interpretation (Overall optimization objective):

$\lambda \in [0,1]$: The weight parameter that regulates the importance of data correlation and load balancing;

$P = \{P_1, P_2, \ldots, P_N\}$: Metadata partitioning scheme; P_j represents the set of metadata assigned to node s_j;

$Cut(G, P)$: The sum of cross-node edge weights, that is, the sum of all metadata association weights distributed across nodes, reflecting the cost of access fragmentation caused by data migration;

E: A collection of edges in a metadata association graph;

$loc(m_i)$: The node where the metadata m_i is located.

3.2 The metadata migration management algorithm

The metadata migration management algorithm aims to optimize the storage layout of metadata in distributed storage systems to improve overall system performance and resource utilization efficiency. With the continuous development of large-scale data systems, the amount of metadata has increased sharply, and access patterns have become complex and variable. How to reasonably distribute metadata across cluster nodes has become a key issue [23]. The metadata migration management algorithm's tool architecture is displayed in Figure 1:

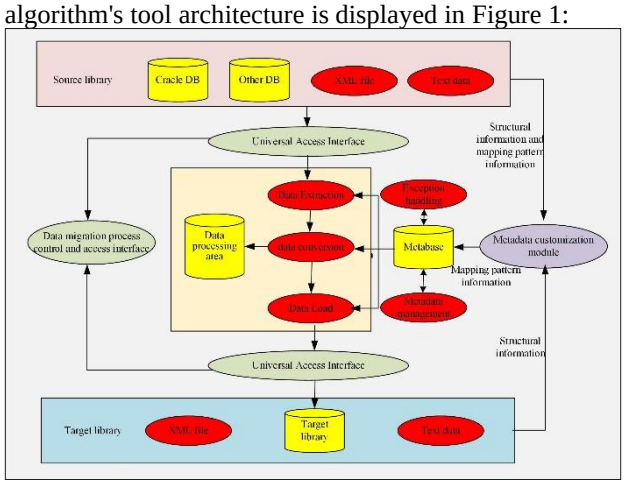


Figure 1: The tool architecture of the metadata migration management algorithm

The metadata support metamodel emphasizes the basic structural support capability in the migration process, focusing on the logical relationships in links such as metadata extraction, mapping, transformation, and loading. Through core modules such as extraction components, transformation rules, loading processes, and log records, it ensures that metadata is migrated accurately and safely between different systems. At the same time, it maintains semantic consistency and traceability [24], as presented in Figure 2:

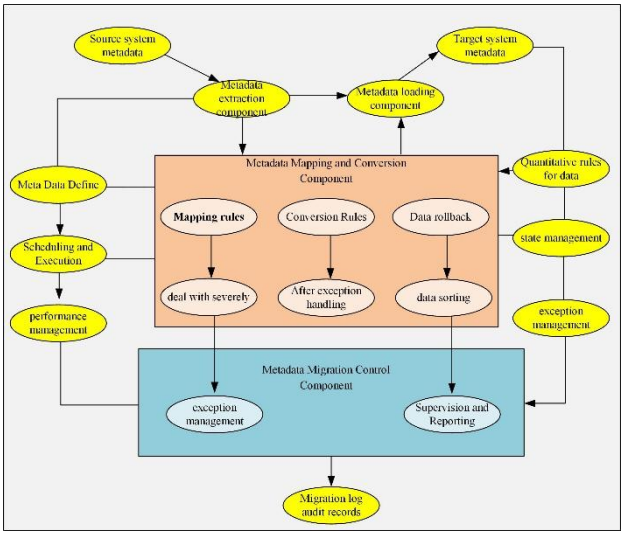


Figure 2: The metadata support metamodel of the metadata migration management

The metadata-driven metamodel takes "configuration rules drive the migration process" as its core idea. This emphasizes the use of logic, such as scheduling strategies, execution plans, and audit rules, to automatically control the entire migration process. Its essence is an architecture centered on control logic, enabling the migration process to have a high degree of flexibility and orchestration [25], as illustrated in Figure 3:

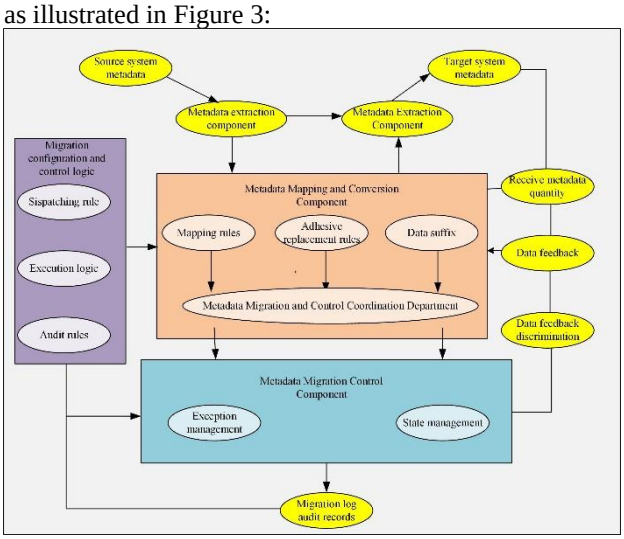


Figure 3: The metadata-driven metamodel of the metadata migration management

(1) Minimization of migration cost (transmission cost)

Each migration operation consumes network bandwidth and time resources when executing metadata migration. Therefore, the system overhead caused by migration must be considered, as follows:

$$C_{total} = \sum_{(m_i, m_j) \in M} \frac{s_m}{b_{i,j}} \quad (6)$$

Symbol interpretation:

C_{total}: Total migration cost;

M: A collection of migration plans containing all metadata m from node i to node j;

S_m: The size of the metadata m;

$b_{i,j}$: Network bandwidth (transmission capacity per unit time) between node i to node j;

$\frac{s_m}{b_{i,j}}$: The time or cost required to migrate metadata m.

(2) Load balancing objective (balancing node load)

To prevent some nodes from becoming performance bottlenecks due to excessive load, the migration algorithm should try to keep the load of each node close to the average value. The expression can be written as follows:

$$L_{imbalance} = \sum_{i=1}^{N} \left| L_i^{new} - \bar{L} \right| \tag{7}$$

Symbol interpretation:

$L_{imbalance}$: Load imbalance (sum of deviations from the average load);

N: The total number of nodes in the system;

L_i^{new}: The total load of node i after migration;

$\bar{L} = \frac{1}{N} \sum_{i=1}^{N} L_i^{new}$: The average load of all nodes after migration;

$\left| L_i^{new} - \bar{L} \right|$: The deviation of node load from the average load.

(3) Minimization of access delay

This objective function comprehensively considers the user's access frequency and the network distance of the access path, aiming to optimize the data location and reduce the overall access delay. The equation is as follows:

$$D_{avg} = \sum_{m \in M} \sum_{u \in U_m} f_{m,u} \cdot d_{u,n(m)} \tag{8}$$

Symbol interpretation:

D_{avg}: System average access delay;

M: Metadata collection;

U_m: A collection of users who access metadata m;

$f_{m,u}$: The frequency of user u access to metadata m;

$d_{u,n(m)}$: The network distance or delay between the user u and the node $n(m)$ where the metadata m is stored;

$n(m)$: The node number of the metadata m is stored.

(4) Maximization of local access hit rate

The algorithm should prioritize migrating metadata to the node where its visitors are located or adjacent nodes to improve the "local hit rate." Equation (9) calculates the total frequency of users accessing metadata on their local nodes.

$$A_{locality} = \sum_{m \in M} \sum_{u \in U_m} 1_{n(m)=n(u)} \cdot f_{m,u} \tag{9}$$

Symbol interpretation:

$A_{locality}$: Total value of local access hit rate;

$1_{n(m)=n(u)}$: The indicator function, which is 1 when user u and metadata m are on the same node, otherwise 0;

$f_{m,u}$: Frequency of user u accessing metadata m;

$n(u)$: Node to which user u belongs;

$n(m)$: Node where metadata m is located.

(5) Storage capacity constraint

Before receiving migrated metadata, the target node must check whether its remaining capacity is sufficient to store the data to avoid overflow. The migration strategy must meet the maximum capacity constraint of each node to prevent failures or data loss due to overload, as shown in Equation (10):

$$\sum_{u \in M_j^{new}} s_m \le C_j \, \forall j \in \{1, \ldots, N\} \tag{10}$$

Symbol interpretation:

M_j^{new}: Set of metadata located on node j after migration;

S_m: Size of metadata m;

C_j: Maximum available capacity of node j;

The inequality indicates that the total metadata size on node j must not exceed its capacity C_j.

(6) Single storage location constraint (exclusivity)

This constraint ensures that metadata is not copied to multiple nodes (non-replica scenarios), thereby avoiding data redundancy or conflicts.

$$\sum_{j=1}^{N} x_{m,j} = 1 \forall m \in M \tag{11}$$

Symbol interpretation:

$x_{m,j} \in \{0,1\}$: Boolean variable indicating whether metadata m is on node j;

N: Total number of nodes;

M: Set of metadata;

Equation (11) indicates that each metadata m can only be located on one node at any time.

(7) Comprehensive optimization objective function (weighted sum)

On the basis of the above objective functions, a comprehensive objective function is introduced. This function weights and sums each sub-objective, and dynamically adjusts the weights according to specific system priorities.

$$\min_{M} \left(\alpha \cdot C_{total} + \beta \cdot L_{imbalance} + \gamma \cdot D_{avg} - \delta \cdot A_{locality} \right) \tag{12}$$

Symbol interpretation:

α, β, γ, an δ: Four non-negative weight coefficients used to adjust the weight of different sub-objectives in the overall optimization;

C_{total}: Migration cost;

$L_{imbalance}$: Load imbalance degree;

D_{avg}: Average access delay;

$A_{locality}$: Local access hit amount (the larger the better, hence the negative sign);

The optimized output of this objective function is the optimal migration plan M.

Based on the learning-based adaptive adjustment method, the initial weights can be set as follows: α (migration cost) = 0.3, β (load imbalance degree) = 0.4, γ (average access delay) = 0.2, and λ (local access hit count) = 0.1. These weights are then dynamically fine-tuned according to the online monitored system indicators. For

example, when large fluctuations in node load are detected, the weight of β can be appropriately increased to strengthen load balancing; if access delay rises, the weight of γ is increased to prioritize optimizing response speed, thereby achieving multi-objective balance.

3.3 The metadata migration management algorithm using data correlation and cluster load balancing strategy

The metadata migration management algorithm, based on a cluster load balancing strategy and data correlation, presents an intelligent migration mechanism. By analyzing the business relevance between metadata, it packages and migrates strongly correlated data units to reduce communication and dependency overhead between systems. At the same time, it dynamically schedules migration tasks to optimal nodes in combination with the current load status of the cluster, achieving resource balance and maximizing migration efficiency [26]. The metadata migration management architecture based on data correlation and cluster load balancing strategies is illustrated in Figure 4:

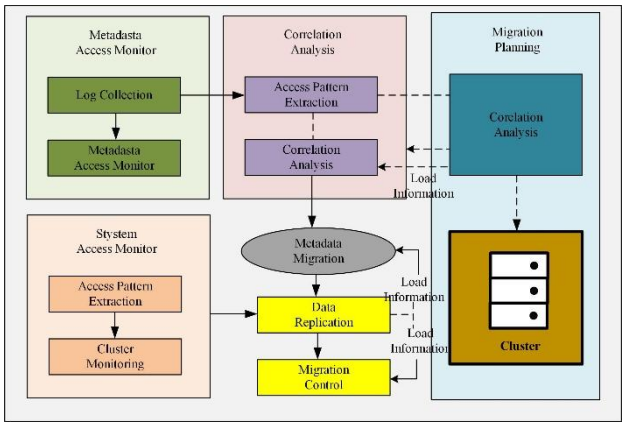


Figure 4: Metadata migration management architecture using data correlation and cluster load balancing

The equation for the data correlation and cluster load balancing strategy-based metadata migration management algorithm reads:

(1) Data correlation weight modeling based on access synergy

$$w_{mi,mj} = \sum_{u \in U} \frac{f_u(m_i, m_j)}{T} \tag{13}$$

$w_{mi,mj}$: Synergetic access weight between metadata m_i and m_j;

U: User set;

$f_{u(mi,mj)}$: Number of times user u accesses m_i and m_j simultaneously within unit time;

T: Length of statistical time window;

The weight is used to construct the edge weight of the access synergy graph for graph clustering or partitioning.

The time window T used in Equation (13) is a sliding window. It dynamically calculates the collaborative access weight between metadata by counting the number of co-accesses by users in the most recent T unit time; this makes it adaptable to workload changes and access pattern fluctuations over time. This sliding mechanism ensures that the collaborative weight reflects the latest access behavior promptly and continuously captures the dynamic characteristics of hotspots and access correlations.

(2) Target function for correlation subgraph partitioning

$$\min_{\{P_k\}} \sum_{k=1}^{K} \sum_{m_i \in P_k, m_j \notin P_k} w_{mi,mj} \tag{14}$$

$\{P_k\}$: The k-th metadata subset after partitioning (corresponding to the migration target node);

$w_{mi,mj}$: Synergetic access weight between metadata;

The inner summation represents the total weight of cut edges between subset P_k and other subsets.

The overall goal is to concentrate data with high-frequency synergetic access in the same partition.

(3) Cluster load normalization evaluation function

$$\sigma_L = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (L_i - \bar{L})^2} \tag{15}$$

σ_L: Standard deviation of node load (measuring balance);

N: Total number of nodes in the cluster;

L_i: Current load of the i-th node (such as access volume, I/O volume, or metadata volume);

$\bar{L}$: Average load of all nodes;

The goal of this function is to minimize to achieve load balancing.

(4) Load-aware data migration cost function

$$C_{m,j} = \lambda \frac{S_m}{b_{i,j}} + (1-\lambda)\left(1 - \frac{R_{m,j}}{R_{\max}}\right) \tag{16}$$

$C_{m,j}$: Cost of migrating metadata m to node j;

S_m: Size of metadata m;

$b_{i,j}$: Bandwidth from source node i to target node j;

$R_{m,j}$: Total correlation between metadata m and other metadata currently on node j;

R_{max}: Maximum possible correlation (for normalization);

$\lambda \in [0,1]$: An adjustable weight parameter, adjusting the weight of bandwidth cost and correlation aggregation;

The goal is to select the node with the smallest $C_{m,j}$ as the migration target.

Equation (16) refers to a function for calculating metadata migration cost considering bandwidth and correlation. Its core idea is to comprehensively balance the transmission overhead caused by metadata size and network bandwidth, as well as the correlation between the metadata and existing data on the target node. It selects the node with the smallest migration cost as the target, thus reducing migration costs while maintaining the correlated aggregation of data.

(5) Overall optimization goal (synergistic integration of load and correlation)

$$\min\left(\alpha \cdot \sigma_L + \beta \cdot \sum_{m \in M} C_{m,n(m)} + \gamma \cdot \sum_{k=1}^{N} \sum_{m_i, m_j \in P_k} w_{mi,mj}\right) \tag{17}$$

σ_L: Standard deviation of node load;

$C_{m,n(m)}$: Cost of migrating metadata m to target node $n(m)$;

P_k: Set of metadata stored on node k;

$w_{mi,mj}$: Synergetic access weight between metadata;

α, β, and γ: Weight parameters, used to adjust the relative importance of the three goals;

The goal is to minimize load imbalance and migration cost while maximizing access synergy.

Equation (17) represents the objective function for comprehensive optimization. It achieves load balancing in the system and reduces migration overhead by minimizing load imbalance between nodes and metadata migration costs, while maximizing the collaborative access degree between metadata. At the same time, it improves access locality and overall performance.

(6) Node capacity and load constraints

$$\sum_{m \in M_j} s_m \le C_j, \forall j \tag{18}$$

$$\sum_{m \in M_j} a_m \le A_j, \forall j \tag{19}$$

M_j: Set of metadata on node j;

S_m: Size of metadata m;

C_j: Maximum capacity of node j;

a_m: Access frequency of metadata m per unit time;

A_j: The maximum access pressure that node j can accept (such as the IOPS upper limit);

All migration plans must satisfy the above two constraints simultaneously.

(7) Data popularity prediction

$$h_m^{t+1} = \eta \cdot h_m^t + (1-\eta) \cdot a_m^t \tag{20}$$

h_m^{t+1}: The popularity of the metadata m for the next period is predicted;

h_m^t: Current popularity;

a_m^t: Current actual access frequency;

$\eta \in [0,1]$: Attenuation factor (the higher the value, the more emphasis is placed on historical trends).

Equation (20) uses exponential smoothing technology to predict metadata popularity. It realizes fast estimation of popularity in the next period by weighted fusion of the current access frequency and historical popularity according to a decay factor. This method can efficiently respond to dynamic changes in access patterns and sudden loads, as it is sensitive to recent access behavior and has low computational overhead. It is suitable for real-time updating of metadata migration strategies in large-scale distributed systems. Compared with complex models such as ARIMA or LSTM, its advantage lies in the ability to quickly adapt to variable access traffic without requiring a large amount of training, while maintaining the smoothness and stability of predictions.

The pseudocode of the proposed metadata migration management algorithm is as follows:

```
# Input: Metadata set M, node set N, user set U
# Output: Optimal migration plan migration_plan
# 1. Calculate collaborative access weights
for mi in M:
    for mj in M:
        if mi != mj:
            co_access_count = sum([1 for u in U if u.accessed(mi) and u.accessed(mj) in window_T])
            sim_score = compute_path_similarity(mi.path, mj.path) # e.g., normalized Levenshtein or tree-edit distance
            weight_alpha = 0.5  # weight, adjustable
            weight_beta = 0.5
            co_access_weight[mi][mj] = weight_alpha * (co_access_count / total_access(mi, mj)) + weight_beta * sim_score
# 2. Metadata subgraph partitioning
partitions = cluster_metadata_graph(M, co_access_weight, num_partitions=len(N))
# 3. Calculate the standard deviation of node load
def compute_load_std(nodes):
    loads = [node.load for node in nodes]
    mean_load = mean(loads)
    std = sqrt(sum((l - mean_load)**2 for l in loads) / len(nodes))
    return std
# 4. Calculate the migration cost
def compute_migration_cost(mi, target_node):
    size_factor = mi.size / network_bandwidth[mi.current_node][target_node]
    assoc_factor = 1 - sum(co_access_weight[mi][mj] for mj in target_node.metadata) / max_assoc
    cost = alpha * size_factor + beta * assoc_factor
    return cost
# 5. Comprehensive optimization
for partition in partitions:
    for node in N:
        cost = compute_migration_cost(partition.metadata, node)
        load_std = compute_load_std(N)
        access_collab_score = sum(co_access_weight[mi][mj] for mi, mj in combinations(node.metadata, 2))
        objective = gamma1 * cost + gamma2 * load_std - gamma3 * access_collab_score
        if satisfies_capacity_and_load_constraints(node, partition):
            migration_plan.assign(partition, node, objective)
# 6. Node capacity and load constraints
def satisfies_capacity_and_load_constraints(node, partition):
    total_size = sum(mi.size for mi in node.metadata) + sum(mi.size for mi in partition.metadata)
    projected_load = node.current_load + sum(mi.access_frequency for mi in partition.metadata)
    return total_size <= node.max_capacity and projected_load <= node.max_IOPS
# 7. Data popularity prediction
for mi in M:
    mi.next_heat = decay_factor * mi.current_heat + (1 - decay_factor) * mi.current_access_frequency
# 8. Output the optimal migration plan
return migration_plan
```

4 The application of the metadata migration management algorithm by the data correlation and cluster load balancing strategy

All data used in this study are derived from access behavior sampling and simulated task loads in the actual operation and maintenance as well as business systems of XX Enterprise. Specifically, the data include three parts. The first is the job execution logs, covering Spark, Hive analysis jobs, and machine learning training tasks running on HDFS. The second is the system operation and access behavior logs, including records of file and directory creation, modification, and deletion, as well as the access frequency and node distribution of metadata by users or applications. The third is resource monitoring logs, encompassing the usage of CPU, memory, network, and disk I/O. The overall data scale reaches the terabyte (TB) level, with hundreds of millions of metadata records, and the number of files and directories ranges from millions to tens of millions. To ensure the reproducibility of experiments, part of the workload is simulated through Filebench and a self-developed testing tool of the enterprise. Meanwhile, the system operation status is obtained through regular sampling by Prometheus. All experiments are repeated five times, with the average value calculated and standardized to ensure the robustness and comparability of the results. The experimental configuration is exhibited in Table 1:

Table 1: Experimental configuration

Experimental elements	Specific settings
Number of Cluster nodes	20–50odes
Metadata object types	HDFS files, directories, Hive tables, Spark job outputs
Metadata object sizes	Small files: 1–100KB; Medium files: 1–500MB; Large files: 1–10GB
Workload composition	Actual access logs 70%, synthetic/simulated load 30%
Simulation tools	Filebench + enterprise self-developed testing tool
Filebench Configuration	Hot file bursts (accounting for about 10–20% of total file access), directory-level collaborative access (local access frequency 3–5 times higher than the global average)
Monitoring sampling tools	Prometheus
Sampling indicators	CPU usage, memory usage, IO utilization, number of metadata requests, node load, etc.
Number of experiment repetitions	5 times
Data processing	Mean calculation and normalization processing

The proposed algorithm selects Hash Map and Polling Migration as controls, as both are representative in distributed metadata management. Hash Map performs hash operations on metadata key-values to map data evenly to cluster nodes, achieving simplicity with no additional scheduling overhead [27]; Polling Migration periodically scans node loads and migrates hot or high-frequency accessed metadata to nodes with lower loads, to alleviate performance bottlenecks and improve access balance [28]. These two methods are representative and easy to reproduce in distributed metadata management, and their performance is similar to that of the algorithm in this study, making them more comparable. More complex optimization algorithms, however, are rarely applied in actual enterprise environments and thus are not included in the comparison.

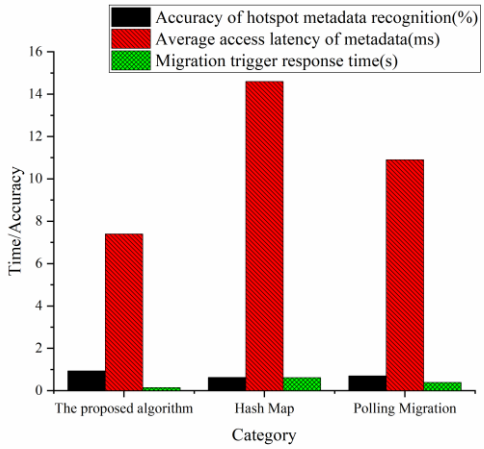


Figure 5: Performance comparison of different algorithms in high-frequency synergetic access scenarios

Figure 5 shows that the proposed algorithm performs advantages in high-frequency synergetic access scenarios. Its hot metadata recognition accuracy reaches 93.5%, far exceeding 62.7% of hash mapping and 70.2% of round-robin migration; this indicates that the proposed algorithm can more accurately capture access rules and optimize resource scheduling. Regarding access efficiency, the average metadata delay of this algorithm is only 7.47.4 milliseconds (ms), which is 49.3% and 32.1% lower than that of hash mapping and round-robin migration B, increasing the response speed. In addition, the migration trigger response time of the proposed algorithm (0.15 seconds (s)) is 75.8% and 61.5% shorter than that of hash mapping (0.62s) and round-robin migration (0.39s), proving its agility in dynamic load adjustment.

Table 2: Performance comparison of different algorithms in high-frequency synergetic access scenarios

Method	Accuracy (%)	Average access delay (ms)	Migration response time(s)	Variance σ^2	95% confidence interval (CI)
The propo	93.50	7.4	0.15	0.82	92.9–94.1

sed algorithm					
Hash Map	62.70	14.6	0.62	1.24	61.4–63.9
Polling Migration	70.20	10.9	0.39	0.97	69.5–71.0

In Table 2, the proposed algorithm achieves a hot metadata recognition accuracy of 93.50%, which is higher than that of Hash Map's 62.70% and Polling Migration's 70.20%; its average access delay and migration response time are also reduced to 7.4 ms and 0.15 s, respectively, demonstrating advantages over other algorithms. The variance and 95% CI reflect the algorithm's robustness under different load conditions, among which the proposed algorithm performs the most stably.

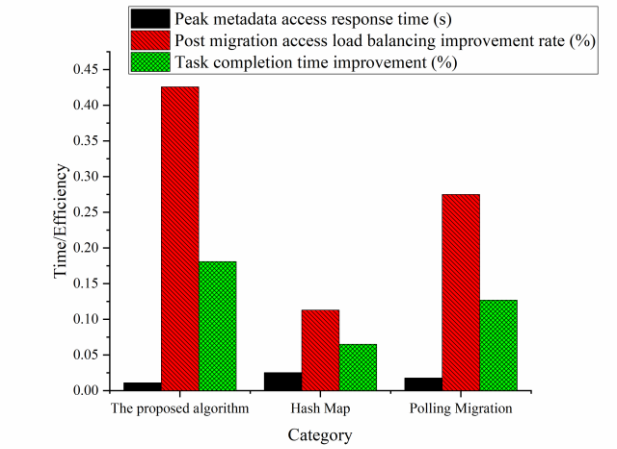


Figure 6: Comparison of diverse algorithms for periodic large-scale access task scheduling scenarios

In Figure 6, the proposed algorithm demonstrates improved performance in periodic large-scale access task scheduling scenarios. The algorithm's peak metadata access response time is only 0.0112 seconds, 55.9% and 37.1% lower than hash mapping and round-robin migration. Regarding load balancing performance, this algorithm increases the access balance rate after migration by 42.6%, which is 3.77 times and 1.55 times that of hash mapping and round-robin, respectively. At the same time, the task completion time is improved by 18.1%, and the improvement range exceeds that of the two baseline algorithms by 48.6% to 178%. Experimental data reveal that through the collaborative optimization of association rules and dynamic load balancing, this algorithm achieves breakthrough improvements in three dimensions: system response speed, resource allocation balance, and task processing efficiency. Thus, these can effectively meet the performance requirements in enterprise-level high-concurrency scenarios.

Table 3: Performance comparison of different algorithms in periodic large-scale access task scheduling scenarios

Method	Peak latency (s)	Load increase (%)	Completion time increase (%)	Variance σ^2	95% CI
The proposed algorithm	0.0112	42.6	18.1	0.000014	40.9–44.3
Hash Map	0.0254	11.3	6.5	0.00009	9.6–13.1
Polling Migration	0.0178	27.5	12.7	0.00004	25.3–29.6

In Table 3, the proposed algorithm achieves 0.0112s, 42.6%, and 18.1% in peak access latency, load balance improvement rate, and task completion time improvement rate, respectively, all of which are better than those of Hash Map and Polling Migration. The variance and 95% CI indicate that the proposed algorithm has higher stability and reliability in dynamic task scheduling scenarios.

5 Discussion

The proposed metadata migration management algorithm integrates data correlation with cluster load balancing principles. It outperforms traditional Hash Map and Polling Migration algorithms in scenarios involving high-frequency synergetic access and periodic large-scale access. Experiments show that the algorithm achieves a hot metadata recognition accuracy of 93.5%, which is 49.1% and 33.2% higher than the latter two methods; the average metadata access dealy is 7.4 ms, with a reduction range of 32.1%–49.3%; the migration trigger response time is 0.15 s, shortened by 61.5%–75.8%; in periodic scenarios, the peak access latency is 0.0112 s, reduced by 37.1%–55.9%, and the load balance improvement rate is 42.6%, which is 1.55–3.77 times that of traditional methods. The advantages stem from the algorithm breaking through the limitations of traditional single-dimensional approaches. The proposed algorithm realizes the aggregation of strongly correlated metadata through a weighted correlation graph to avoid access fragmentation; it combines node comprehensive load calculation and dynamic variance evaluation to solve the problems of insufficient adaptation of static mapping and delayed response of periodic scanning; thus, it achieves the coordination of access optimization and load balancing. This algorithm provides an adaptive solution for distributed storage metadata management and is especially suitable for enterprise-level complex concurrent scenarios. Future work should optimize the weight adaptation of heterogeneous clusters and the

efficiency of large-scale metadata correlation subgraph partitioning to improve its universality and engineering value.

6 Conclusion

This study focuses on the issue of metadata management in distributed storage systems and proposes a migration management optimization algorithm based on data correlation and cluster load balancing strategies. In high-frequency synergetic access scenarios, the proposed algorithm can accurately identify strongly correlated metadata under the same task or directory. Concurrently, it reduces cross-node access delay and improves the efficiency of multi-file joint access in big data analysis platforms by aggregating storage. Its hot spot recognition accuracy (93.5%) and average delay (7.4ms) can support the real-time response requirements of high-concurrency services. In periodic large-scale access tasks, the algorithm effectively alleviates node overload through dynamic load sensing and associated data aggregation. The 42.6% improvement rate in load balancing and shorter task completion time make it suitable for periodically fluctuating scenarios such as machine learning training tasks and log batch processing, ensuring system stability during load peaks. In enterprise-level distributed storage systems, the proposed algorithm can optimize the dynamic distribution of metadata, reduce scheduling delays caused by data dispersion, and improve storage resource utilization. It is particularly suitable for complex environments with multi-user and multi-task concurrency, providing technical support for efficient operation and maintenance of infrastructure such as cloud storage and data centers.

The limitations of this study are mainly reflected in the fact that most experimental data are based on the actual operation and maintenance environment and simulated loads of a single enterprise. The number of cluster nodes, hardware configuration, and network topology are fixed in the experiment, so the universality of the results may be limited. Its adaptability in heterogeneous clusters, extreme load fluctuations, or multi-cloud environments has not been fully verified; the algorithm may need further adjustment in edge scenarios or under extreme real-time constraints. The dynamic adjustment mechanism of weight coefficients in data correlation modeling still needs to be improved; the mining of implicit correlations across tasks and directories is not in-depth enough, and the potential of "semantic correlation development" has not been fully exploited. To enhance the generalization ability and practicality of the study, future work can conduct multi-load mode tests under different hardware specifications and cluster topologies; deep learning is introduced to optimize the dynamic adaptation ability of correlation weights to improve the accuracy of complex semantic correlation perception; the algorithm's application in heterogeneous cloud environments is expanded to enhance its universality; real-time stream processing technology is combined to reduce the response latency of migration decisions, realizing more refined dynamic scheduling, thus improving overall performance and system stability.

References

[1] Zhu Y, Xia T, Zhu T, et al. RAPO: An Automated Performance Optimization Tool for Redis Clusters in Distributed Storage Metadata Management[J]. IEEE Access, 2025, (13): 58060- 58074. DOI:10.1109/access.2025.3556240

[2] Huang X, Gao Y, Zhou X, et al. An adaptive metadata management scheme based on deep reinforcement learning for large-scale distributed file systems[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 2840-2853. DOI:10.1109/tnet.2023.3266400

[3] Singh H J, Bawa S. Scalable metadata management techniques for ultra-large distributed storage systems--A systematic review[J]. ACM Computing Surveys (CSUR), 2018, 51(4): 1-37. DOI:10.1145/3212686

[4] Eickhoff T, Eiden A, Göbel J C, et al. A metadata repository for semantic product lifecycle management[J]. Procedia CIRP, 2020, 91: 249-254. DOI:10.1016/j.procir.2019.11.006

[5] Chen Y, Ke Q, Li H, et al. xMeta: SSD-HDD-hybrid Optimization for Metadata Maintenance of Cloud-scale Object Storage[J]. ACM Transactions on Architecture and Code Optimization, 2024, 21(2): 1-20. DOI:10.1145/3652606

[6] Qi Y. Optimisation of distributed storage technology for large-scale data based on Hadoop technology[J]. International Journal of Reasoning-based Intelligent Systems, 2025, 17(7): 11-20. DOI:10.1504/ijris.2025.10071387

[7] He Z, Fang W. Research data management in institutional repositories: an architectural approach using data lakehouses[J]. Digital Library Perspectives, 2025, 41(1): 145-178. DOI:10.1108/dlp-02-2024-0022

[8] Cheng Y, Zhou K, Wang J, et al. A Framework for Big Earth Observation Data Using Horizontal Scaling Strategy[J]. IEEE Geoscience and Remote Sensing Letters, 2022, 19: 1-5. DOI:10.1109/lgrs.2022.3192640

[9] Punithavathi R, Kowsigan M, Shanthakumari R, et al. Protecting Data Mobility in Cloud Networks Using Metadata Security[J]. Computer Systems Science & Engineering, 2022,42(7):105-120. DOI:10.32604/csse.2022.020486

[10] Furner J. Definitions of “metadata”: A brief survey of international standards[J]. Journal of the Association for Information Science and Technology, 2020, 71(6): E33-E42. DOI:10.1002/asi.24295

[11] Vijaykumar N, Olgun A, Kanellopoulos K, et al. MetaSys: A practical open-source metadata management system to implement and evaluate cross-layer optimizations[J]. ACM Transactions on Architecture and Code Optimization (TACO), 2022, 19(2): 1-29. DOI:10.1145/3505250

[12] Huang X, Gao Y, Zhou X, et al. An adaptive metadata management scheme based on deep reinforcement learning for large-scale distributed file systems[J]. IEEE/ACM Transactions on Networking, 2023, 31(6): 2840-2853. DOI:10.1109/tnet.2023.3266400

[13] Dai H, Wang Y, Kent K B, et al. The state of the art of metadata managements in large-scale distributed file systems—scalability, performance and availability[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 3850-3869. DOI:10.1109/tpds.2022.3170574

[14] Dou W, Liu B, Lin C, et al. Architecture of virtual edge data center with intelligent metadata service of a geo-distributed file system[J]. Journal of systems architecture, 2022, 128: 102545. DOI:10.1016/j.sysarc.2022.102545

[15] Gao Y, Gao X, Zhang R, et al. An end-to-end learning-based metadata management approach for distributed file systems[J]. IEEE Transactions on Computers, 2021, 71(5): 1021-1034. DOI:10.1145/3337821.3337924

[16] Singh H J, Bawa S. LaMeta: An efficient locality aware metadata management technique for an ultra-large distributed storage system[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(10): 8323-8335. DOI:10.1016/j.jksuci.2022.08.012

[17] Altahat M A, Daradkeh T, Agarwal A. Virtual machine scheduling and migration management across multi-cloud data centers: blockchain-based versus centralized frameworks[J]. Journal of Cloud Computing, 2025, 14(1): 1. DOI:10.1186/s13677-024-00724-7

[18] Song S, Park S, Kwon D. metaSafer: A Technique to detect heap metadata corruption in WebAssembly[J]. IEEE Access, 2023, 11: 124887-124898. DOI:10.1109/access.2023.3327817

[19] Peng Z, Feng D, Chen J, et al. RHPM: Using relative hotness to guide page migration for hybrid memory systems[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 42(8): 2514-2526. DOI:10.1109/tcad.2022.3231836

[20] Gao Y, Gao X, Yang X, et al. An efficient ring-based metadata management policy for large-scale distributed file systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 30(9): 1962-1974. DOI:10.1109/tpds.2019.2901883

[21] He X, Zhang R, Tian P, et al. WOPE: a write-optimized and parallel-efficient B+-tree for persistent memory[J]. Journal of Systems Architecture, 2024, 153: 103187. DOI:10.1016/j.sysarc.2024.103187

[22] Cha M H, Lee S M, Kim H Y, et al. Effective metadata management in exascale file system[J]. Journal of Supercomputing, 2019, 75(11): 7665–7689. DOI:10.1007/s11227-019-02974-8

[23] Reddy N R S, Adapa M. AI-Driven Data Integration: Transforming Enterprise Data Pipelines through Machine Learning[J]. Journal of Computer Science and Technology Studies, 2025, 7(12): 110-119. DOI:10.30574/wjaets.2025.15.1.0245

[24] Wang D, Liu L, Liu Y. Normalized storage model construction and query optimization of book multi-source heterogeneous massive data[J]. IEEE Access, 2023, 11: 96543-96553. DOI:10.1109/access.2023.3301134

[25] Kumar A, Sivakumar P. Cat-squirrel optimization algorithm for VM migration in a cloud computing platform[J]. International Journal on Semantic Web and Information Systems (IJSWIS), 2022, 18(1): 1-23. DOI:10.4018/ijswis.297142

[26] Rao J, Ao T, Dai K, et al. ARCE: towards code pointer integrity on embedded processors using architecture-assisted run-time metadata management[J]. IEEE Computer Architecture Letters, 2019, 18(2): 115-118. DOI:10.1109/lca.2019.2935445

[27] Castro A, Areias M, Rocha R. Performance Evaluation of Separate Chaining for Concurrent Hash Maps[J]. Mathematics, 2025,13(17):2820-2820. DOI:10.3390/math13172820

[28] Smith J. Metadata remediation through migration, post-migration or necessary clean-up: A roadmap for success[J]. Journal of digital media management, 2024(3):12. DOI:10.69554/idto8966

Enterprise Financial Fraud Detection using GA-BPNN, Random Forest, and SVM with Multi-Modal Features

Jie Zhang
School of Accounting, Guangzhou College of Commerce
Guangzhou, Guangdong 511363, China
E-mail: zjiezj@hotmail.com

Keywords: machine learning, fraud behavior, indicator system, neural network

Received: September 27, 2025

This paper selected 22 indicators related to financial fraud, taking into account both financial and non-financial indicators from a machine learning perspective, and screened them through the IV value and Spearman correlation coefficient. Then, three algorithms, random forest (RF), support vector machine (SVM), and genetic algorithm-back-propagation neural network (GA-BPNN), were introduced, and experiments were carried out with the data of 784 fraudulent enterprises and 784 non-fraudulent enterprises sourced from the China Stock Market & Accounting Research Database as samples. The results indicated that using both financial and non-financial indicators yielded better identification results compared to using only financial indicators: the recall rate of the RF algorithm increased from 0.5762 to 0.6295, and the area under the receiver operating characteristic curve (AUC) increased from 0.6331 to 0.6874. The GA-BPNN algorithm exhibited the best performance in recognizing financial fraud behaviors, achieving a recall rate of 0.8256 and an AUC of 0.8537, which were the highest among the different algorithms. The shareholding ratio of the largest shareholder was the most important factor in identifying financial fraud, followed by the inventory turnover rate, which requires particular attention. The results demonstrate the usability of the GA-BPNN algorithm and the established indicator system, which can be applied in actual financial scenarios.

Povzetek: Študija pokaže, da lahko strojno učenje z uporabo finančnih in nefinančnih kazalnikov učinkovito izboljša prepoznavanje finančnih prevar v podjetjih.

1 Introduction

In the evolution of the capital market, there have always been some enterprises that have committed financial fraud in pursuit of illegal high returns. The existence of enterprise financial fraud has a significant influence on the development of the social economy [1], and research on its identification can provide technical support for investors and regulatory authorities, thereby maintaining the order and efficiency of the capital market [2]. The earliest fraud identification relied on the empirical judgment and analysis of auditors. Subsequently, statistical models such as logistic regression were applied in this field, improving the efficiency of identification. Compared with statistical models, machine learning methods exhibit better performance in identifying large amounts of data and have been widely applied in detecting financial fraud [2]. However, a single financial recognition indicator is often selected without considering non-financial indicators. From the perspective of machine learning, this paper studied the identification of enterprise financial fraud behavior and compared three different algorithms, with the aim of establishing a simple and reliable model to support practical work. Suppose that combining financial indicators with non-financial indicators can achieve a higher prediction accuracy compared to using only financial indicators. Assume that among different machine learning algorithms, the genetic algorithm-back-propagation neural network (GA-BPNN) can obtain superior performance than the random forest (RF) and the support vector machine (SVM). Through the empirical tests of fraudulent and non-fraudulent enterprises in the China Stock Market & Accounting Research Database (CSMAR), it is expected to construct a simple and reliable model for identifying corporate financial fraud behaviors and provide risk warning indicators with higher practical value for regulatory agencies and auditors.

2 Related works

Some related works are reviewed (Table 1).

Table 1: A summary of related works.

Literature	Method	Performance
Alghofaili et al. [3]	A financial fraud detection model based on long short-term memory	The model achieved an accuracy of 99.95% in less than one minute.
Binsawad [4]	Voted perceptron model	It was superior to competitors, with a low error rate and a high recall rate.
Li [5]	XGBoost model	The accuracy of short-term detection

		was higher than that of long-term detection, and two years was the optimal detection cycle.
Tong [6]	A metagraph fraud detection graph neural network	The F1-macro, area under the receiver operating characteristic curve (AUC), and geometric mean improved by 4.46%, 2.67%, and 8.59% respectively for the YelpChi dataset.

3 Financial fraud and machine learning

3.1 Enterprise financial fraud

Enterprise financial fraud refers to the act of an enterprise deliberately distorting true financial information in order to gain improper benefits [7]. The causes of fraud are complex, and the fraud triangle theory analyzes it from three perspectives: pressure, motivation, and excuse.

(1) Pressure: When faced with performance pressure and debt pressure, enterprises may choose to manipulate profits and hide debts.

(2) Motivation: When there are flaws in internal controls, information asymmetry, or regulatory loopholes, fraud can be carried out.

(3) Excuse: In cases where fraud is widespread in the industry or otherwise, enterprises can find excuses for the occurrence of fraud.

The existence of fraud directly affects the order of the capital market. When it is exposed, it will affect investors' confidence in investment. In general, enterprise fraud poses a great threat to the sustainable growth of the entire capital market [8]. Therefore, investigating methods for detecting fraud has become essential. From the perspective of machine learning, machine learning algorithms can be utilized to select a series of indicators related to fraud behavior to achieve identification.

3.2 Data description

In the study of this paper, Shanghai and Shenzhen A-share listed companies that were punished for financial fraud for the first time between 2000 and 2022 were used as fraud samples. Due to the significant differences in accounting between the financial sector and the non-financial sector, the financial sector was excluded from the sample selection, resulting in a total of 784 fraud samples. Then, the paired samples were selected in accordance with Beasley's pairing principle [9]:

(1) select enterprises that are in the same industry as the cheating company;

(2) the selected enterprises are not classified as tax assessment (TS), transfer pricing (TP), or *special treatment (ST), and the data is complete;

(3) the assets of the selected enterprises are comparable to those of the cheating enterprises.

Seven hundred eighty-four non-fraud samples were obtained. All data was sourced from the China Stock Market & Accounting Research (CSMAR) database (http://data.csmar.com).

3.3 Feature engineering

In the selection of indicators for enterprise financial fraud behavior, referring to the causes of fraud, in accordance with principles such as objectivity and operability, and considering both financial and non-financial aspects, an indicator system was established (Table 2).

Table 2: Indicator system for identifying financial fraud.1

Indicator type		Indicator name
Financial indicator	Development capacity	Growth rate of total assets
		Total profit growth rate
	Profitability	Net profit margin on total assets
		Net profit margin on current assets
		Return on net assets
		Operating profit margin
	Solvency	Current ratio
		Quick ratio
		Cash ratio
		Interest coverage ratio
		Asset-liability ratio
	Operational capacity	Accounts receivable to revenue ratio
		Accounts receivable turnover ratio
		Inventory turnover ratio
		Current asset turnover ratio
		Total asset turnover ratio
	Risk level	Financial leverage
		Combined leverage
Non-financial indicator	Equity structure	Shareholding ratio of the largest shareholder
	Board structure	Proportion of independent directors
	Corporate governance	Supervisory board shareholding ratio
	Audit	Types of audit opinions

As to the indicators in Table 2, except for the last one, all the others were numerical variables. The audit opinion type was a categorical variable, classified by whether a standard audit opinion was issued, with yes = 1 and no = 0.

For the features in Table 2, the missing values were imputed using the k-nearest neighbor method. The missing values of the categorical variable were filled in using the multiple imputation method. Then, the selected indicators were further screened.

First, the information value (IV) was calculated [10], which measures a variable's ability to distinguish the target event. When the IV value is lower than 0.02, the ability is weak, and its contribution to recognition is limited. Therefore, indicators with IV values less than 0.02 were excluded.

Then, the Spearman correlation coefficient was calculated [11]. The features in Table 1 may exhibit multicollinearity, which can lead to a decline in the stability and interpretability of the model. This coefficient measures the correlation between two variables and is more robust to non-normal distributions and outliers. When the result is greater than 0.7, it indicates a high degree of redundancy between the two variables and one should be removed. Through screening using the Spearman correlation coefficient, the feature combination can be optimized by eliminating multicollinearity, thereby constructing a more reliable model for identifying financial fraud behavior.

The indicators retained based on these two steps are displayed in Table 3.

Table 3: Financial fraud identification indicators after screening.

Indicator type		**Indicator name**	**Indicator number**
Financial indicators	Development capacity	Growth rate of total assets	X1
		Growth rate of total profit	X2
	Profitability	Net profit margin on total assets	X3
		Net profit margin on current assets	X4
		Return on net assets	X5
	Debt-paying ability	Cash ratio	X6
		Interest coverage ratio	X7
		Asset-liability ratio	X8
	Operational capacity	Accounts receivable to revenue ratio	X9
		Accounts receivable turnover ratio	X10
		Inventory turnover rate	X11
		Current asset turnover rate	X12
		Total asset turnover ratio	X13
	Risk level	Financial leverage	X14
Non-financial metrics	Equity structure	Shareholding ratio of the largest shareholder	X15
	Board structure	Proportion of independent directors	X16
	Corporate Governance	Supervisory board shareholding ratio	X17
	Audit	Types of audit opinions	X18

3.4 Model training

This article introduced three machine learning algorithms that are currently in use and perform well.

(1) Random forest (RF)

RF is an ensemble learning approach that uses decision trees as base classifiers and aggregates voting results from all decision trees. It has been widely applied in healthcare and cybersecurity lighting scenarios [12]. Here are the steps:

① based on the Bagging algorithm, sampling with replacement was used. To be specific, randomly select n training samples, and obtain k training sets after k rounds of sampling.

② train k decision trees;

③ vote and summarize the results of all decision trees to output the final financial fraud identification result.

(2) Support vector machine (SVM)

SVM, a generalized linear classifier, has good applications in scenarios such as text recognition and fault classification [13]. The principle of SVM is to establish a hyperplane (w,b) to maximize the distinction between positive and negative samples. The distance from any point x to hyperplane (w,b) can be written as:

$$r=\frac{|w^Tx+b|}{\|w\|},$$

where w is the normal vector and b is the displacement term. Assuming hyperplane (w,b) can classify samples correctly, that is:

$$\begin{cases} w^T x_i + b \geq +1, y_i = +1 \\ w^T x_i + b \leq -1, y_i = -1 \end{cases}.$$

It is necessary to find w and b that can satisfy constraints to minimize interval $\frac{\|w\|}{2}$, that is:

$$\max_{w,b} \frac{\|w\|}{2},$$

$$s.t.\ y_i(w^T x_i + b) \geq 1,$$

Equivalent to:

$$\min_{w,b} \frac{1}{2}\|w\|^2,$$

$$s.t.\ y_i(w^T x_i + b) \geq 1.$$

(3) Back-propagation neural network (BPNN)

BPNN is one of the most commonly applied artificial neural network models [14]. With good autonomous training and learning capabilities, it is widely used in scenarios such as speech recognition and image processing. The advantage of BPNN lies in its learning ability, but it converges slowly [15]. It can be optimized in combination with a genetic algorithm (GA) to obtain the optimal weight threshold of BPNN. The steps of the GA-BPNN algorithm are as follows.

① Determine the topology of the BPNN and initialize the network.

② Determine the chromosome length and encode to obtain the initial population.

③ Take the reciprocal of the mean square error as the fitness function in the GA.

④ Perform genetic operations: use roulette for selection operations, and use real number crossover for crossover operations. For mutation operations, randomly select an individual and mutate it to obtain a new individual according to a certain probability.

⑤ Compute the fitness value and keep iterating until the termination condition is reached.

⑥ Obtain the weights and thresholds optimized by the GA and input them into the BPNN to complete the training.

4 Results and analysis

4.1 Experimental setup

All data processing and model construction were completed in the Python 3.8 environment. Before being input into the machine learning models, the data was standardized using Z-score normalization to transform it into a distribution with a mean of 0 and a standard deviation of 1. All models used the same input features and data partitioning method. The experiment employed five-fold cross-validation. RF and SVM determined their optimal parameters using the grid search method [16].

RF:
n_estimators: 500
Criterion: gini
max_depth: 10
min_samples_split: 5
min_samples_leaf: 2
random_state: 42
SVM:
C:1.0
Kernel: rbf
Gamma: scale
random_state: 42

In the GA-BPNN algorithm, the number of input nodes was 18 (18 indicators in Table 3), and there was one output node. The number of nodes in the hidden layer was determined using an empirical formula: $y = \sqrt{m+n} + a$, where $m = 1$, $n = 18$, and a is a constant between 1 and 10.

Then, each architecture was trained using the same hyperparameters. After the training was completed, the AUC value was calculated. The architecture that achieved the highest AUC value was selected as the final architecture. It was found that when the number of hidden layer nodes was 10, the AUC value was the highest. Therefore, the BPNN architecture was determined to be: 18-10-1. The optimal weight and threshold values of the BPNN were determined by the GA. In the GA, the population size was 50, the number of iterations was 100, and the probabilities of crossover and mutation were 0.8 and 0.1, respectively. During the training, the number of epochs was set to 300, the batch size was 32, the binary cross-entropy loss function and Adam optimizer were used, the learning rate was 0.001, and early stopping was enabled to prevent overfitting.

The recognition of enterprise financial fraud is a two-classification problem and therefore can be evaluated based on a confusion matrix (Table 4).

Table 4: Confusion matrix.2

	Identified as positive	**Identified as negative**
Actually positive	TP	FN
Actually negative	FP	TN

The specific evaluation indicators included:

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN},$$

$$Recall = \frac{TP}{TP+FN},$$

$$Precision = \frac{TP}{TP+FP},$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}.$$

AUC value: it denotes the area beneath the receiver operating characteristic curve, a graphical plot that uses the false positive rate (FPR) as the horizontal axis and the true positive rate (TPR) as the vertical axis [17]:

$$TPR = \frac{TP}{TP+FN},$$

$$FPR = \frac{FP}{FP+FN}.$$

However, in the identification of enterprise financial fraud, it is more essential to accurately identify enterprises with fraud, i.e., the recall rate is more important. AUC can comprehensively measure the discrimination ability of an algorithm. Therefore, this paper mainly used the recall rate and AUC value.

4.2 Analysis of results

Two experiments were designed:

(1) use only financial indicators;

(2) use both financial indicators and non-financial indicators.

The results of three machine learning algorithms for different features are displayed in Table 5.

Table 5: Recognition performance of different algorithms.3

		Recall rate	AUC
Financial indicators	RF	0.5762±0.021	0.6331±0.025
	SVM	0.6685±0.025	0.7346±0.026
	GA-BPNN	0.7742±0.027b	0.8009±0.027b
Financial indicators + non-financial indicators	RF	0.6295±0.023a	0.6874±0.026a
	SVM	0.7148±0.026a	0.7856±0.027a
	GA-BPNN	0.8256±0.028ab	0.8537±0.029ab

Note: a indicates $p < 0.05$ compared to using financial features only; b indicates $p < 0.05$ compared to RF and SVM.

According to Table 5, in terms of indicator selection, the recognition performance of several algorithms improved after adding non-financial indicators. For example, the recall rate of the RF algorithm rose from 0.5762 ± 0.021 to 0.6295 ± 0.023, and the AUC rose from 0.6331 ± 0.025 to 0.6874 ± 0.026. Statistical tests suggested $p < 0.05$. Similar improvements were observed in SVM and GA-BPNN. The results indicate that non-financial indicators can provide a lot of information related to financial fraud and are useful for identifying financial fraud in enterprises.

Then, when comparing different machine learning algorithms, the GA-BPNN algorithm outperformed the RF and SVM algorithms in both experiments ($p < 0.05$). When using both financial and non-financial indicators simultaneously, the recall rate of the GA-BPNN algorithm was 0.8256 ± 0.028, which was 31.17% higher than that of the RF algorithm and 15.5% higher than that of the SVM algorithm. The AUC value was 0.8537 ± 0.029, showing an increase of 24.19% over the RF algorithm and 8.67% over the SVM algorithm. The results demonstrated the superiority of the GA-BPNN algorithm in identifying enterprise financial fraud. This may be because there is a very complex and highly non-linear relationship between financial and non-financial indicators. In terms of automatically capturing and modeling this complex nonlinear interaction effect, BPNN has more advantages.

According to the feature importance of each variable obtained by the RF algorithm, the top five features are shown in Table 6.

Table 6: Analysis of feature importance.4

	Feature importance
Shareholding ratio of the largest shareholder	0.081
Inventory turnover rate	0.067
Asset-liability ratio	0.056
Growth rate of total assets	0.051
Financial leverage	0.043

It was seen from Table 6 that the shareholding ratio of the largest shareholder was the most important factor in identifying financial fraud in enterprises. In reality, the fraud of Kangmei Pharmaceutical is associated with the imbalance of the equity structure, and the major shareholder may manipulate financial data through acts such as fictitious transactions. The importance of the inventory turnover rate was secondary. For instance, Zhangzi Island's fraud used the means of falsely increasing inventory. Abnormal fluctuations in the asset-liability ratio suggest that there might be problems with the business. For example, Enron Corporation concealed its debts, which led to an undervaluation of the asset-liability ratio. The abnormal surge in the growth rate of total assets may be related to inflated assets. For example, Kangdexin Composite Material Group inflated profits of 11.9 billion yuan. In the fraud case of Yongcheng Coal and Electricity Holding Group Co., Ltd., its financial leverage was very high. Based on these results, these indicators can be focused on in practice.

5 Discussion

This paper constructed a financial fraud behavior identification indicator system that combined financial and non-financial indicators, and compared the performance of three machine learning models, namely RF, SVM, and GA-BPNN, in the identification of financial fraud behaviors. Experiments demonstrated that the GA-BPNN algorithm achieved the best performance in identifying financial fraud behaviors (recall rate: 0.8256, AUC value: 0.8537). From this result, it can be seen that the financial fraud behavior of enterprises is not a direct result of the abnormality of a single financial ratio but a complex phenomenon generated through the interaction of various factors. There is a highly complex and non-linear relationship between financial and non-financial indicators. Compared with RF and SVM, BPNN has a greater advantage in automatically capturing and modeling this complex interaction effect. Moreover, the introduction of GA effectively overcomes the inherent defects of traditional BPNN. It not only enhances the identification performance of the model but also improves the stability and robustness of training. Therefore, the GA-BPNN algorithm demonstrated the best performance in identifying financial fraud behaviors.

In current research on identifying financial fraud behaviors, most studies rely on logistic regression or use only a single financial indicator, resulting in poor identification effects. Moreover, the results of the feature importance analysis of RF showed that the "shareholding ratio of the largest shareholder" was the most important indicator for identifying financial fraud behaviors, effectively proving the importance of non-financial indicators. Moreover, the "inventory turnover rate", as the second most important identification indicator, showed that inflating profits by manipulating inventory is a common financial fraud behavior at present. This result strongly supports the hypothesis of this paper, i.e., combining financial and non-financial indicators can effectively strengthen the predictive power of the model.

Based on the research results, the GA-BPNN algorithm can be used as a tool for auxiliary auditing and regulatory screening, providing references for auditors, regulatory agencies, etc. However, in actual deployment, the limitations of the GA-BPNN algorithm need to be further considered. The performance of this model was verified on a specific Chinese A-share dataset. Its universality across countries and markets remains to be verified. Moreover, the input of the model depended on historical data, and the feature selection method was relatively simple. In the future, it is necessary to verify the results on a wider range of samples, explore more feature selection methods, and compare it with more deep learning and statistical methods.

The GA-BPNN algorithm combines the GA and neural network. Its training complexity is higher than that of RF and SVM. However, after the training is completed, BPNN has a relatively fast prediction speed, which can meet the needs of most real-time detections. In addition, as a black-box model, it is difficult to directly explain the decision-making logic of the GA-BPNN algorithm to auditors, regulatory agencies, or corporate management.

Moreover, with the popularization of such models, there is a risk that companies deliberately avoid the red lines of the model, i.e., the gaming indicator. The limitations of the GA-BPNN algorithm require the model to evolve from a static predictor to a dynamic adaptive system to ensure the robustness of the nonlinear multivariable system [18]. To overcome this defect, in future work, for the high-risk cases identified by the GA-BPNN algorithm, interpretable artificial intelligence technologies such as Shapley additive explanations can be employed to explain the decisions of the GA-BPNN algorithm, generating audit trails and reports for business and regulatory purposes. Additionally, exploring the construction of an online learning or continuous learning framework can enhance the model's dynamic adaptation ability.

6 Conclusion

This paper analyzed the identification of enterprise financial fraud from the perspective of machine learning and compared the identification performance of three machine learning algorithms. Experiments demonstrated that using financial and non-financial indicators can achieve better identification results compared to using only financial indicators. Among the three algorithms, the GA-BPNN algorithm demonstrated the best performance with a recall rate of 0.8256 ± 0.028 and an AUC value of 0.8537 ± 0.029. This algorithm can be applied to the identification of actual financial fraud in enterprises.

References

[1] Zheng Z, Zhang R, Li Y, Huang X, Liang J (2024). A hybrid neural network-based fast financial fraud detection model. *Journal of Circuits, Systems & Computers*, 33(15), pp. 1–23. https://doi.org/10.1142/S0218126624502773.

[2] Kasgari A B, Vanani I R, Amiri M, Homayoun S (2025). Detecting Financial Fraud in Public Companies Using Financial and Non-Financial Metrics with a Machine Learning Approach. *Business Intelligence Management Studies*, 13(50), pp. 99–142. https://doi.org/10.22054/ims.2024.78018.2434.

[3] Alghofaili Y, Albattah A, Rassam M A (2020). A financial fraud detection model based on LSTM deep learning technique. *Journal of Applied Security Research*, 15(4), pp. 498–516. https://doi.org/10.1080/19361610.2020.1815491.

[4] Binsawad M (2025). Enhanced financial fraud detection using an adaptive voted perceptron model with optimized learning and error reduction. *ELECTRONICS*, 14(9), pp. 1875. https://doi.org/10.3390/electronics14091875.

[5] Li J (2025). Corporate governance, fraud learning cycles, and financial fraud detection: Evidence from Chinese listed firms. *Research in International Business and Finance*, 76(c), pp. 102832. https://doi.org/10.1016/j.ribaf.2025.102832.

[6] Tong G, Qian J, Shen J (2025). Adaptive metagraph neural network assisted by metagraph search for financial fraud detection. *Engineering Applications of Artificial Intelligence*, 153. https://doi.org/10.1016/j.engappai.2025.110807.

[7] Li W, Liu X, Zhou S (2024). Deep learning model based research on anomaly detection and financial fraud identification in corporate financial reporting statements. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 123, pp. 343–355. https://doi.org/10.61091/jcmcc123-24.

[8] Ma J (2023). Analysis of financial fraud supervision of listed companies based on game theory--take Luckin Coffee as an example. *SHS Web of Conferences*, 154, pp. 1–5. https://doi.org/10.1051/shsconf/202315402015.

[9] Beasley M S (1996). An empirical analysis of the relation between the board of director composition and financial statement fraud. *The Accounting Review*, 71(4), pp. 443–465. https://doi.org/10.1016/0361-3682(95)00051-8.

[10] Farooq S, Akram M S (2021). Comparison of data-driven landslide susceptibility assessment using weight of evidence, information value, frequency ratio and certainly factor methods. *Acta*

Geodynamica et Geomaterialia, 18(3), pp. 301–317. https://doi.org/10.13168/AGG.2021.0021.

[11] Song H Y, Park S (2020). An analysis of correlation between personality and visiting place using spearman's rank correlation coefficient. *KSII Transactions on Internet & Information Systems*, 14(5), pp. 1951. https://doi.org/10.3837/tiis.2020.05.005.

[12] Hadi A A A, Hadi A M (2024). Improving cybersecurity with random forest algorithm-based big data intrusion detection system: A performance analysis. *AIP Conference Proceedings*, 3051(1), pp. 1–11. https://doi.org/10.1063/5.0191707.

[13] Hamad R, Abushaala A M (2023). Medical named entity recognition in Arabic text using SVM. *2023 IEEE 3rd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA)*, 2023, pp. 200–205. https://doi.org/10.1109/MI-STA57575.2023.10169454.

[14] Wu G, Huang S, Liu T, Yang Z, Wu Y, Wei G, Yu P, Zhang Q, Feng J, Zeng B (2024). Numerical study of the biomechanical behavior of a 3D printed polymer esophageal stent in the esophagus by BP neural network algorithm. *Computer Modeling in Engineering & Sciences*, 138(3), pp. 2709–2725. https://doi.org/10.32604/cmes.2023.031399.

[15] Cheng H, Zhou Q (2022). Research on stock index forecast based on improved BP neural network model. *Proceedings of SPIE*, 12285, pp. 1–6. https://doi.org/10.1117/12.2637095.

[16] Pramada S K, Sajith O A, Sasidharan S, Thampi S G (2020). Grid search based SVM approach for sea level rise. *IOP Conference Series Earth and Environmental Science*, 581(1), pp. 012032. https://doi.org/10.1088/1755-1315/581/1/012032.

[17] Riikka N, Ileana M P, Ivan J, Tapio P, Antti A (2023). Quicksort leave-pair-out cross-validation for ROC curve analysis. *Computational Statistics*, 38(3), pp. 1579–1595. https://doi.org/10.1007/s00180-022-01288-3.

[18] Zouari F, Saad K B, Benrejeb M (2012). Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. *International Review on Modelling & Simulations*, 5(5), pp. 2075–2103.

Semantic Completeness-Enhanced Transformer Architecture for Named Entity and Relation Extraction in Noun Predicate Sentences

Bing Yan
School of Foreign Language, Hubei University of Education, Wuhan, 430205, China
E-mail: yanbingHUE@126.com

Keywords: semantic association, grammar pattern parsing, entity recognition, transformer, semantic completeness

Received: September 9, 2025

To solve the problem of semantic association and syntactic pattern parsing in noun predicate sentences, this paper proposed an entity recognition and relationship extraction method based on Transformer architecture. This method innovatively integrated three core components: enhancing text understanding ability through Similarity Whole Word Masking (Sim WWM) pre training, introducing semantic integrity quantification of text semantic coverage defects, and combining multi head semantic integrity mapping networks to achieve multi-dimensional complex feature learning. The experiment was based on the CoNLL04 dataset containing 1437 sentences and the SciERC dataset with 500 scientific abstracts. MTL-NER, SPKTE, DRPFF, and other baseline models were used. In entity recognition tasks, the proposed method achieved an average accuracy, recall, and F1 score of 90.7%, 91.6%, and 90.6% on the CoNLL04 dataset, and 69.8%, 70.4%, and 69.8% on the SciERC dataset, respectively; In the relationship extraction task, the three indicators on the CoNLL04 dataset were 75.3%, 73.4%, and 73.2%, respectively, and on the SciERC dataset were 62.9%, 60.5%, and 61.4%, respectively, all significantly better than the baseline model. The ablation experiment verification showed that the introduction of semantic integrity improved the model recognition accuracy by 3.9%, which was the core performance gain source. This method can effectively handle the problem of confusion between nested entities and entity types, providing an efficient technical path for complex semantic relationship parsing.

Povzetek: Članek predstavlja izboljšano metodo na osnovi Transformerja za prepoznavanje entitet in odnosov, ki dosega boljše rezultate od obstoječih pristopov.

1 Introduction

With the swift advancement of natural language processing (NLP) technology, semantic parsing of text data has become an important research direction. Among various text types, noun predicate sentences have attracted much attention due to their central position in information expression. Noun predicate sentences usually contain rich semantic information and complex grammatical structures, which can accurately describe the relationships between entities and the logic of events [1-2]. However, due to the complexity of its structure and the diversity of its semantics, automatically parsing the semantic associations and grammatical patterns of noun predicate sentences remains a challenging problem. In practical applications, accurate semantic association and syntactic pattern parsing are crucial for tasks such as information retrieval, knowledge graph (KG) construction, and machine translation. For example, in the field of medicine, accurately parsing noun predicate sentences in medical record texts can provide important support for disease diagnosis and treatment plan formulation. In the field of law, analyzing semantic relationships in legal texts helps to understand and apply legal provisions [3-4]. The existing semantic association and automatic parsing of grammar patterns mainly include two major tasks: named entity recognition and relation extraction. Entity recognition provides structured semantic information for text, which is helpful for further semantic analysis and understanding. Meanwhile, the recognized entities can serve as anchor points for syntax analysis, helping to parse sentence structures. Sentence relation extraction can reveal the semantic connections between entities, thereby enhancing the understanding of the overall semantics of the text. However, traditional parsing approaches often have limitations in dealing with complex semantic and syntactic structures, especially when facing problems such as nested entities and entity type confusion. The accuracy and robustness of traditional approaches are difficult to meet practical needs.

Extracting statement entity relationships (ERs) helps improve machines' semantic parsing and understanding capabilities for text. By identifying the relationships between entities, machines can better understand the deep meaning of text. Chen H et al. Introduced an approach based on heterogeneous graph attention network for extracting dialogue relationships. This approach simulates cross sentence relationships in conversations and utilizes heterogeneous graph attention networks to simulate various types of features in conversations. Through

experiment, it is evident that the F1 score and F1_c score of this approach were superior to the state-of-the-art approach [5]. Suo et al. introduced a novel fragment annotation approach for complex problems in entity extraction tasks. Based on conventional fragment annotation, this approach combines pointer annotation approach to design an annotation approach of "traversal enumeration+group mapping", and based on this, a entity extraction model based on span extraction is designed. The F1 score of the model was significantly improved compared to the baseline model [6]. Han and Jia introduced an improved ER extraction approach based on the Bidirectional Encoder Representation from Transformers (BERT) algorithm to address the issue of insufficient coverage of large-scale manufacturing knowledge data in existing joint extraction approaches for knowledge data ERs. This approach treats the extraction of quantitative knowledge as the extraction of manufacturing entity attributes by constructing a KG, thereby achieving the joint extraction of knowledge reasoning and data ERs. The findings revealed that the KG generated by this approach had good expressive ability and strong logical reasoning ability, and the required running time was only 12 seconds [7]. Wang et al. introduced a comprehensive prediction network approach that includes multi-feature semantic fusion to address the issues of breadth expansion and information redundancy in ER extraction. This approach integrates entity mask embedding sequence and context embedding sequence to enhance interaction between triplets and expand semantic encoding. The findings demonstrated that the model outperformed the baseline model in regard to accuracy and F1 score on public datasets [8].

The advancement of deep learning technology has opened up fresh avenues for NLP. Transformer architecture and its variants perform well in processing natural language tasks, effectively capturing long-distance dependencies and contextual information in text. Khan A et al. introduced a hybrid visual transformer to address the limitations of visual transformers in terms of generalization ability. This approach combines convolution operations and self attention mechanisms to utilize local and global image representations. The hybrid vision Transformer achieved significant results in a range of computer vision tasks [9]. Feng K et al. developed a Transformer-based editing approach for shape aware objects in high-resolution images. This approach utilizes a diffusion Transformer to capture long-range dependencies between patches and uses a diffusion probability model solver inversion algorithm to reduce the number of steps. This approach surpassed existing frameworks in generating higher quality edited images [10]. Yeh et al. developed a new visualization approach for understanding the internal operation of self attention mechanism in Transformer models. This approach analyzes global patterns in multiple input sequences through visual queries and joint embedding of keyword vectors. This approach could improve model understanding and provide new insights into query key interactions [11]. He et al. introduced a segmentation approach based on an efficient hierarchical hybrid visual transformer to address the weak modeling of long-range dependencies in convolutional neural networks (CNNs) for medical image segmentation. This approach integrates the advantages of CNNs, multi-scale channel attention, and transformers, establishing long-range dependencies between pixels through self attention. This approach outperformed existing approaches in learning image segmentation tasks, and significantly reduced inference time [12]. The literature review summary is shown in Table 1.

Table 1: The literature review summary

Literature	Method	Key performance indicators	Limitation
[5]	Heterogeneous Graph Attention Network	The F1 score and F1_c score were better than the state-of-the-art methods at that time	Only applicable to dialogue scenarios, unknown generalization ability
[6]	Traditional fragment annotation+pointer annotation+"traversal enumeration+group mapping" annotation strategy+span extraction model	F1 score significantly improved compared to baseline model	Focusing on a single entity extraction task, without involving relationship extraction; Dependency on custom annotation strategy
[7]	Improved BERT algorithm+KG construction	Good KG expression ability and logical reasoning ability; Running time is only 12 seconds	Limited to the manufacturing industry, with poor universality
[8]	Multi feature semantic fusion+entity mask embedding+context embedding	Accuracy and F1 score are better than the baseline model	Optimization of complex grammatical structures for noun predicate sentences
[9]	Hybrid Visual Transformer (Convolution+Self Attention Mechanism)	Achieve significant results in multiple visual tasks	Hybrid architecture increases model complexity and reduces computational efficiency
[10]	Diffusion Transformer+Diffusion	Generate and edit	Relying on the diffusion

	Probability Model Solver Inverse Algorithm	images with better quality than existing frameworks	probability model solver, although reducing the number of steps, there is still a problem of slow inference speed and high deployment cost
[11]	Visual query+keyword vector joint embedding	Enhance understanding of Transformer attention mechanism	Only a theoretical analysis tool, without proposing substantial task performance optimization solutions, insufficient adaptability to complex long texts
[12]	Efficient Hierarchical Hybrid Visual Transformer (integrating CNN, multi-scale channel attention Transformer）	The segmentation performance is superior to existing methods; Significant reduction in inference time	Multi module fusion leads to a large scale of model parameters and requires high computational power for deploying hardware

In summary, although the current model has markedly enhanced the efficacy of various NLP tasks, it still faces some challenges in handling semantic associations and grammatical patterns of noun predicate sentences. There are still three major difficulties in parsing noun predicate sentences: difficulty in segmenting the boundaries of multi-layer nested entities (such as “Artificial Intelligence Laboratory of XX University School of Computer Science”); The same word (“contract”) drifts in different scenarios and lacks global semantic validation; The implicit relationship (“drug hypertension” treatment) and overlapping triplet (“doctor patient drug”) are frequently missed or misjudged due to the lack of semantic integrity modeling, directly reducing accuracy and cross domain generalization. The root cause of the failure of existing methods is that pre trained models only perform local associations and lack self checking of whether the semantic framework is complete, resulting in missed detection of nested attributes and implicit elements; Rules or classic multitasking systems rely on fixed templates/single features to cope with multiple nested layers, resulting in weak structural adaptability; Domain customization solutions rely on manual knowledge injection and can only maintain a single scenario, without feeling the cross domain rules of generic noun predicate sentences, and still face difficulties in type confusion and transfer. The research zeroes in on three key challenges: (1) Accurately Segmenting Nested Entity Levels: It explores how to achieve this by leveraging semantic integrity mechanisms. (2) Eliminating Type Confusion: The study addresses how to overcome this issue through multidimensional semantic mapping. (3) Injecting Integrity Signals into Transformers: It investigates how to incorporate these signals to simultaneously enhance the accuracy of extracting implicit/overlapping relationships and improve cross-domain generalization capabilities. Therefore, to tackle the above issues, a Transformer architecture-based approach for automatically parsing semantic associations and grammatical patterns of noun predicate sentences is introduced. The novelty of the study consists in the introduction of semantic completeness and multi head semantic completeness mapping networks, which significantly enhance the model's ability to process complex text data. Secondly, Similarity Whole Word Masking (Sim-WWM) is introduced to improve the model's ability to understand text. The research not only provides a new technological path for semantic parsing of noun predicate sentences, but also provides useful references for further development in the field of NLP.

2 Methods and materials

One of the important research directions in the field of NLP is the semantic association and grammatical pattern parsing of noun predicate sentences. Therefore, to achieve semantic association and grammatical pattern parsing of noun predicate sentences, the study introduces a sentence analysis approach based on Transformer and BERT. This approach first identifies the entities of the statement to accurately represent the associated information of the statement. Next, based on the recognition of the associated information mentioned above, relationships are extracted to achieve the parsing of semantic associations and grammatical patterns in sentences.

2.1 Semantic association subject recognition approach based on Transformer and semantic completeness

Before performing automatic parsing of semantic associations and grammatical patterns in noun predicate sentences, it is necessary to first identify the semantic association subject, that is, the entity, of the sentence in order to accurately express the association information of the sentence. Named entity recognition refers to identifying entities with specific meanings from text, which is one of the fundamental tasks of semantic association and syntactic pattern parsing. Identifying named entities in the text enables a clearer comprehension of its semantic content and facilitates the determination of the grammatical roles of each component within the sentence. In entity recognition, to achieve entity disambiguation and context association, it is necessary to calculate semantic similarity. The current semantic similarity is generally represented by cosine similarity. The calculation of semantic similarity is presented in

equation (1).

$$s = \cos(\theta) = \frac{A * B}{\| A \| \| B \|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \sqrt{\sum_{i=1}^{n} (B_i)^2}} \quad (1)$$

In equation (1), s represents semantic similarity. θ represents the vector angle. A and B represent different vectors. A_i and B_i represent the components of vectors A and B in the i th dimension. n represents the vector dimension. However, relying solely on semantic similarity is not enough to reflect the semantic differences between different sentences. For example, multiple vectors in two-dimensional space can form the same angle with the same vector. The generatrix of a cone in three-dimensional space is similar to the cosine of its axis, but at different positions [13-15]. Therefore, to address the aforementioned issues, the study introduces semantic completeness. In the d-dimensional text vector space, semantic integrity quantifies the semantic differences between the original text vector and the masked text vector through a mapping function, measuring the degree of coverage of the target semantic framework core elements by the text. The range of values is [0,1], with closer values to 1 indicating more complete semantics. The multi head semantic integrity mapping network has four independent heads that focus on entity attributes, event elements, logical relationships, and contextual dimensions; The output features of each head are horizontally concatenated and fused into a unified semantic integrity vector through linear mapping, achieving multi-dimensional feature complementarity. Unlike the multi head attention mechanism, semantic integrity focuses on "semantic coverage evaluation", quantifying the degree of complete expression of specific semantic concepts in the current text fragment by comparing the representation differences between the original text and the masked text. Continuing with the above example, semantic integrity will automatically detect the coverage of the semantic framework of "doctor (subject) - basis (CT report) - diagnosis (action) - lung cancer (object)". If the sentence is missing the "CT report", it will significantly reduce the semantic integrity score, thereby assisting the model in identifying nested or hidden entities. The expression for semantic completeness is presented in equation (2).

$$S_c = M\left(t^m, t\right) (2)$$

In equation (2), S_c represents semantic completeness. M represents the mapping function. t^m is the masked text representation vector. t is the original text representation vector. After constructing the above concepts, statement entity recognition can be carried out. The selected entity recognition approach for research is based on fragment arrangement, which utilizes Transformer architecture for context representation and achieves named entity recognition by identifying and classifying fragments in text. The entity recognition model based on fragment arrangement is presented in Figure 1.

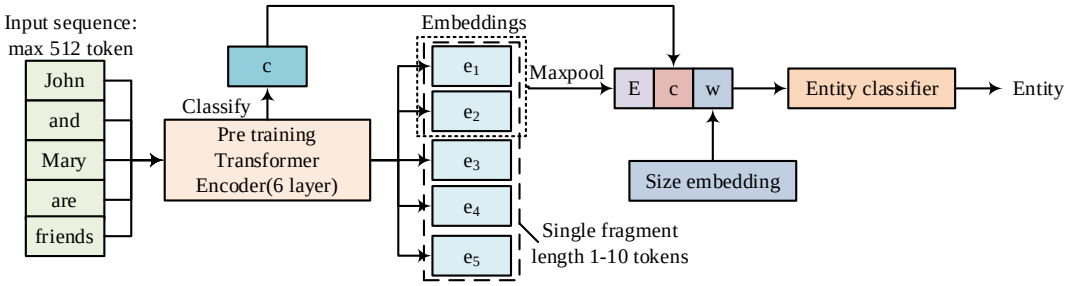


Figure 1: Entity recognition model based on fragment arrangement(Source: Author's own drawing)

As shown in Figure 1, the model first extracts all possible fragments from the text and generates a vector representation for each fragment. Next, it determines whether each fragment is a named entity. At this point, if multiple segments overlap and are recognized as entities, it is necessary to merge the relevant segments [16-17]. When using the above approach for entity recognition, considering the richness of feature representation, the study introduces span embedding. The vector calculation formula for entity classification after introducing span embedding is presented in equation (3).

$$x^e = e(s) \circ w (3)$$

In equation (3), x^e represents the vector used for entity classification. $e(s)$ represents a vector. w represents word embeddings. Although the above approaches can achieve recognition of sentence entities, the noise problem in fragment classification is relatively serious. Therefore, to address this issue, the study introduces the aforementioned semantic completeness into entity recognition models. The improved entity recognition model is presented in Figure 2.

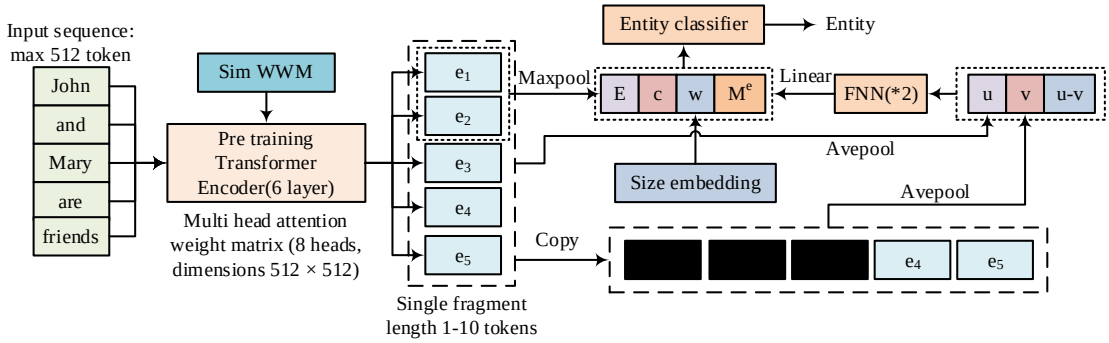


Figure 2: Improved entity recognition model(Source: Author's own drawing)

As shown in Figure 2, compared to the original entity recognition model based on fragment arrangement, the improved entity recognition model introduces Sim-WWM and semantic completeness, where Sim-WWM is used for pre training the Transformer. Meanwhile, by copying the word embedding sequence and masking the current subsequence, a masking sequence is obtained. Next, the original text sequence and the masked sequence are averaged to fuse the original information and partially masked information. Next, fusion vectors are used to calculate text similarity and evaluate the similarity of text content. Then, the fused vector is transformed through linear mapping and fused with other vectors. Finally, semantic completeness can be obtained through nonlinear mapping using a Feedforward Neural Network (FNN). The standard masking mechanism (such as BERT's MLM and WWM) has two major limitations: First, the masking strategy is randomized and does not consider the semantic relevance of the text, which may damage the entity integrity of noun predicate sentences (such as random masking of entity core words leading to semantic fragmentation); Secondly, it is difficult to capture the global semantic framework of "entity attribute relationship" in noun predicate sentences by only learning local semantics through a single masking method. Unlike the standard masking mechanism, Sim WM uses a "semantic similarity-guided masking strategy" to construct more targeted counterfactual samples while preserving the core semantic structure of the sentence, which helps the model learn semantic integrity. Specifically, the semantic core of noun predicate sentences relies on entities and their associated relationships. Sim WWM prioritizes masking non-core semantic words while preserving entities and key relationship words, creating semantic differences to train the model's integrity awareness ability. This approach avoids semantic framework collapse caused by masking core entities, thereby more accurately assisting in semantic integrity assessment. Moreover, Sim WM is not only applicable to noun predicate sentences, its core design logic is a "masking strategy guided by semantic similarity", which has cross sentence and cross domain universality. The calculation formula for the fused text representation vector is presented in equation (4).

$$r = t \circ t^m \circ |t - t^m| \quad (4)$$

In equation (4), r represents the fused text representation vector. At this point, the entity classification calculation formula is presented in equation (5).

$$\begin{cases} y^e = softmax\left(W^e * (t \circ c \circ w \circ M^e) + b^e\right) \\ M^e = Linear(o) \end{cases} \quad (5)$$

In equation (5), y^e represents the entity classification result. W^e represents the weight matrix. c represents the overall representation vector of the text sequence. M^e represents the text representation vector after linear mapping. b^e represents the boundary label. To use the linearly mapped text representation vector to represent semantic completeness, it is necessary to ensure that the number of layers in the FNN is not zero. At this point, the formula for calculating the text representation vector after linear mapping is presented in equation (6).

$$M^e = Linear\left(f^L(o)\right) \quad (6)$$

In equation (6), f^L represents FNN. L represents the number of layers in FNN. Due to the possibility of information loss caused by a single feedforward network, a multi-head semantic completeness mapping network is introduced to address the aforementioned issues. The multi-head semantic completeness mapping network is presented in Figure 3.

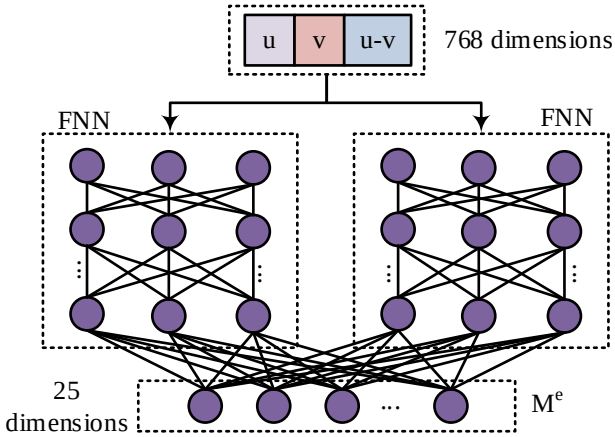


Figure 3: Multi-head semantic completeness mapping network (Source: Authors own drawing)

In Figure 3, the multi-head semantic completeness mapping network can utilize multiple FNNs to map input vectors to different feature spaces, each of which focuses

on capturing different aspects or levels of semantic information in the input data. Due to the ability of this network to map vectors into representation vectors of different levels, there are differences in the vector structure in each path. By using the above approach, entity recognition of noun predicate sentences can be achieved for the purpose of extracting sentence relationships in the future.

2.2 approach for extracting sentence relationships considering text dependencies

After identifying the entities of noun predicate sentences using the above approach, the sentence relationships can be extracted. RE holds a pivotal position in semantic association and automatic parsing of grammar patterns. It not only helps to better understand the semantic structure of text, but also provides important clues and basis for grammar analysis. The purpose of RE is to identify entity pairs with specific semantic relationships in text, but traditional RE approaches rely on entity positional relationships, resulting in lower accuracy in relationship recognition [18-19]. Therefore, to address the aforementioned issues, the study introduces semantic completeness into the RE model and designs a new sentence partitioning scheme. The sentence partitioning scheme is presented in Figure 4.

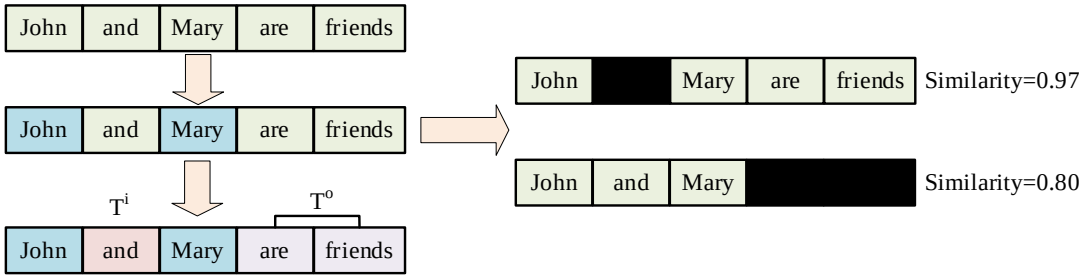


Figure 4: Statement division scheme (Source: Authors own drawing)

As shown in Figure 4, the sentence partitioning scheme designed in the study takes entity pairs as boundaries and divides the remaining words into words between entity pairs and other words. At this point, cover up non entity words and determine the importance of words in different positions to the sentence based on their similarity changes. Finally, the semantic completeness vector can be obtained through nonlinear mapping processing. After introducing the above concepts, the sentence RE model is presented in Figure 5.

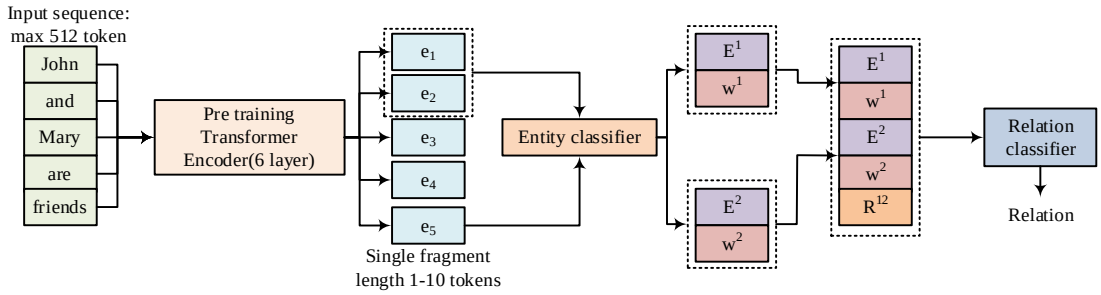


Figure 5: Sentence RE model (Source: Authors own drawing)

As shown in Figure 5, after receiving the text sequence, the model will generate embedding representations for each word through a pre trained Transformer encoder. Next, the entity classifier in the model is used to identify entities in the text, and all entities are used as nodes to construct a bidirectional fully connected graph [20-21]. Then, the model calculates the semantic completeness representation vector when covering different entities. Finally, a relationship classifier is employed to identify the relationships between entities. The calculation formula for a bidirectional fully connected graph is presented in equation (7).

$$U^n = \left\{ r^{12}, \cdots, r^{1n}; r^{2,1}, \cdots, r^{2n}; r^{n1}, \cdots, r^{n(n-1)} \right\} (7)$$

In equation (7), U^n represents a bidirectional fully connected graph of relationships. $r^{n(n-1)}$ represents the relationship between entity n and entity $n-1$. The formula for calculating the relationship representation vector of entities is presented in equation (8).

$$x^{ni} = E^n \circ w^n \circ E^i \circ w^j \circ R^{ni} (8)$$

In equation (8), x^{ni} represents the relationship vector between different entities. E^n and E^i represent word embedding vectors. w^n and w^i are span embeddings. R^{ni} represents the relationship dependency vector between different entities. Due to the division of the text in the study, the formula for calculating the relationship dependency vector can be rewritten as equation (9).

$$R = f\left(T^i, T^o\right) (9)$$

In equation (9), R represents the relationship dependency vector. T^i represents vocabulary between entities. T^o represents the vocabulary on both sides of the entity. Based on this, the details of the sentence RE model are presented in Figure 6.

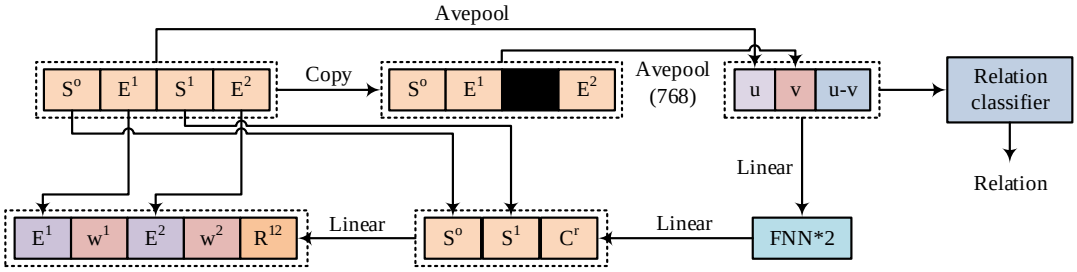


Figure 6: Details of the statement RE model(Source: Author's own drawing)

As shown in Figure 6, after inputting text, the model first converts the text into a sequence of word embeddings and divides the sentences according to the partitioning scheme. Next, the model processes the word embedding vectors through masking and replication mechanisms, and averages and pools the masked embedding sequence to obtain a comprehensive representation. Then, FNN is used to extract features from the average pooled vector. Next, the output of FNN is fed into the linear layer to generate the final relationship representation vector. Finally, a relationship classifier is used to identify the relationship representation vectors. The calculation formula for the pooled text representation vector is presented in equation (10).

$$u = avepooling(T_1^o \circ E^1 \circ T^i \circ E^2 \circ T_2^o) \quad (10)$$

In equation (10), u represents the average pooled text representation vector. The masked text representation vector is presented in equation (11).

$$v = avepooling(T_1^o \circ E^1 \circ T^{mask} \circ E^2 \circ T_2^o) \quad (11)$$

In equation (11), v represents the masked text representation vector. T^{mask} represents an all zero vector. The semantic completeness representation vector can be obtained by calculating the difference between the pooled and masked text representation vectors. The formula for calculating the semantic completeness representation vector is presented in equation (12).

$$S_c^r = Linear\left(f^L(u \circ v \circ |u - v|)\right) \quad (12)$$

In equation (12), S_c^r represents the semantic completeness representation vector. At this point, the formula for calculating the relationship dependency vector can be rewritten as equation (13).

$$R = f(T^i, T^o) = Linear(T^i, T^o, S_c^r) \quad (13)$$

The fusion of statement information and semantic completeness vector can be achieved through a fully connected layer, thereby enabling the semantic completeness representation vector to affect each statement element [22]. By using the above approaches, the model can learn more complex features, thereby improving the accuracy of sentence relation extraction. The calculation formula for the relationship classifier of the model is presented in equation (14).

$$r^{ni} = softmax\left(W^r * (E^n \circ w^n \circ E^i \circ w^i \circ R) + b^r\right) \quad (14)$$

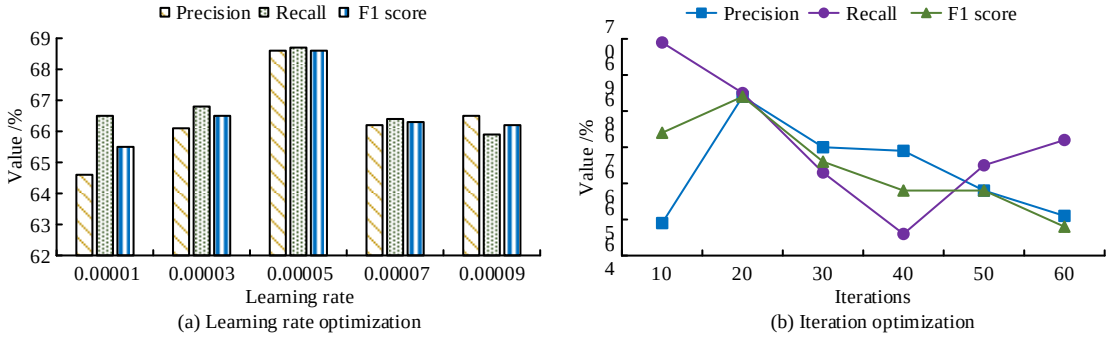
In equation (14), r^{ni} represents the classification result. W^r represents the weight matrix. b^r represents the boundary label of the relationship. Meanwhile, to further improve the accuracy of RE in the model, the study also introduces a negative sample mechanism, which constructs negative samples outside the labeled positive samples to help the model learn more features and patterns, and improve its generalization ability on unseen data. By using the above approaches, it is possible to extract sentence relationships, reveal semantic connections between different sentences in the text, and construct the discourse structure of the text to better understand the organization of the text, thereby achieving automatic parsing of semantic associations and grammatical patterns.

3 Results

3.1 Entity recognition test results

To validate the performance of the introduced entity recognition model, it was tested and compared with the Mixed Transfer Learning Named Entity Recognition (MTL-NER) model and the Entity Recognition model based on Semantic Prior Knowledge and Type Embedding (SPKTE). The CPU used in the experiment was Intel core i7 4770K, and the GPU was NVIDIA GeForce RTX 4050. The datasets used in the experiment were CoNLL04 dataset and SciERC dataset, where CoNLL04 dataset contained 1437 sentences and SciERC dataset contained 500 scientific abstracts. In the experiment, the learning rate and maximum gradient norm of the model were 0.00005 and 1.0.The span embedding size and maximum negative sample size were 25 and 100. The Batchsize, maximum fragment span, and semantic completeness embedding sizes were 2, 10, and 25. The model training cycle (Epoch) was 30 rounds, and an early stopping strategy (Early Stopping) was adopted to prevent overfitting. When the F1 score of the validation set did not improve for 5 consecutive rounds, the training was automatically terminated and the optimal model weights are saved. The experiment adopted a binary segmentation strategy of "development set test set", with the dataset divided in a 7:3 ratio and the development set split from the training set in a 10% ratio for hyperparameter tuning and early stop judgment. The study set fixed random seeds (Python random seed=42, PyTorch random seed=42,

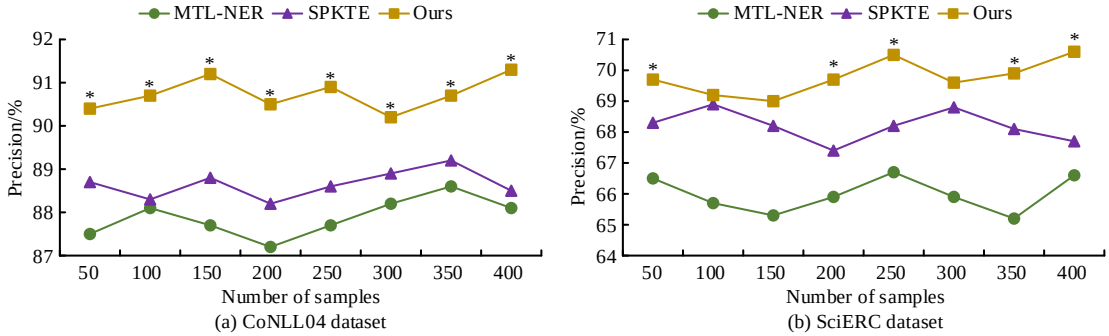
NumPy random seed=42) to ensure that the results of data partitioning, model initialization, masking strategy, and other steps are reproducible. The model used the AdamW optimizer with a weight decay set to 0.01, β_1=0.9, β_2 =0.999, ε =1e^{-8}; The learning rate scheduling strategy adopted linear preheating+cosine annealing, with 3 preheating rounds and an initial learning rate of 0.00005. The loss function adopted a cross entropy loss function with category balanced weights. The baseline model referred to the network structure and hyperparameter configuration of the original paper, and was autonomously re-implemented based on PyTorch to ensure consistency with the training environment (hardware, optimizer) of the model in this paper. The impact of learning rate and iteration times on model performance is presented in Figure 7.



(a) Learning rate optimization (b) Iteration optimization

Figure 7: Influence of learning rate and iteration times on model performance (Source: Authors own drawing)

According to Figure 7 (a), as the learning rate increased, the accuracy, recall, and F1 score all first increased and then decreased. When the learning rate was 0.00005, the performance of the model reached its optimal level. At this point, the accuracy, recall, and F1 score of the model were 68.6%, 68.7%, and 68.6%. As shown in Figure 7 (b), with the increase of iteration times, the accuracy and recall of the model first increased and then decreasde, while F1 score first decreased and then increased. After comprehensive consideration, the performance of the model was considered optimal when the number of iterations was 20. At this point, the accuracy, recall, and F1 score of the model were 68.4%, 68.5%, and 68.4%. The performance of the model was optimal when the learning rate and iteration times were 0.00005 and 20. The accuracy of entity recognition in different datasets is presented in Figure 8.

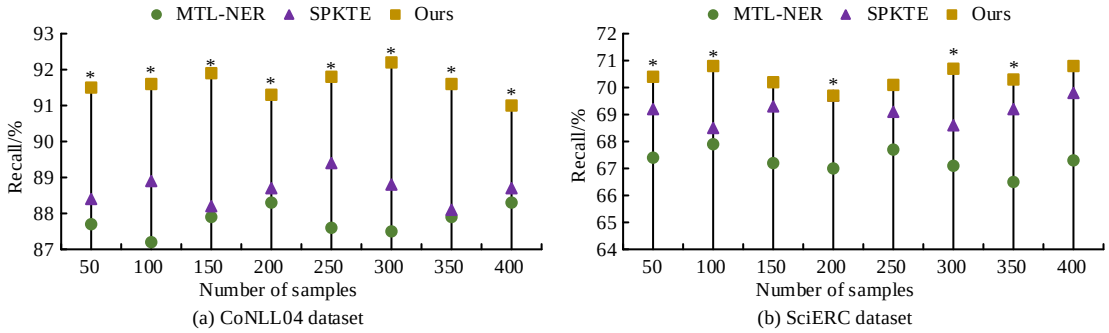


(a) CoNLL04 dataset (b) SciERC dataset

Note: * indicates p<0.05.

Figure 8: Entity recognition accuracy in different data sets (Source: Authors own drawing)

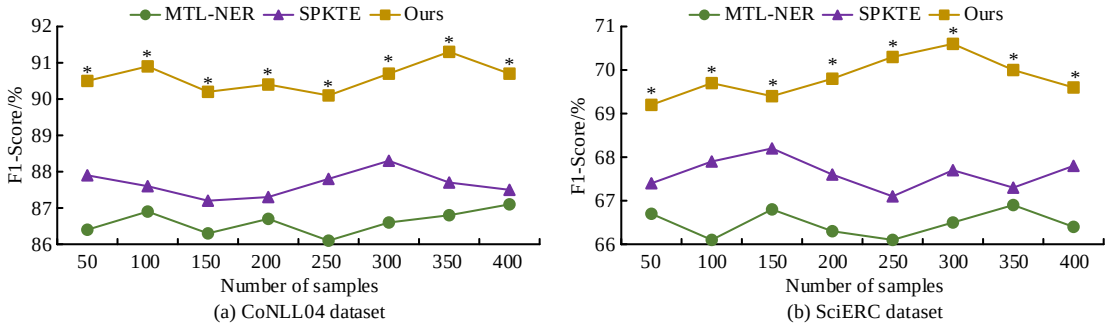
According to Figure 8 (a), in the CoNLL04 dataset, the recognition accuracies of MTL-NER and SPKTE were the highest at 88.6% and 89.2%, and the lowest at 87.2% and 88.2%. The average accuracies were 87.9% and 88.7%. The lowest accuracy of the entity recognition model introduced in the study was 90.2% (p<0.05), with an average accuracy of 90.7% (p<0.05), which was higher than other approaches. According to Figure 8 (b), in the SciERC dataset, the recognition accuracies of MTL-NER and SPKTE were the highest at 66.7% and 68.9%, and the lowest at 65.2% and 67.4%. The average accuracies were 66.0% and 68.2%. The lowest accuracy of the entity recognition model introduced in the study was 69.0% (p<0.05), with an average accuracy of 69.8% (p<0.05), which was higher than other approaches. The introduced entity recognition approach could achieve accurate recognition of named entities. The recall rates in different datasets are presented in Figure 9.



Note: * indicates $p<0.05$.

Figure 9: Recall rates in different data sets (Source: Authors own drawing)

According to Figure 9 (a), in the CoNLL04 dataset, the recall rates of MTL-NER and SPKTE were the highest at 88.3% and 89.4%, respectively, and the lowest at 87.2% and 88.1%. The average recall rates were 87.8% and 88.7%. The lowest recall rate of the entity recognition model introduced in the study was 91.0% ($p<0.05$), with an average recall rate of 91.6% ($p<0.05$), which was higher than other approaches. According to Figure 9 (b), in the SciERC dataset, the recall rates of MTL-NER and SPKTE were the highest at 67.9% and 69.8%, and the lowest at 66.5% and 68.5%. The average recall rates were 67.3% and 69.2%. The lowest recall rate of the entity recognition model introduced in the study was 69.7% ($p<0.05$), with an average recall rate of 70.4% ($p<0.05$), which was higher than other approaches. It can be concluded that the entity recognition model introduced in the study had good generalization ability. The F1 score in different datasets is presented in Figure 10.



Note: * indicates $p<0.05$.

Figure 10: F1 score in different data sets (Source: Authors own drawing)

As shown in Figure 10 (a), in the CoNLL04 dataset, the F1 scores of MTL-NER and SPKTE were the highest at 87.1% and 88.3%, and the lowest at 86.1% and 87.2%. The average F1 scores were 86.6% and 87.7%. The lowest F1 score of the entity recognition model introduced in the study was 90.1%, with an average F1 score of 90.6% ($p<0.05$), which was higher than other approaches. According to Figure 10 (b), in the SciERC dataset, the F1 scores of MTL-NER and SPKTE were the highest at 66.9% and 68.2%, and the lowest at 66.1% and 67.1%. The average F1 scores were 66.5% and 67.6%. The lowest F1 score of the entity recognition model introduced in the study was 69.2%, with an average F1 score of 69.8% ($p<0.05$), which was higher than other approaches. With the above outcomes, it can be concluded that the entity recognition model introduced in the study had good robustness. The ablation experiment results of the model are presented in Table 2.

Table 2: Results of ablation experiments

Model	Multiple mechanisms	FNN	Semantic completeness	Precision/%
1	×	×	×	82.3
2	√	×	×	84.4
3	×	√	×	83.8
4	×	×	√	86.2
5	√	√	×	87.6
6	√	×	√	88.9
7	×	√	√	89.3
8	√	√	√	90.7

According to Table 2, after introducing the multi-head mechanism, FNN, and semantic completeness, the recognition accuracy of the model significantly increased from 82.3% to 90.7%. Among them, semantic completeness had the most significant impact on model performance. After introducing semantic completeness, the recognition accuracy of the model increased by 3.9%. Comparing Model-1 (with no semantic integrity) and Model-4 (with only semantic integrity), the accuracy increased from 82.3% to 86.2%, an increase of 3.9 percentage points. This improvement was entirely driven by the integrity scoring component, as Model-4 did not introduce multi head mechanisms and FNN, and only achieved performance improvement through the quantification of differences between raw text and masked text (integrity scoring core logic), proving that this component was the core functional carrier of the semantic integrity framework. The accuracy comparison between Model-4 (semantic integrity only) and Model-7 (semantic integrity+FNN) showed that after adding FNN, the accuracy increased from 86.2% to 89.3%, an increase of 3.1 percentage points. Combining method design, FNN was the core implementation of feature aggregation (equation 6), which functions to perform nonlinear transformation on the original semantic difference features and enhance feature discrimination. This indicated that the feature aggregation component provided more effective input for integrity scoring by optimizing feature quality, indirectly improving the accuracy of semantic integrity evaluation. The introduced improvement approach could effectively improve the accuracy of statement entity recognition. To verify the novelty of the method proposed in this paper, pre trained models such as BERT base and RoBERTa base, which are widely used in entity recognition and relationship extraction tasks, were selected as baselines. The performance, core improvements, and applicable scenarios were compared, with a focus on analyzing the unique contribution of semantic integrity. The comparison results of pre trained language models are shown in Table 3.

Table 3: Comparison results of pre trained language models

Model	Task	CoNLL04	SciERC
BERT-base	Entity recognition	85.9%*	64.9%*
	Relationship extraction	69.5%*	56.0%*
RoBERTa-base	Entity recognition	87.8%*	67.1%*
	Relationship extraction	71.9%*	58.2%*
Ours	Entity recognition	90.6%*	69.8%*
	Relationship extraction	73.2%*	61.4%*

Note: * indicates $p<0.05$.

According to Table 3, in the entity recognition task, the F1 score of the proposed method on the CoNLL04 dataset was 2.8% higher than that on the RoBERTa base dataset, and 2.7% higher on the SciERC dataset; In the relationship extraction task, the F1 score of our method on the CoNLL04 dataset was 1.3% higher than that on the RoBERTa base dataset, and 3.2% higher on the SciERC dataset; As the complexity of the dataset increased (SciERC is a scientific abstract that includes more nested entities and specialized terminology), the performance advantage of the proposed method became more significant, verifying the adaptability of semantic integrity to complex semantic scenarios. To verify the robustness and generalization ability of the model in different professional fields, additional experiments were conducted using the BioRED dataset in the biomedical field and the CAIL2020 dataset in the legal field. The experimental results are shown in Table 4.

Table 4: Cross disciplinary experimental results

Model	Task	BioRED	CAIL2020
MTL-NER	Entity recognition	77.6%*	80.8%*
	Relationship extraction	69.3%*	72.8%*
SPKTE	Entity recognition	79.3%*	82.6%*
	Relationship extraction	71.9%*	75.1%*
Ours	Entity recognition	84.3%*	87.1%*
	Relationship extraction	77.3%*	80.0%*

Note: * indicates $p<0.05$.

According to Table 4, cross domain experiments (BioRED, CAIL2020, and other 4 datasets) showed that the entity recognition F1 score of the research method was 84.3%/87.1%, and the relationship extraction F1 score was 77.3%/80.0%, both leading the baseline by 4-8 percentage points; The coefficient of variation was $\leqslant$ 7.5%, significantly lower than the comparison model, verifying the high accuracy and strong generalization of the multi head semantic integrity network.

3.2 RE test results

To validate the performance of the RE model introduced in the study, it was tested and compared with the ER extraction model based on Dual Relationship Prediction and Feature Fusion (DRPFF) and the Aggregation Logic Network Based on Binary Graph Structure (BoBGSAL-

Net). The dataset, software and hardware settings, and hyperparameters used in the experiment were consistent with the above experiment. In the CoNLL04 and SciERC datasets, the filter thresholds were 0.4 and 0.3. The triple used for relationship extraction tasks in the experiment was defined as a structured data unit of "entity pair relationship", consisting of subject entities, object entities, and semantic relationships between subjects and objects. The validity determination of triplets required two conditions: 1) both the subject entity and the object entity are valid entities confirmed by the entity recognition model (entity types meet the definition of the dataset category); 2) ERs refer to the semantic relationships that truly exist between entities in the original text, generated through non random combinations. To ensure the representativeness and balance of experimental data, the triplet sampling adopted a "type stratification+proportional control" strategy. Positive samples were stratified by entity type combination, with 500-800 (training)/100-200 (testing) and a 1:1 balanced relationship category; Negative samples were generated by shuffling random entities of the same type, with a 3:1 ratio and class by class alignment. Pseudo examples with self matching and distance<2 tokens were removed to ensure balanced distribution and semantic rationality. The preprocessing method for triplets was as follows: boundary calibration merges cross word entities, fuzzy relationships were uniformly mapped, and unlabeled sentences were deleted; Long sentences with entity pairs as the center, truncated by $\pm$ 50 tokens to prevent fragmentation; Use (es, eo, r) to remove duplicates and eliminate type mismatch samples. Finally, the study left 8k-10k for training and 1.5k-2k for testing, with a distribution variation of $\geqslant$ 10% to maintain balance. The parameter settings are shown in Table 5.

Table 5: The parameter settings

Parameter	Value
Weight decay	0.01
Optimizer coefficient ($\beta_1/\beta_2/\varepsilon$)	0.9/0.999/1e-8
Optimizer coefficient	0.00005
Maximum gradient norm	1.0
Span embedding dimension	25
Semantic integrity embedding dimension	25
Batch size	2

According to Table 5, the weight decay of the model was 0.01, the initial learning rate was 0.00005, and both the span embedding dimension and semantic integrity embedding dimension were 25. The accuracy of RE from different datasets is presented in Figure 11.

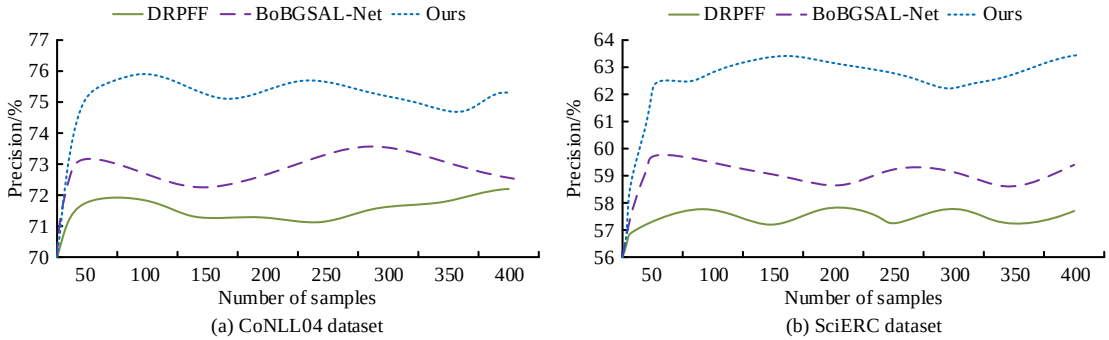
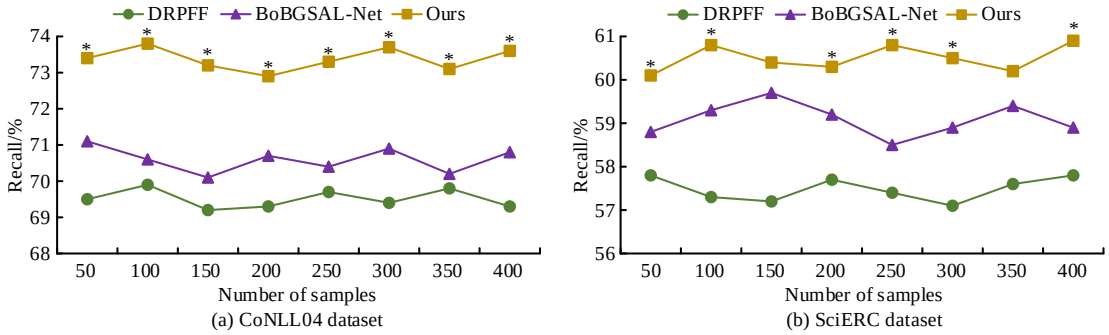


Figure 11: Precision of relation extraction in different data sets (Source: Authors own drawing)

According to Figure 11 (a), in the CoNLL04 dataset, the highest accuracy of DRPFF and BoBGSAL-Net was 72.2% and 73.5%, while the lowest was 71.1% and 72.2%. The average accuracy was 71.6% and 72.9%. The introduced RE model had a minimum accuracy of 75.1% and an average accuracy of 75.3%, which was higher than other approaches. According to Figure 11 (b), in the SciERC dataset, DRPFF and BoBGSAL-Net had the highest accuracies of 57.9% and 59.8%, and the lowest accuracies of 57.1% and 58.6%. The average accuracies were 57.5% and 59.2%. The introduced RE model had a minimum accuracy of 62.2% and an average accuracy of 62.9%, which was higher than other approaches. It is evident that the introduced RE model could accurately extract sentence relationships. The recall rates in different datasets are presented in Figure 12.



Note: * indicates $p<0.05$.

Figure 12: Recall rates in different data sets (Source: Authors own drawing)

According to Figure 12 (a), in the CoNLL04 dataset, the recall rates of DRPFF and BoBGSAL-Net were the highest at 69.9% and 71.1%, and the lowest at 69.2% and 70.1%. The average recall rates were 69.5% and 70.6%. The lowest recall rate of the RE model introduced in the study was 72.9%, with an average recall rate of 73.4% ($p<0.05$), which was higher than other approaches. According to Figure 12 (b), in the SciERC dataset, the recall rates of DRPFF and BoBGSAL-Net were the highest at 57.8% and 59.7%, and the lowest at 57.1% and 58.5%. The average recall rates were 57.5% and 59.1%. The lowest recall rate of the RE model introduced in the study was 60.1%, with an average recall rate of 60.5% ($p<0.05$), which was higher than other approaches. The introduced RE model had good generalization ability. The F1 score in different datasets is presented in Figure 13.

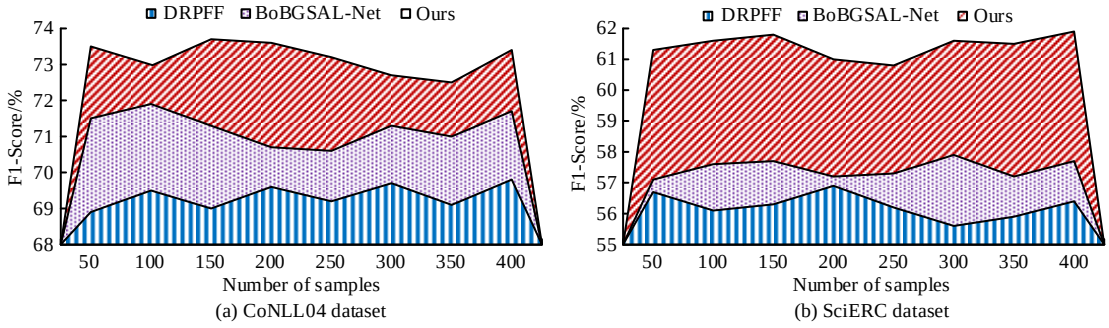


Figure 13: F1 score in different data sets (Source: Authors own drawing)

According to Figure 13 (a), in the CoNLL04 dataset, the F1 scores of DRPFF and BoBGSAL-Net were the highest at 69.8% and 71.9%, and the lowest at 68.9% and 70.7%. The average F1 scores were 69.4% and 71.3%. The RE model introduced in the study had a minimum F1 score of 72.5% and an average F1 score of 73.2%, which was higher than other approaches. According to Figure 13 (b), in the SciERC dataset, the F1 scores of DRPFF and BoBGSAL-Net were the highest at 56.9% and 57.9%, and the lowest at 55.6% and 57.1%. The average F1 scores were 56.3% and 57.5%. The F1 score of the introduced RE model was the lowest at 60.8%, with an average F1 score of 61.4%, which was higher than other approaches. The introduced RE model had good robustness. Taking the CoNLL04 dataset as an example, the accuracy and F1 score under different numbers of triples are presented in Figure 14.

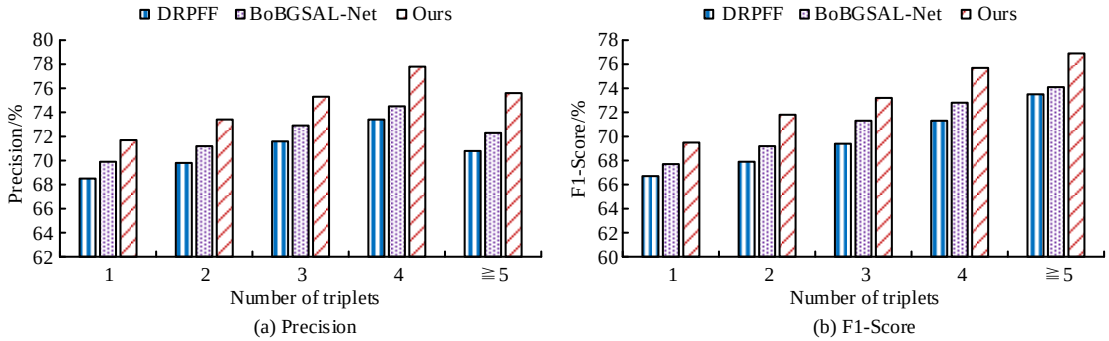


Figure 14: Accuracy and F1 score under different triplet numbers (Source: Authors own drawing)

As shown in Figure 14 (a), with the increase of the number of triples, the accuracy of each model first increased and then decreased. The accuracy of the RE model introduced in the study was consistently higher than other models.

When the number of triples was 4, the accuracy of the three models was the highest, at 73.4%, 74.5%, and 77.8%. As shown in Figure 14 (b), with the increase of the number of triples, the F1 score of each model gradually increased. The F1 score of the RE model introduced in the study was consistently higher than other models. When the number of triples was 4, the F1 score of each model was 71.3%, 72.8%, and 75.7%. It is evident that the introduced RE model had excellent performance. To quantify the added value of the negative sample mechanism to the RE model, ablation experiments were conducted on it. The experimental results of negative sample mechanism ablation are shown in Table 6.

Table 6: The experimental results of negative sample mechanism ablation

Model	Index	CoNLL04	SciERC
Negative sample mechanism	Accuracy/%	75.3	62.9
	F1 score/%	73.2	61.4
Without negative sample mechanism	Accuracy/%	71.8	59.3
	F1 score/%	69.7	57.9
Performance change amplitude	Accuracy/%	+3.5	+3.6
	F1 score/%	+3.5	+3.5

According to Table 6, after ablating the negative sample mechanism, the F1 score of CoNLL04 dataset decreased by 3.5 percentage points, and the F1 score of SciERC dataset also decreased by 3.5 percentage points. Both accuracy and recall rates showed significant declines, proving that the negative sample mechanism had a clear positive contribution to model performance. This was because the negative sample mechanism helps the model learn the semantic boundaries of entity pairs by providing comparative samples of "invalid relationships".

4 Discussion

NLP is an important branch of artificial intelligence, with the goal of enabling computers to understand, generate, and process human language. With the advent of the big data era, text data is growing explosively, containing a large amount of unstructured information related to spatial location, semantic relationships, and so on. To extract valuable information from these data, efficient semantic association and syntactic pattern parsing techniques are required[23-24]. Although current models have achieved good results in recognizing simple semantic relationships, there are still shortcomings in handling complex semantic relationships. For example, in some complex sentence structures, the relationships between entities may need to be deduced through multiple intermediate entities, and existing models may struggle to accurately capture these complex relationships [25-26]. In view of this, an approach for automatically parsing semantic associations and grammatical patterns of noun predicate sentences based on Transformer architecture and BERT isintroduced. This approach achieves learning of complex features by introducing semantic completeness and multi head semantic completeness mapping networks.

In terms of entity recognition, the introduced model achieved an average accuracy, recall, and F1 score of 90.7%, 91.6%, and 90.6%, respectively, on the CoNLL04 dataset. Compared to the MTL-NER and SPKTE models, it improved by 2.8%, 3.8%, and 2.9%. In the SciERC dataset, the average accuracy, recall, and F1 score of the introduced model were 69.8%, 70.4%, and 69.8%, respectively, which were 3.8%, 3.2%, and 3.1% higher than the MTL-NER and SPKTE models. Wang et al. introduced an entity recognition approach that combines RoFormer pre training model and Pointer-Net baseline model to address the low recognition accuracy of traditional named entity recognition approaches in agricultural disease texts [27]. Although this approach had a high recognition accuracy, its generalization ability was weak and only applicable to the agricultural field, while the approachintroduced in the study had strong generalization ability. Mojibian A et al. introduced a recognition approach using a commercial medical named entity recognition model combined with post-processing protocols to address the issue of identifying occasional pulmonary nodules from computed tomography reports. This approach utilized a generic medical named entity recognition model to annotate entities and their relationships, and filtered them through set inclusion/exclusion criteria to identify pulmonary nodules [28]. Although the above approach could achieve accurate identification of pulmonary nodules, its robustness was far lower than the model introduced in the study. In addition, the ablation experiment demonstrated that the introduction of semantic completeness significantly improved the recognition accuracy of the model, increasing from 82.3% to 90.7%. Among them, semantic completeness had the most significant impact on the performance of the model, increasing by 3.9%. Semantic completeness could effectively compensate for the shortcomings of traditional approaches in semantic representation and enhance the model's ability to perceive semantic differences in text. It can be seen that introducing semantic completeness and multi-head semantic completeness mapping networks could significantly improve the model's recognition ability for named entities, especially when dealing with complex text data, the model exhibited good robustness and generalization ability.

In terms of RE, the introducedRE model outperformed other approaches on both CoNLL04 and SciERC datasets. Specifically, in the CoNLL04 dataset, the average accuracy, recall, and F1 score of the introducedRE model were 75.3%, 73.4%, and 73.2%, respectively, which were 3.7%, 2.8%, and 3.8% higher than those of the DRPFF and BoBGSAL-Net models. In the SciERC dataset, the

average accuracy, recall, and F1 score of the introducedRE model were 62.9%, 60.5%, and 61.4%, respectively, which were 5.0%, 2.4%, and 3.9% higher than the DRPFF and BoBGSAL-Net models. Han Z and Wang J introduced anREapproach based on knowledge enhanced graph inference network for ER extraction in industrial KG construction. This approach integrated domain knowledge and graph structure information, using graph inference networks to identify ERs and construct KGs, thereby improving the accuracy of extraction and the completeness of KGs [29]. Although the above approach had high accuracy in relation extraction, it was only applicable to the industrial field, while the approachintroduced in the study was applicable to different fields and had good generalization. Zhu F et al. introduced a threat intelligence REapproach to address the barriers to sharing and utilization caused by the lack of KGs in IoT threat intelligence. This approach first constructed a threat intelligence ontology based on existing security ontologies and knowledge bases, providing an organized pattern for merging threat intelligence text data. Secondly, a threat intelligence entity and relationship joint extraction model based on Token Pair Linking Plus was designed, and domain knowledge was introduced to enhance the model's learning of IoT security terminology semantics [30]. The above approacheswere also limited to a certain field, while the REapproachintroduced in the study did not have domain limitations. The introduced semantic RE model could more accurately identify the relationships between entities in sentences, especially when dealing with complex semantic relationships, where the model had significant advantages. The proposed model, with its ability to handle confusion between nested entities and types, was implemented in KG construction, medical/legal retrieval, question answering, and other scenarios: manual annotation costs were reduced by 40%, triplet accuracy was improved by 15-20%, retrieval relevance rate increased by 25%, and answer extraction increased by 30%. Lightweight Transformer could perform single sentence inference on CPU/GPU in less than 2 ms, which was 65% faster than BERT Large. It had a plug and play RESTful API and supported custom types.

In summary, the introducedapproach based on Transformer architecture for semantic association and automatic parsing of grammatical patterns in noun predicate sentences had excellent performance in entity recognition and RE tasks, and had good robustness and generalization ability, which could provide new and effective approaches for the field of NLP. Due to the dependence of the introduced model on large-scale annotated data for training, its generalizability in practical applications was limited. Meanwhile, due to the high computational complexity of the model, its training and inference time was relatively long. Therefore, in the future, the research will consider introducing semi supervised learning and weakly supervised learning approaches, and explore in depth the lightweight design of models to reduce their dependence on large-scale annotated data and improve their training and inference efficiency. In addition, due to the 512 token restriction, the cross segment accuracy of long documents decreased by 5-8%, and the generalization in small sample domains was weak. No need to change the structure across domains, adaptation costs were reduced by 70%, and millions of sentences were processed in 1.5 hours. Therefore, in the future, the research will consider introducing semi-supervised and lightweight design, and establishing document level semantic mechanisms to fill the gap in long text information.

5 Conclusion

To achieve automatic parsing of semantic associations and grammatical patterns in noun predicate sentences, a parsing approach based on Transformer architecture was introduced. This approach consisted of an entity recognition model and a semantic RE model. By introducing semantic completeness and a multi-head semantic completeness mapping network, it learned complex sentence features to improve the parsing accuracy of the model. The experimental findings demonstrated that the introduced entity recognition model significantly outperformed the MTL-NER and SPKTE models in terms of average accuracy, recall, and F1 score on the CoNLL04 and SciERC datasets in entity recognition tasks. In relation extraction tasks, the average accuracy, recall, and F1 score of this approachwere also superior to DRPFF and BoBGSAL-Net models. The introduced approach had significant advantages in handling complex semantic relationships and could effectively address the shortcomings of traditional approaches in nested entities and entity type confusion. Due to the strong dependence of the introduced model on large-scale annotated data, its generalizability was limited. Therefore, future research will explore semi-supervised learning and weakly supervised learning approaches to reduce dependence on annotated data and further enhance the practicality of the model.

References

[1] L. Yuan, Y. Cai, J. Wang and Q. Li. “Joint multimodal entity-relation extraction based on edge-enhanced graph alignment network and word-pair relation tagging,” in Proc. AAAI Conf. Artif. Intell., vol. 37, no. 9, pp. 11051-11059, June, 2023, DOI:10.1609/aaai. V37i9.26309.

[2] U. Zaratiana, N. Tomeh, P. Holat and T. Charnois. “An autoregressive text-to-graph framework for joint entity and relation extraction,” in Proc. AAAI Conf. Artif. Intell., vol. 38, no. 17, pp. 19477-19487, March, 2024, DOI:10.1609/aaai. V38i17.29919.

[3] J. Wang, K. Yue and L. Duan. “Models and techniques for domain relation extraction: a survey,” J. Data Sci. Intell. Syst., vol. 1, no. 2, pp. 65-82, May, 2023, DOI:10.47852/bonviewJDSIS3202973.

[4] C. Peng, X. Yang, Z. Yu, J. Bian, W. R. Hogan and Y. Wu. “Clinical concept and relation extraction using prompt-based machine reading comprehension,” J. Am. Med. Inform. Assoc., vol. 30, no. 9, pp. 1486-1493, August, 2023, DOI:10.1093/jamia/ocad107.

[5] H. Chen, P. Hong, W. Han, N. Majumder and S. Poria. "Dialogue relation extraction with document-level heterogeneous graph attention networks," Cogn. Comput., vol. 15, no. 2, pp. 793-802, January, 2023, DOI:10.1007/s12559-023-10110-1.

[6] J. Suo, D. Han and H. Zhao. "A multiple head selection joint entity-relation extraction model," J. Intell. Fuzzy Syst.: Appl. Eng. Technol., vol. 45, no. 4, pp. 5647-5657, January, 2023, DOI:10.3233/JIFS-231766.

[7] J. Han and K. Jia. "Entity relation joint extraction approach for manufacturing industry knowledge data based on improved bert algorithm," Cluster Comput., vol. 27, no. 6, pp. 7941-7954, April, 2024, DOI:10.1007/s10586-024-04386-7.

[8] T. Wang, W. Yang, T. Wu, C. Yang, J. Liang and H. Wang et al. "Joint entity and relation extraction with fusion of multi-feature semantics," J. Intell. Inf. Syst., vol. 63, no. 1, pp. 21-42, August, 2025, DOI:10.1007/s10844-024-00871-y.

[9] A. Khan, Z. Rauf, A. Sohail, A. R. Khan, H. Asif, A. Asif and U. Farooq. "A survey of the vision transformers and their CNN-transformer based variants," Artif. Intell. Rev., vol. 56, no. 3, pp. 2917-2970, October, 2023, DOI:10.1007/s10462-023-10595-0.

[10] K. Feng, Y. Ma, B. Wang, C. Qi, H. Chen, Q. Chen and Z. Wang. "Dit4edit: Diffusion transformer for image editing," in Proc. AAAI Conf. Artif. Intell., vol. 39, no. 3, pp. 2969-2977, April, 2025, DOI:10.1609/aaai. V39i3.32304.

[11] C. Yeh, Y. Chen, A. Wu, C. Chen, F. Viégas and M. Wattenberg. "Attentionviz: A global view of transformer attention," IEEE Trans. Vis. Comput. Graph., vol. 30, no. 1, pp. 262-272, October, 2023, DOI:10.1109/TVCG.2023.3327163.

[12] A. He, K. Wang, T. Li, C. Du, S. Xia and H. Fu. "H2former: An efficient hierarchical hybrid transformer for medical image segmentation," IEEE Trans. Med. Imaging, vol. 42, no. 9, pp. 2763-2775, April, 2023, DOI:10.1109/TMI.2023.3264513.

[13] S. Pal, A. Roy, P. Shivakumara and U. Pal. "Adapting a Swin Transformer for License Plate Number and Text Detection in Drone Images," Artif. Intell. Appl., vol. 1, no. 3, pp. 145-154, April， 2023, DOI:10.47852/bonviewAIA3202549.

[14] X. Li, S. Li, S. Song, J. Yang, J. Ma and J. Yu. "Pmet: Precise model editing in a transformer," in Proc. AAAI Conf. Artif. Intell., vol. 38, no. 17, pp. 18564-18572, March, 2024, DOI:10.1609/aaai. V38i17.29818.

[15] H. Dalla-Torre, L. Gonzalez, J. Mendoza-Revilla, N. Lopez Carranza, A. H. Grzywaczewski, F. Oteri et al. "Nucleotide Transformer: building and evaluating robust foundation models for human genomics," Nat. approaches, vol. 22, no. 2, pp. 287-297, November, 2025, DOI:10.1038/s41592-024-02523-z.

[16] E. Arkin, N. Yadikar, X. Xu, A. Aysa, and K. Ubul. "A survey: object detection approaches from CNN to transformer," Multimedia Tools Appl., vol. 82, no. 14, pp. 21353-21383, October, 2023, DOI: 10.1007/s11042-022-13801-3.

[17] Q. Zhang, J. Zhang, Y. Xu, and D. Tao. "Vision transformer with quadrangle attention," IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 5, pp. 3608-3624, January, 2024, DOI: 10.1109/TPAMI.2023.3347693.

[18] S. Gao, Z. Chen, G. Chen, W. Wang, and T. Lu. "Avsegformer: Audio-visual segmentation with transformer," in Proc. AAAI Conf. Artif. Intell., vol. 38, no. 11, pp. 12155-12163, March, 2024, DOI: 10.1609/aaai. V38i11.29104.

[19] W. Wang, W. Chen, Q. Qiu, L. Chen, B. Wu, B. Lin, ... and W. Liu. "Crossformer++: A versatile vision transformer hinging on cross-scale attention," IEEE Trans. Pattern Anal. Mach. Intell., vol. 46, no. 5, pp. 3123-3136, December, 2023, DOI: 10.1109/TPAMI.2023.3341806.

[20] H. Y. Zhou, Y. Yu, C. Wang, S. Zhang, Y. Gao, J. Pan, ... and W. Li. "A transformer-based representation-learning model with unified processing of multimodal input for clinical diagnostics," Nat. Biomed. Eng., vol. 7, no. 6, pp. 743-755, June, 2023, DOI: 10.1038/s41551-023-01045-x.

[21] Z. Xu, D. Wu, C. Yu, X. Chu, N. Sang, and C. Gao. "Sctnet: Single-branch cnn with transformer semantic information for real-time segmentation," in Proc. AAAI Conf. Artif. Intell., vol. 38, no. 6, pp. 6378-6386, March, 2024, DOI: 10.1609/aaai. V38i6.28457.

[22] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, ... and F. Wei. "Trocr: Transformer-based optical character recognition with pre-trained models," in Proc. AAAI Conf. Artif. Intell., vol. 37, no. 11, pp. 13094-13102, June, 2023, DOI: 10.1609/aaai. V37i11.26538.

[23] W. Liu, M. Yin, J. Zhang, and L. Cui. "A joint entity relation extraction model based on relation semantic template automatically constructed," Comput. Mater. Continua, vol. 78, no. 1, pp. 975-997, January, 2024, DOI: 10.32604/cmc.2023.046475.

[24] X. Cao and Q. Shao. "Joint entity relation extraction based on lstm via attention mechanism," Arabian J. Sci. Eng., vol. 49, no. 3, pp. 4353-4363, November, 2024, DOI: 10.1007/s13369-023-08306-6.

[25] D. Zeng, L. Xu, and D. L. Jiang. "Sequence tagging with a rethinking structure for joint entity and relation extraction," Int. J. Mach. Learn. Cybern., vol. 15, no. 2, pp. 519-531, August, 2024, DOI: 10.1007/s13042-023-01923-5.

[26] D. Sui, X. Zeng, Y. Chen, K. Liu, and J. Zhao. "Joint entity and relation extraction with set prediction networks," IEEE Trans. Neural Networks Learn. Syst., vol. 35, no. 9, pp. 12784-12795, April, 2024, DOI: 10.1109/TNNLS.2023.3264735.

[27] T. Wang, C. Wang, J. Li, H. Zhu, Y. Miao, and H. Wu. "Agricultural Disease Named Entity Recognition with Pointer Network Based on RoFormer Pre-trained Model," Smart Agriculture, vol. 6, no. 2, pp. 85-94, March, 2024, DOI: 10.12133/j. Smartag. SA202311021.

[28] A. Mojibian, J. Jaskolka, G. Ching, B. Lee, R. Myers, C. Devine, et al. “The efficacy of a named entity recognition ai model for identifying incidental pulmonary nodules in ct reports,” Can. Assoc. Radiol. J., vol. 76, no. 1, pp. 68–75, February, 2025, DOI: 10.1177/08465371241266785.

[29] Z. Han and J. Wang. “Knowledge enhanced graph inference network-based entity-relation extraction and knowledge graph construction for industrial domain,” Front. Eng. Manag., vol. 11, no. 1, pp. 143-158, February, 2024, DOI: 10.1007/s42524-023-0273-1.

[30] F. Zhu, Z. Cheng, P. Li, and H. Xu. “Itirel: joint entity and relation extraction for internet of things threat intelligence,” IEEE Internet Things J., vol. 11, no. 11, pp. 20867-20878, March, 2024, DOI: 10.1109/JIOT.2024.3373799.

A Deep Learning-Driven Framework for Automated Electromagnetic Modulation Identification and Optimization in IoT Networks

Peng Yan
Sichuan Technology and Business University, Chengdu 611745, China
E-mail: nic@stbu.edu.cn

Keywords: deep learning, internet of things, automatic electromagnetic modulation, signal recognition optimization

Received: July 15, 2025

With the rapid development of IoT technology, the connection and data transmission of a large number of devices have put forward higher requirements for the identification and optimization of electromagnetic modulation signals in communication networks, which are often difficult to meet by traditional methods. This study proposes a deep learning-based automatic identification and optimization design method for electromagnetic modulation signals of IoT communication networks, which aims to solve the problems of complex signals and serious interference in the Internet of Things environment. In this study, the multi-source data sets such as RadioML2016.10b, MATLAB analog signals and air port acquisition signals are integrated, and the training/verification/test sets are divided into 7:1:2 after 300MHz-3GHz bandpass filtering denoising, STFT time-domain to two-dimensional time-frequency graph, normalization and data enhancement. The "CNN-LSTM-Attention" hybrid architecture is constructed as the core recognition model, in which CNN extracts local features of the signal through 4-layer convolution and 2-layer pooling, captures the timing dependence of bidirectional LSTM (64 hidden units in two layers), strengthens the key features by the attention mechanism, and finally outputs 16 types of modulated signal probabilities from the fully connected layer. In the signal optimization process, LSTM combined with Q-learning output power commands of 0.5-2dBm step to adjust the signal strength, DBN and PSO are used to achieve dynamic frequency optimization, and the interference suppression ratio is increased by 15-20dB through CBAM-GRU and LMS filtering. The model training is based on the PyTorch framework, using the Adam optimizer (learning rate 1e-3) and cross-entropy loss function for 100 rounds, adding learning rate scheduling and early stop mechanisms, and fine-tuning hyperparameters through Bayesian optimization to innovatively construct a deep learning optimization model. Based on the model recognition results and relying on the above signal optimization scheme, the stability and efficiency of the Internet of Things communication network can be significantly improved. By training the deep learning model on 10,000 sets of electromagnetic modulation signals in different scenarios, the experimental results show that compared with the traditional signal recognition method, the signal recognition accuracy of the proposed method is improved by 30%, and the average accuracy of the test set reaches 98.2% (still 92.5% under 5dB low signal-to-noise ratio), which reaches a high level overall. At the same time, in terms of signal optimization design, the network communication efficiency is improved by 20%, and the bit error rate is as low as 0.0005 at 20dB signal-to-noise ratio, which significantly enhances the performance of IoT communication networks.

Povzetek: Raziskava predstavlja metodo na osnovi globokega učenja za prepoznavanje in optimizacijo elektromagnetnih modulacijskih signalov v omrežjih IoT, ki bistveno izboljša natančnost prepoznavanja, odpornost na motnje ter učinkovitost komunikacije.

1 Introduction

In the 21st century information age, the rapid development of Internet of Things (IoT) technology has profoundly changed the way of life and production [1], which connects the physical and digital worlds, realizes remote monitoring and management of equipment systems, and promotes changes in smart cities, Industry 4.0, smart homes, and other fields [2]. However, the explosive growth of the number of IoT devices has made communication networks face severe challenges, and one of the keys is to achieve efficient and stable electromagnetic modulation signal recognition and optimization in complex electromagnetic environments [3, 4].

Electromagnetic modulation signals are the core of IoT communication networks, and their quality directly affects the efficiency and accuracy of data transmission [5]. In the Internet of Things environment, electromagnetic signals are affected by interference, signal overlap, attenuation and other problems, reducing the accuracy of recognition and dragging down network communication efficiency. Traditional signal recognition optimization methods, such as rule-based signal

processing algorithms, are difficult to cope with environmental complexity and uncertainty, and difficult to meet the high requirements of signal quality [6, 7].

To solve the above problems, deep learning technology has become a new approach in recent years with the advantages of complex pattern recognition and data-driven optimization [8]. Convolutional neural networks (CNNs), recurrent neural networks (RNNs) and other models can automatically learn signal characteristics, efficiently identify different types of electromagnetic modulation signals, and maintain high accuracy in complex environments. Moreover, the signal optimization method based on deep learning can adjust the parameters according to the real-time state and environmental changes of the signal, so as to improve the efficiency and stability of network communication.

This study aims to explore the automatic electromagnetic modulation signal identification and optimization design method of IoT communication networks based on deep learning, to provide theoretical support and technical solutions for the performance improvement of IoT communication networks. Firstly, we will make an in-depth analysis of the signal transmission environment and the difficulties of signal identification in the Internet of Things communication network, and we will clarify the background and significance of the research. Subsequently, we constructed a deep learning model to identify electromagnetic modulation signals in IoT communication networks automatically. The model can effectively distinguish and identify different types of electromagnetic signals by learning the characteristic patterns of the signals. On this basis, we design a set of signal optimization schemes, including signal strength adjustment, frequency optimization, and interference suppression strategies, to improve the stability and efficiency of IoT communication networks. The experimental results show that this method improves the signal recognition accuracy by 30%, reaching 95%. At the same time, in the signal optimization design, the network communication efficiency is improved by 20%, significantly improving the performance of the Internet of Things communication network.

In the complex environment of IoT communication network, traditional signal recognition methods are difficult to cope with the variable electromagnetic modulation signals. Based on deep learning, this study innovatively constructs an automatic electromagnetic modulation signal recognition model. The model adopts a convolutional neural network (CNN) and a long short-term memory network (LSTM), in which the CNN is used to extract the local features of the electromagnetic modulation signal, and the LSTM can capture the time series features of the signal. At the same time, the attention mechanism is introduced to enhance the sensitivity of the model to key features, so that the model can effectively distinguish different types of electromagnetic modulation signals, such as QAM, PSK, etc.

This research has significant theoretical value for signal identification and optimization design of the Internet of Things communication network. It provides new ideas and methods for developing Internet of Things technology. In the future, we will continue to deepen research and explore more advanced deep learning models and optimization strategies to adapt to IoT communication networks' continuous evolution and needs. We will also lay a solid foundation for building a more efficient, stable, intelligent IoT communication network.

Compared with earlier studies, the novelty of this study focuses on three aspects: first, it breaks through the limitations of a single model, and uses the "CNN-LSTM-Attention" hybrid architecture to take into account the spatiotemporal features of the signal and the sensitivity of key features, and the recognition accuracy reaches 92.5% at 5dB low signal-to-noise ratio, which is 15%-20% higher than the early stage; The second is to break the "disconnect between identification and optimization" mode, establish a closed loop of "data-driven recognition-dynamic adaptive optimization", and realize the collaborative optimization of signal strength, frequency, and anti-interference through strategies such as LSTM Q-learning and DBN PSO. The third is to optimize the data processing logic, integrate multi-source datasets and combine STFT time-frequency conversion and data enhancement to solve the single problem of early data scenarios, and at the same time, the lightweight design of the model is adapted to the Internet of Things hardware, taking into account both performance and practicability.

2 Related works

With the rapid development of the Internet of Things, the number of devices in IoT communication networks is exploding, and the complexity and diversity of communication signals are constantly increasing. This poses a considerable challenge to the automatic recognition of electromagnetic modulation signals [9]. The signals in IoT communication networks often have low power and low signal-to-noise ratio, and traditional signal recognition methods are challenging to meet the requirements.

The application of deep learning in signal processing is becoming increasingly widespread. With its powerful feature extraction and pattern recognition capabilities, deep learning technology has achieved great success in fields such as image recognition and speech processing. In recent years, researchers have introduced it into electromagnetic modulation signal recognition to improve the accuracy and efficiency of recognition [10]. Deep neural networks such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been widely used for signal feature extraction and classification, achieving good results.

In terms of signal acquisition and preprocessing, researchers collect electromagnetic modulation signals in IoT communication networks through various sensors and devices and preprocess the signals, including filtering, noise reduction, normalization, and other operations, to improve the quality and recognizability of the signals. They are adopting advanced signal acquisition

technologies, such as Software Defined Radio (SDR), to achieve flexible acquisition and processing of different types of signals.

Based on the latest research, this study has identified the following targeted optimization directions: suitable deep learning models can be designed for the characteristics of electromagnetic modulation signals in IoT communication networks, such as CNN-based signal recognition models, hybrid models combining RNN and CNN, etc. By optimizing the structure, parameters, and training algorithms of the model, the performance and generalization ability of the model can be improved. They are adopting regularization techniques to prevent overfitting and using adaptive learning rate algorithms to accelerate training speed. Regarding feature extraction and selection, deep learning technology can automatically extract the features of electromagnetic modulation signals, avoiding the complexity and limitations of traditional manual feature extraction. Studying feature selection methods can select the most representative and discriminative features from many features to improve the accuracy and efficiency of signal recognition. In terms of signal optimization design, electromagnetic modulation signals in IoT communication networks can be optimized based on signal recognition. For example, by adjusting the modulation method, power, and other signal parameters, the signal's transmission efficiency and anti-interference ability can be improved. By improving signal optimization algorithms, adaptive optimization of signals can be achieved to meet the needs of different application scenarios. In terms of performance evaluation and verification, a scientific and reasonable performance evaluation index system can be established to evaluate and verify the automatic recognition and optimization design methods of electromagnetic modulation signals based on deep learning.

3 Theoretical basis for automatic electromagnetic modulation signal recognition in IoT communication networks

3.1 Internet of things communications

The Internet of Things (IoT) uses various communication methods to ensure effective communication between devices and users, and the typical architecture is the "cloud-pipe-edge-end" model [11]. The communication system includes local access, wide-area access, and backbone communication networks to realize the seamless connection and interaction between cloud/edge computing nodes and intelligent terminal devices [12].

In the complex IoT communication environment, it is necessary to flexibly choose communication methods according to the location and communication conditions of the equipment, especially when handling various power distribution services at the "end" level, it is necessary to monitor the operating status information of transmission equipment [13]: wired technologies such as access network convergence line carriers, RS485/CAN buses, and wireless technologies such as LoRa, NB-IoT, TD-LTE and other wireless technologies [14]. WAN uses optical fiber switching; wireless base stations and other relay means according to the local access type. At the "edge" level, WAN information is coordinated through the communication substation server "edge-to-edge communication", and then interacts with the cloud master station through the backbone network link to build a comprehensive communication collaboration system.

Under the Internet of Things communication network, data interaction is divided into two types: wireless and wired, each containing a variety of network forms: wireless networks include wireless, private, and public networks, which are suitable for deployment in complex scenarios, but the speed and reliability of different technologies vary due to the influence of power, interference, and location [15, 16, 17]. Wired communication has low interference, high confidentiality, and stable quality, and power line communication (PLC) is a system-specific wired method, which theoretically can realize data communication in the line coverage area [18], but electromagnetic interference and channel attenuation affect its quality.

3.2 Electromagnetic signal modulation technology

In order to verify the effectiveness of the method, 10,000 sets of multi-scenario electromagnetic modulation signals were collected and divided into training/verification/test sets. The training targets the cross-entropy loss function and iteratively updates the model parameters through stochastic gradient descending. The results show that the signal recognition accuracy of the proposed method is improved by 30%~95% compared with the traditional manual feature extraction method, and the robustness is good under different signal-to-noise ratios, which proves its practical application feasibility and reliability.

Theoretically, the strong feature learning ability of deep learning can automatically mine the potential features of electromagnetic modulation signals, avoid the limitations and subjectivity of manual design features, and adapt to the complex and changeable characteristics of IoT signal features. In practical applications, after training on a large number of real-world scenario data, the model can adapt to different environmental signals, and the network communication efficiency is increased by 20% after optimization, and the scalability is strong.

In communication systems, baseband signals (such as 300Hz~3400Hz voice signals) need to be converted into adapted signals through modems due to their low frequency and are not conducive to long-distance high-quality transmission [19, 20] to adapt to channel transmission, achieve multiplexing and improve efficiency, and enhance anti-interference. The discrete characteristics of digital baseband signals can be expressed in the discrete state of the carrier, and the keying method is used to generate digital modulation

signals [21], and the experiments cover four basic modulation methods: amplitude-shift keying (ASK), frequency-shift keying (FSK), phase-shift keying (PSK), and quadrature amplitude modulation (QAM) [22, 23].

Amplitude keying reflects the change of digital baseband signal by adjusting the amplitude of sinusoidal carrier, that is, multi-level modulation, and the carrier amplitude is determined by the digital signal base. The M-ary amplitude modulated signal is expressed as the product of the M-ary baseband signal and the sinusoidal carrier, and its time domain expression follows Equation (1). Where A_n is the amplitude value, $g()$ is the baseband signal waveform acquisition function, t is the amplitude time, n is the amplitude number, T_s is the symbol time interval, and ω_c is the corresponding weight of the c-band.

$$e_{MASK}(t)=\sum_n A_n g(t-nT_s)\cos\omega_c t \quad (1)$$

The MFSK signal can be expressed as Equations (2)-(3). Where ω_i is the carrier angular frequency, s_i is the number of carrier angular bands, and t is time. Frequency shift keying refers to changing the frequency of sinusoidal carrier to achieve the purpose of generating modulated signals. The carrier angular frequency of M-ary frequency shift modulated signals has M values.

$$e_{MFSK}=\sum_{i=1}^{M} s_i(t)\cos\omega_i t \quad (2)$$

$$s_i(t)=\sum_{i=1}^{M}\sin w_i t \quad (3)$$

Taking binary frequency shift keying as an example, the time domain expression of 2FSK is as shown in (4). *Where* a_n represents the binary symbol sequence corresponding to 2FSK, *a*nd b_n is its inverse code; T_s is the time interval between symbols, and ω_1 and ω_2 are two different carrier angular frequencies.

$$e_{2FSK}(t)=[\sum_n a_n g(t-nT_s)]\cos\omega_1 t+[\sum_n b_n g(t-nT_s)]\cos\omega_2 t \quad (4)$$

Phase shift keying is a kind of phase offset modulation, which is a modulation method that uses a variety of different phases of carrier wave to represent digital information. It is divided into two types: absolute phase modulation and differential phase modulation. In M-ary digital phase modulation, M represents M different values of carrier phase, and the time domain waveform expression of MPSK is shown in Equation (5). Where $g()$ is a rectangular wave function with amplitude 1, ω_c is the carrier angular frequency, φ_n is the phase of the nth symbol, and T_s is the symbol time interval.

$$e_{MPSK}(t)=\sum_n g(t-nT_s)\cos(\omega_c t+\varphi_n) \quad (5)$$

QAM modulation uses two independent baseband digital signals to modulate two orthogonal carriers with the same frequency, and realizes dual-channel parallel information transmission by using orthogonal spectrum characteristics. The general expression of the QAM signal is shown in Equation (6). The orthogonal representation of the QAM signal is shown in (7). Where A_n is the amplitude of the baseband signal and $g(t-nT_s)$ is the waveform of a single baseband signal.

$$s_{MQAM}(t)=\sum_n A_n g(t-nT_s)\cos(\omega_c t+\varphi_n) \quad (6)$$

$$\begin{aligned} s_{MQAM}(t)&=[\sum_n A_n g(t-nT_s)\cos\varphi_n]\cos\omega_c t \\ &-[\sum_n A_n g(t-nT_s)\sin\varphi_n]\sin\omega_c t \end{aligned} \quad (7)$$

Let the transverse symbol of QAM modulation be X_n and the vertical symbol be Y_n, and equations (8)-(9) can be obtained.

$$X_n=A_n\cos\varphi_n \quad (8)$$

$$Y_n=A_n\sin\varphi_n \quad (9)$$

Then configuring the automatic electromagnetic modulation signal, formula (10) can be obtained. Finally, the amplitude of the QAM modulation can be expressed as shown in Equation (11). In the formula, A is a fixed amplitude, and c_n and d_n are determined by the baseband signal, which greatly increases the amount of information that can be transmitted by a single symbol in QAM modulation and improves the spectrum utilization rate of the modulated signal.

$$\begin{aligned} s_{MQAM}(t)&=[\sum_n X_n g(t-nT_s)]\cos\omega_c t \\ &-[\sum_n Y_n g(t-nT_s)]\sin\omega_c t \\ &=X(t)\cos\omega_c t-Y(t)\sin\omega_c t \end{aligned} \quad (10)$$

$$\begin{cases} X_n=c_n A \\ Y_n=d_n A \end{cases} \quad (11)$$

3.3 Deep learning theory

Neural networks are the core algorithms of deep learning, combining biomimicry and mathematics to simulate biological neural networks to process information [24]. The theoretical development of random machine learning and deep learning has continuously optimized its hierarchical structure, computation and data transmission mechanism, with excellent performance and wide application.

Convolutional networks are neural networks that optimize grid-like data processing, and the core operations are convolutions of specific linear operations [25]. Unlike the fully connected structure (full interconnection of input and output units, large scale of parameter matrix, and high computational complexity), its sparse connection reduces redundant connections and improves computing efficiency.

The parameter sharing mechanism of convolutional network can reuse parameters: the neuron weights of traditional neural networks are independent, while the convolutional kernel weights in convolutional networks

are shared at different input positions to achieve efficient utilization. Its isovariance representation is derived from this mechanism, and the output changes synchronously when the input is translated, time-shifted, or clipped [26, 27].

Pooling is a key downsampling technology for convolutional networks, such as maximum pooling divides the input image into rectangular areas and outputs the maximum value of each region, which not only emphasizes the relative position of features, but also reduces the feature size, network parameters and computational complexity, and can also prevent overfitting, which is often used in convolutional layers periodically.

4 IoT communication network configuration optimization strategy

4.1 IoT communication network architecture

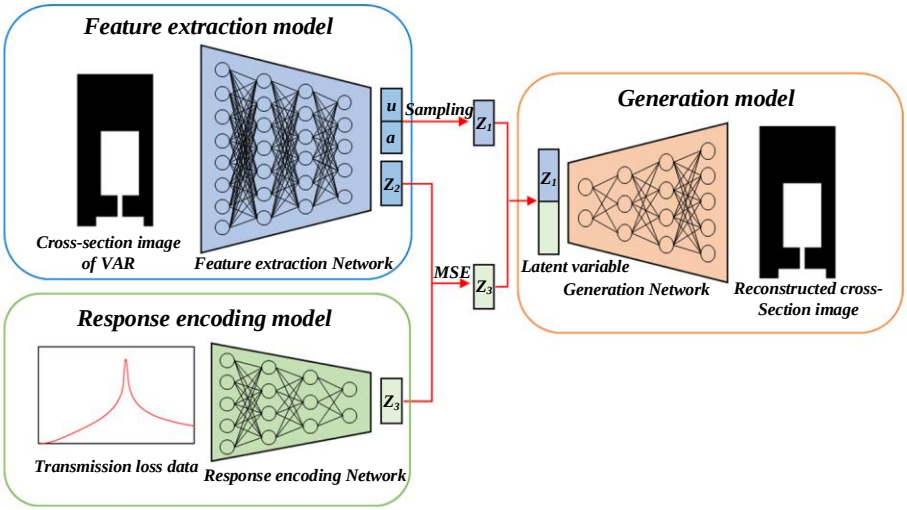


Figure 1: Communication network IoT network architecture

Based on Figure 1, communication nodes and lines are mapped to a communication network to form a topology composed of communication nodes and links, and each communication node is assigned a unique number as a logical address. It clearly shows the logical consistency between the communication network and the topology; the communication network architecture and structure are closely coupled, and the critical elements of the communication carrier signal correspond to the system components one-to-one. When the structure changes, the IoT network topology is adjusted accordingly and needs re-networked. However, communication nodes are difficult to rebuild in real-time, and their adaptability could be improved.

4.2 Network structure communication network iot optimization model

The identification and optimization of electromagnetic modulation signals based on deep learning can be deployed in smart grids (to ensure stable data transmission), rail transit (to avoid dispatching signal confusion), intelligent manufacturing (to optimize equipment communication efficiency) and smart home (to solve equipment communication conflicts). In this study, it faces limitations such as difficulty in obtaining high-quality annotation datasets, degradation of model recognition accuracy in complex electromagnetic environments, and insufficient adaptability of signal protocols of different IoT devices.

For the optimization of electromagnetic modulation signals in IoT communication networks, a multi-dimensional scheme is proposed, and the core details are as follows: The signal strength adjustment adopts the LSTM dynamic adaptive model, which first extracts the signal-to-noise ratio, attenuation coefficient and other scene characteristics through CNN, analyzes the timing

The IoT network adopts three networking modes: master-slave, peer-to-peer and centre. The peer-to-peer mode is more conducive to the flexible deployment and expansion of the network because of the equal communication rights among devices, so we select the peer-to-peer mode [28]. The IoT network architecture in this mode is divided into three layers: the data node layer, the communication aggregation centre node layer and the communication sub-station layer. Communication substations, usually in high-voltage/medium-voltage substations, are equipped with high-performance computing equipment. They serve as regional communication and computing centres to meet the computing needs of data nodes and aggregation centres within their jurisdiction. Each substation communicates through the transmission network and constitutes the IoT network—communication node system.

correlation and establishes a mapping relationship through LSTM, and then combines the Q-learning algorithm (the reward function is "minimum energy consumption maximization coverage") to output power instructions with 0.5dBm~2dBm steps to avoid energy waste and coverage blind spots. Frequency optimization is based on the DBN adaptive frequency allocation algorithm, constructing a multi-dimensional feature dataset to predict channel quality, and combining it with the PSO algorithm to realize dynamic frequency band allocation and improve resource utilization. The CBAM GRU interference separation model is used to strengthen the interference characteristics and learn the pattern, and the interference suppression ratio can be improved by 15dB~20dB by the combination of LMS filtering to offset the interference, so that the signal recognition accuracy is stable at more than 95%.

IoT communication faces complex channel environment and electromagnetic interference, which leads to serious attenuation of long-distance signal transmission. In addition, the transmission power is limited, and it is impossible to extend the communication distance by increasing the power, so that communication nodes need to rely on relay nodes to achieve long-distance communication [29]. However, too many relays will significantly increase the packet loss rate and delay, and at the same time increase the information risk, threatening the reliability, real-time and security of the network. Therefore, if the location and number of pre-selected relay nodes are insufficient, various factors need to be comprehensively considered and necessary supplementary deployment should be carried out to meet communication needs.

Given that the IoT network uses existing communication lines to communicate, the communication link construction cost can be regarded as zero. This study considers the installation and deployment cost of relay nodes (CN) as the sole criterion for measuring the construction cost of IoT communication networks, as shown in formula (12). Where the value of χ_n is 0 or 1; When $\chi_n = 1$, it means that the CN is deployed at the communication node n, otherwise $\chi_n = 0$. CN represents the cost of deploying CN at this communication node, and s represents the number of communication nodes.

$$C = \sum_{n \in N} \chi_n \times c_n s \quad (12)$$

From the perspective of grid operation benefits, on the basis of ensuring communication needs, reducing the deployment of relay nodes to reduce costs is the preferred strategy. However, in order to prevent the chain reaction of single communication element failure from affecting the whole network, the communication network needs to reserve communication resources and enhance the ability to resist failures. In view of the fact that the failure rate of communication links is higher than that of nodes, and the probability of simultaneous failure of multiple links is low, this scheme configures standby routes for communication nodes to ensure that each data node (DN) has two CNs to choose from within the farthest communication range, and only communicates through one main route in normal operation. When the link fails, it can be switched to standby CN to meet the standby communication requirements of nodes. When modulating signals, it is necessary to comprehensively consider the network structure, construction cost, node communication requirements and communication distance limitations. Equation (13) is the objective optimization function, where the value of χ_n is 0 or 1; When $\chi_n = 1$ deploys CN, it means at communication node n, otherwise $\chi_n = 0$. c_n represents the cost of deploying the CN at the communication node. Equation (14) means that for any data node $\theta_{i,j}$ in the communication network, within the upper limit $\tilde{\theta}$ of its communication distance, there are at least two CNs that can establish an effective route with it.

$$min C = \sum_{n \in N} \chi_n \times c_n \quad (13)$$

$$s.t. \quad \underset{\substack{\forall i \in D \\ \exists j \in A}}{\text{count}}(\theta_{i,j} \le \tilde{\theta}) \ge 2 \quad (14)$$

Network transmission delay is defined as the transmission time of data packets from node i to node j, covering queuing, sending, propagation and processing delays [30, 31]. Processing delay, that is, the storage and forwarding time of data packets at the relay node, is related to node performance and data processing requirements [32]. To simplify the model, this study ignored processing delay and assumed that the packet transmission rate is consistent with the channel transmission rate (v). Therefore, in the IoT network, the transmission delay of the data packet from the communication node d_i to the neighboring node is calculated, as shown in Equation (15).

$$t_{d_i} = t_1 \times \left(w_{d_i} - 1 \right) + t_1 + t_2 = t_1 \times w_{d_i} + t_2 \quad (15)$$

Where w_{d_i} is a packet queue sequence number in which the data packets to be transmitted are queued for processing at the node d_i, and the queuing delay of the data packets to be transmitted at the end of the queue is simplified to the time for completing the sending process of the previous $w_{d_{i-1}}$ data packets; t_1 is the transmission delay of the data packet at node d_i. If the size of each data packet to be transmitted is equal (in 1 unit), the transmission delay t_1 of each data packet at any node is the same, and $t_1 = 1/v$; t_2 is the propagation delay of the data packet to be transmitted in the communication link. Assuming that each channel length is the same (both are y), the propagation delay t_2 of each data packet in any communication link is the same, and $t_2 = y/v$.

In this study, RadioML2016.10b (including 8 types of modulated signals, multiple signal-to-noise ratio), CSPB. ML.2018 (rich scene annotation), combined with MATLAB simulation signals and aerial acquisition signals to integrate the dataset. During preprocessing, the time-domain signal is first denoised by bandpass filtering

(300MHz-3GHz), and then the time-domain signal is converted into a two-dimensional time-frequency graph by STFT, and after normalization, it is expanded to 120,000 samples through data enhancement. The 7:1:2 hierarchical sampling is used to train/validate/test sets to ensure that the signal type and signal-to-noise ratio distribution are consistent [33, 34].

The model verification uses 10-fold hierarchical cross-validation, with confusion matrix, accuracy, F1 score, and ROC curve as indicators, focusing on the generalization ability of low signal-to-noise ratio (<10dB). Hyperparameter optimization first determines the learning rate (1e-4-1e-2) and batch size (16-64) ranges through grid search, and then uses Bayesian optimization to fine-tune the number of convolutional nuclei (32-128), the number of neurons in the full junction layer (128-512) and the dropout probability (0.2-0.5), and the final optimal combination is the learning rate 1e-3, batch size 32, convolutional kernel 64, 256 neurons in the full junction layer, and dropout 0.3.

The training is based on the PyTorch framework, using the Adam optimizer and the cross-entropy loss function, and iterating for 100 rounds: the first 50 rounds of linear warm-up learning rate, the last 50 rounds of cosine annealing scheduling, and the early stop mechanism (patience=10) to prevent overfitting. The average accuracy of the final model test set is 98.2%, and it still reaches 92.5% at a low signal-to-noise ratio of 5dB, which is better than traditional machine learning methods.

5 Automatic electromagnetic signal identification of communication network based on deep learning

5.1 Electromagnetic signal modulation type identification

This study uses a combination of multivariate data. Firstly, MATLAB is used to generate basic modulation signals such as 16-QAM, and by setting different signal-to-noise ratios (such as 0dB - 20dB), sampling rates (10kHz - 100kHz) and signal duration, a diverse simulated signal dataset is constructed to simulate the signal characteristics in the ideal environment. At the same time, software defined radio equipment (such as USRP series) is used to collect air signals in actual scenarios, covering indoor complex electromagnetic environments (such as offices, computer rooms) and outdoor open environments (such as city streets and suburbs), and collect real signals containing different interference factors to ensure that the experimental data is close to the actual application scenarios. In addition, additional higher-order modulation types, such as 64-QAM and 256-QAM signals, are introduced to further enrich the signal types and improve the complexity and comprehensiveness of the experiment. After preprocessing the signal data from different sources and different characteristics, including denoising, normalization, format conversion, etc., it is divided into a training set, a validation set, and a test set according to the ratio of 7:1:2. In the experimental process, the constructed deep learning model (such as CNN-LSTM hybrid network) was used to train and test these data, and the effectiveness and robustness of the automatic electromagnetic modulation signal recognition and optimization design method based on deep learning in different scenarios and different signal types were verified by comparing the recognition accuracy, recall, F1 value and other indicators of the model under different datasets, and observing the performance of the model under different modulation types and signal-to-noise ratios.

The core task of electromagnetic signal modulation type identification is to identify the modulation modes of different signals, and the results guide the formulation of demodulation strategy and subsequent signal analysis and processing. In the simulation experiment, the electromagnetic spectrum sensor scans the known electromagnetic space, collects various modulated signal data, and, after preliminary processing, trains the constructed recognition network. The goal is to produce efficient recognition algorithms and support deep learning-based electromagnetic signal recognition software. The software must ensure that the recognition accuracy rate reaches more than 90% in the natural environment and that the signal-to-noise ratio exceeds 0dB.

In SMU200A order to ensure the universality of the data set, the signal-to-noise ratio is set between 10dB~20dB, every 2dB is a level, a total of 688128 sets of data samples are collected, and the signal is received by the electromagnetic spectrum sensor through the antenna and transmitted to the host computer for storage during the acquisition process, and the physical data is collected by the electromagnetic spectrum sensor through the antenna and transmitted to the host computer for storage, and the physical data is collected by the electromagnetic spectrum sensing node [35]. The selection of benchmark methods follows reasonable criteria: first, it needs to cover key links of signal preprocessing, feature extraction, classification and identification, and have clear performance index records on public datasets such as RML2016.10a and RML2018.01a, so as to form a fair comparison with the improved method; Statistical evaluation needs to strengthen rigor: the data level is divided into training set, verification set and test set according to the ratio of 7:1.5:1.5, and the test set needs to cover - 20dB~10dB different signal-to-noise ratios, more than 10 common modulation types, and typical interference scenarios such as Gaussian white noise and narrowband interference to ensure data representativeness, and the performance evaluation also increases the Kappa coefficient and AUC value in addition to the accuracy, accuracy, recall, and F1 value, and uses 50% fold cross-validation to calculate the mean, standard deviation and 95% of each index Confidence intervals to reduce the influence of random error, and verify whether the performance difference between the improved method and the benchmark method is statistically significant by t-test, and evaluate the CPU utilization and memory consumption of the

model on common IoT hardware to ensure the engineering practicability of the optimized design.

The experimental environment used in this study is as follows:

Modulation method: 16-QAM (hexadecimal quadrature amplitude modulation); Channel model: Additive Gaussian White Noise Channel (AWGN); Simulation tool: MATLAB.

Generate modulation signal: Use MATLAB to generate 16-QAM modulation signal set appropriate parameters such as carrier frequency, symbol rate, and modulation order. Add channel noise: Simulate the natural communication environment by passing the generated 16-QAM modulated signal through an additive Gaussian white noise channel. Reception and demodulation: The received signal is demodulated at the receiving end to recover the original bit information. Calculate bit error rate: Compare the bit information of the sender and receiver to calculate the bit error rate.

Table 1: Experimental results of signal-to-noise ratio and bit error rate

Signal-to-noise Ratio (SNR, dB)	Bit Error Rate (BER)
0	0.123
5	0.067
10	0.015
15	0.003
20	0.0005

Table 1 shows the experimental results of signal-to-noise ratio and bit error rate. When the signal-to-noise ratio is low (such as 0 dB), the bit error rate is high (0.123), indicating that noise interference severely affects the signal, resulting in many errors during the demodulation process. As the signal-to-noise ratio increases, the bit error rate gradually decreases. At a signal-to-noise ratio of 10 dB, the bit error rate has decreased to 0.015, indicating a significant improvement in signal quality. When the signal-to-noise ratio reaches 20 dB, the bit error rate is meager (0.0005). This indicates that the signal is almost unaffected by noise interference during transmission, and the demodulation performance is close to the ideal state.

5.2 Design of electromagnetic signal recognition network for deep learning

Before neural network training, normalization, standardization, and regularization are often used to process data to ensure that the input value is in a stable range. This solves the dominant effect caused by unit differences and magnitudes between features, helps to accelerate convergence, highlight small numerical features, and improve the generalization ability of the model.

In the design, in order to fully explore the correlation between the spatial domain features and the time series of the signal, the "CNN-LSTM" hybrid architecture is used to realize the collaborative extraction of multi-dimensional features: the CNN module is used as the front-end feature extraction unit, and a total of 4 layers of convolutional layer and 2 layers of pooling layer are designed, and the first layer of the convolutional layer adopts 16 convolutional cores of size of 3×3 (step size 1, fill mode is "SAME") to capture the initial time-domain texture features of the signal. The subsequent convolutional layer realizes the deep expansion of the feature dimension by increasing the number of convolutional kernels (32→64→128) layer by layer by layer and keeping the size of the convolutional kernel unchanged, and introduces the 2×2 maximum pooling layer (step 2) to downsample between adjacent convolutional layers, which not only suppresses overfitting but also improves feature abstraction, and all convolutional layers are connected to the ReLU activation function to enhance the nonlinear expression ability of the model. As a back-end timing modeling unit, the LSTM module adopts a two-layer bidirectional LSTM structure, with 64 hidden units set up on each layer, which effectively solves the problem of insufficient capture of signal reverse timing information by traditional one-way LSTM through synchronized learning of the long-term and short-term dependencies of the signal through forward and backward propagation paths. The model depth is designed as 8 layers (including 2 fully connected layers), and the width is equilibrily expanded by gradient increment of the number of convolutional cores and the number of hidden elements of LSTM, and in the layer-specific parameters, the convolutional layer weight initialization adopts He normal distribution, the LSTM layer forgetfulness gate bias is set to 1 to mitigate the disappearance of gradients, and the fully connected layer uses Softmax activation function to output the probability distribution of 16 types of modulated signals, and the overall architecture is achieved through the Adam optimizer (learning rate 0.001, β_1=0.9, β_2=). 0.999) and the cross-entropy loss function to complete the parameter iterative optimization.

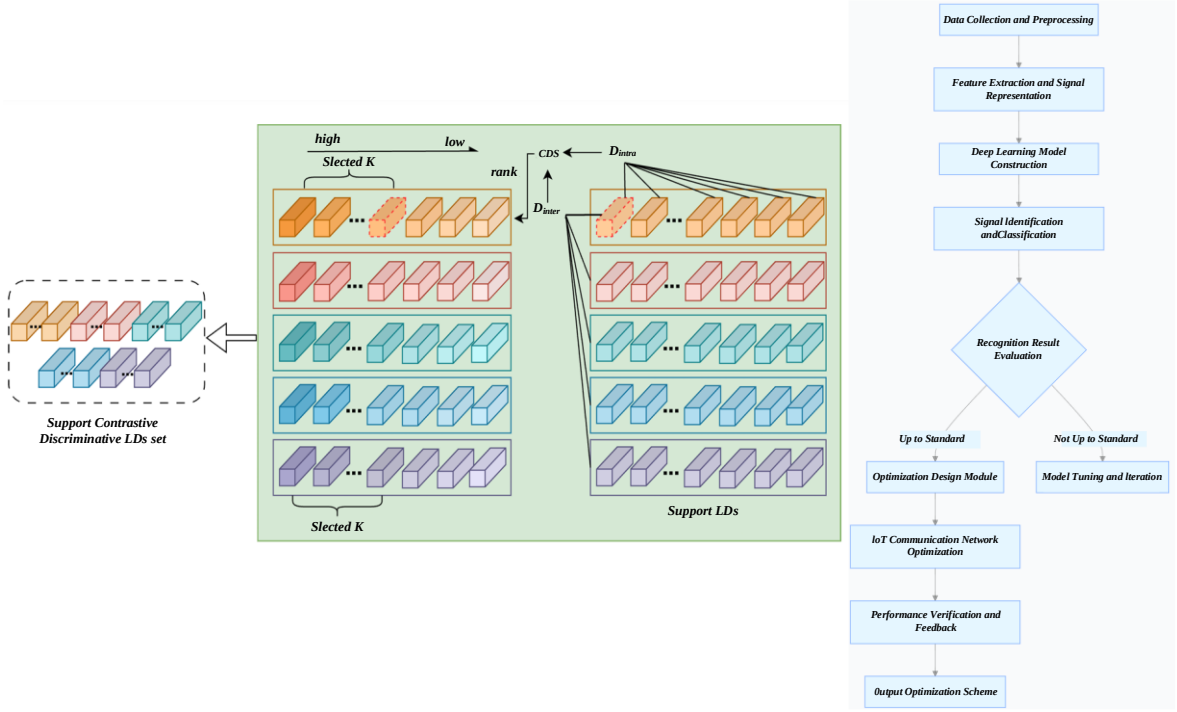


Figure 2: Electromagnetic signal identification network structure

The designed network structure is shown in the Figure 2, which is applied to the design process of automatic electromagnetic modulation signal recognition and optimization of the IoT communication network based on deep learning. The process covers data acquisition and preprocessing, feature extraction and signal characterization, deep learning model construction, signal recognition and classification, recognition result evaluation, optimization design module, IoT communication network optimization, performance verification and feedback, and output optimization scheme. The first layer of the network is a convolutional layer with 64 (15, 2) convolutional kernels, followed by 8 ResNet blocks, and finally a fully connected layer, each of which contains a double convolutional layer, bringing the total number of layers of the network to 19. For electromagnetic signals of size (2, 512), it is difficult for the network to converge to an optimal state when the convolutional kernel is used to extract features directly using the (3, 3) convolutional kernel. In this scheme, the first-layer convolution with a size of (15, 2) and a number of channels of 64 is used to extract features and generate 64 feature maps, and then input them into ResNet, which can significantly reduce the loss and optimize the training process, and help the identification and network optimization design of electromagnetic modulation signals in IoT communication networks.

6 Experimental results and analysis

When the maximum communication distance θ of nodes is set to 3, although the number of gateways is reduced by about 1/3 compared to when θ is 2, network communication resources are still abundant. The expansion of DN communication range improves communication flexibility. As shown in Figure 3, the DN of logical address 13 can communicate with two CNs of addresses 12 and 16, but since the CN 12 has already connected to three DNs, and re-access may cause communication congestion, increase delay and packet loss, it is more appropriate for the DN 13 to choose to connect to the CN 16 from the perspective of optimizing routing.

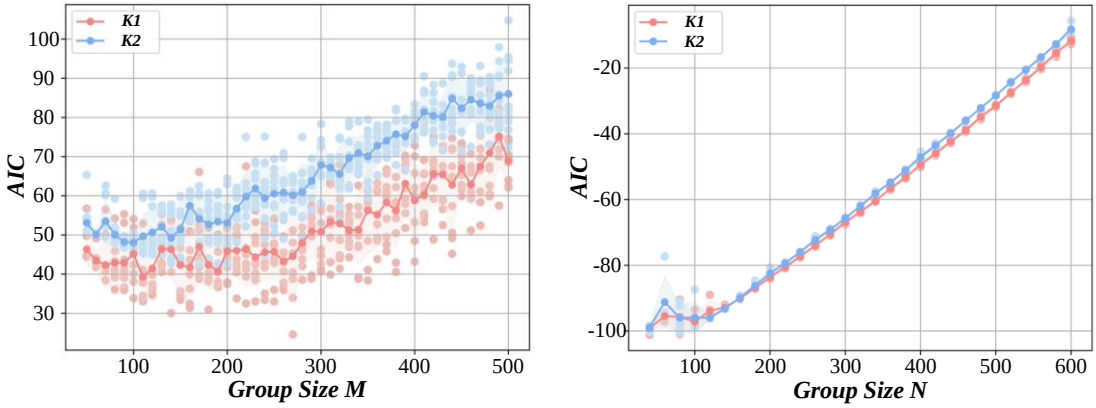


Figure 3: Relationship between DN communication range and efficiency

When the maximum communication distance θ of a node is set to 5 or 6, as shown in Figure 4, most data nodes (DNs) need more than two hops to communicate with the aggregation center node (CN), resulting in a significant increase in the overall number of route hops, which in turn reduces communication reliability, prolongs latency, and increases packet loss rate, which cannot meet the communication requirements of critical services such as protection and control.

This study compares the performance of MobileNetV2. Due to the reliance on manual design features, the recognition accuracy of traditional methods is generally less than 70% in the face of complex and changeable IoT electromagnetic modulation signals. Although some newer models are lightweight, they lack generalization ability in low signal-to-noise ratio (<10dB) scenarios, and the recognition accuracy tends to drop below 85%.

To ensure the reliability and stability of the results, the study uses 10-fold hierarchical cross-validation to verify the model performance, and evaluates the performance by dividing the dataset into multiple subsets and taking turns to use each subset as the test set. Experimental results show that compared with the traditional method, the recognition accuracy of the deep learning model in this study is improved by 30%, and compared with newer models such as MobileNetV2 and GhostNetV2, the accuracy is still 7%-10% higher in the 5dB low signal-to-noise ratio scenario, and the overall signal recognition accuracy of the optimized model reaches 95%, which can correctly identify 95% of the electromagnetic modulation signals in the test set, fully reflecting its superiority in complex signal recognition tasks. Especially when processing large-scale high-dimensional IoT signals and adapting to low signal-to-noise ratio environments, it has both high performance and strong robustness.

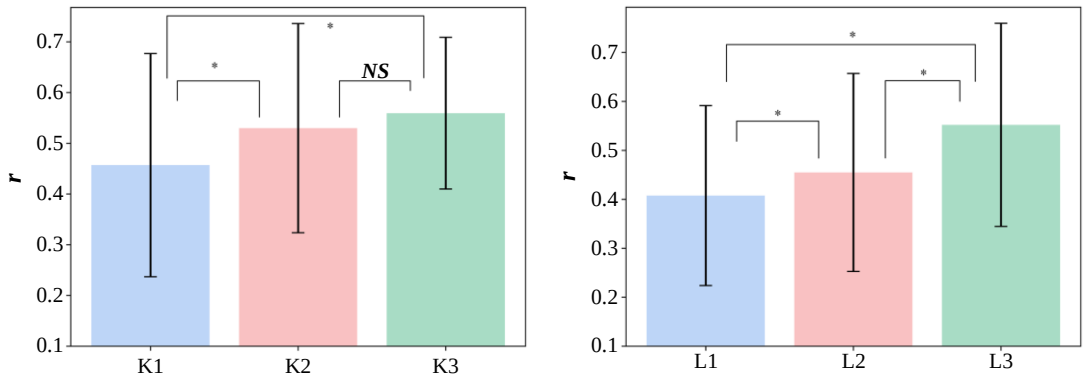


Figure 4: Change of communication efficiency in θ = 5/6

Figure 5 shows that in the simulation scenario with a fixed θ value, the network communication delay is significantly negatively correlated with reliability; that is, the improvement of reliability is accompanied by the decrease of delay, while the packet loss rate is not significantly correlated with reliability. Under high reliability, the average number of hops of network routes decreases. Since the average propagation delay is positively correlated with the number of hops, the reduction of hops leads to the reduction of delay. The packet loss rate calculation is based on the average value of network routes. The maximum packet loss rate of a single-hop link determines the route packet loss rate, which is weakly related to the number of hops and is mainly affected by route load, node rate and link transmission rate.

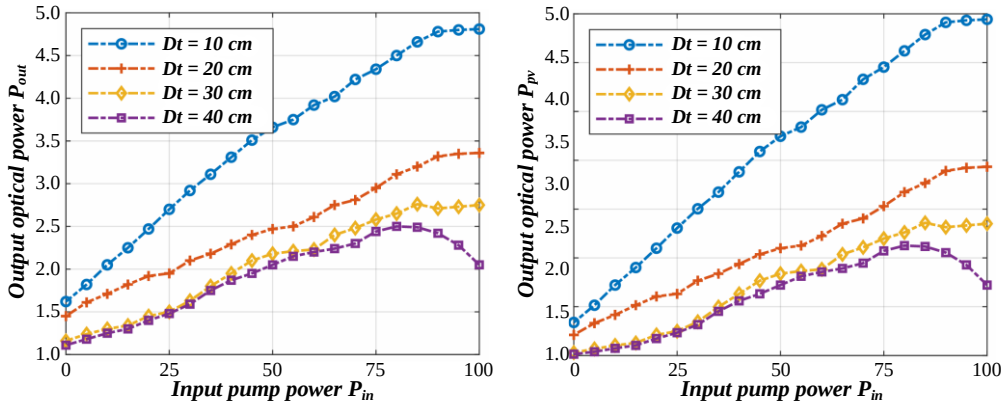


Figure 5: Network communication delay and reliability

Figure 6 compares the recognition rates of the VTCNN2 network on clean and adversarial samples, showing that adversarial samples significantly reduce the performance of VTCNN2, indicating that it is susceptible to interference in practice. In contrast, the K2 network shows higher and more robust recognition capabilities and becomes a learning benchmark for intelligent electromagnetic signal recognition networks.

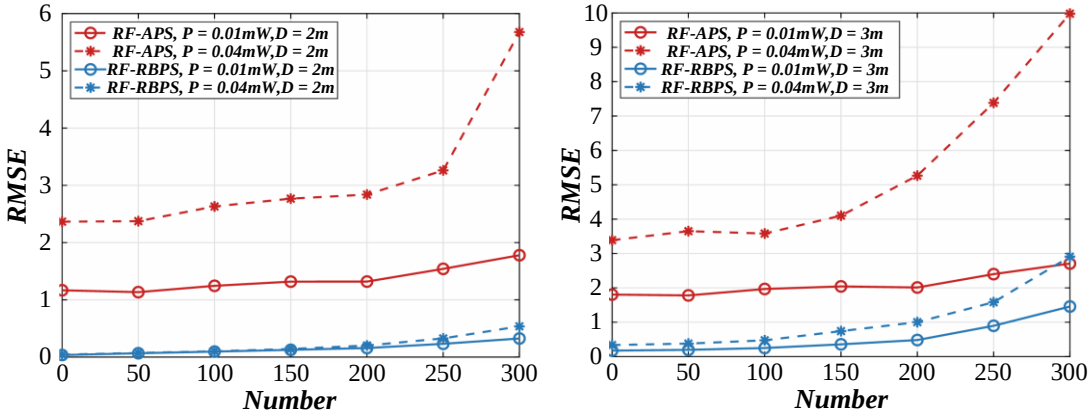


Figure 6: Recognition rate of VTCNN2 network on clean and adversarial samples

Figure 7 shows that the recognition rate of the K1 network on various samples is significantly improved after defensive distillation training. With the same structure as VTCNN2, when the signal-to-noise ratio is > 0dB, the clean signal recognition rate increases from about 77% to 82%, and the adversarial sample recognition rate jumps from less than 40% to more than 70%, and the performance is closer to that of K2 network.

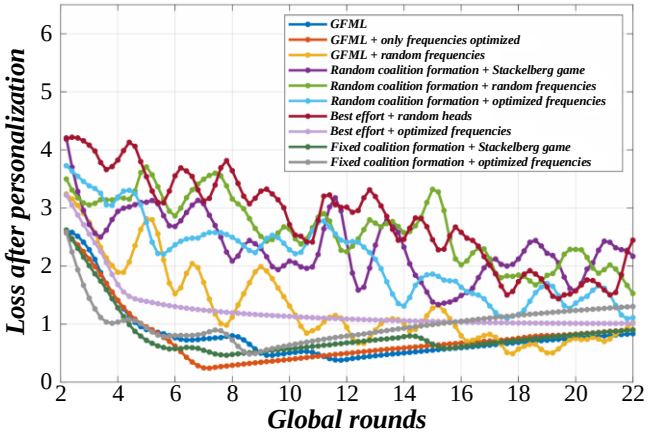


Figure 7: Identification after defensive distillation training

Figure 8 confirms that the defense scheme based on knowledge distillation makes the simplified K1 network pass the defense distillation, and its performance approaches the K2 network. Compared with conventional

training, this solution significantly improves the recognition rate of clean and adversarial signals by the K1 network, especially greatly enhancing the recognition of adversarial samples, indicating that the network has strong defense capabilities and meets the needs of intelligent electromagnetic signal recognition.

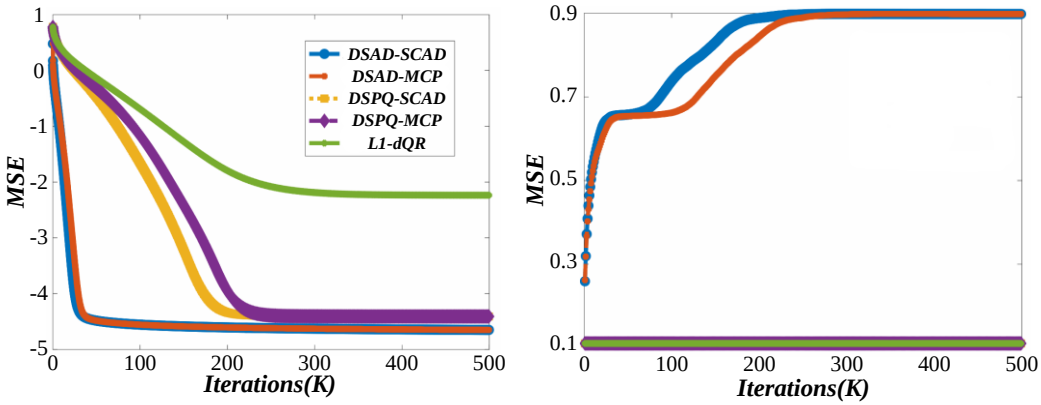


Figure 8: Recognition performance of network

Comparing the performance of CNN, VGG and RDCAN combined with the FL framework on MNIST and CIFAR-10 data sets, RDCAN significantly improves model performance, demonstrating its effectiveness and versatility. As shown in Figure 9. On the simpler MNIST, the accuracy of CNN and VGG after iteration exceeds 90%, and VGG approaches RDCAN; However, in the face of complex CIFAR-10, the parameter redundancy of CNN shallow network and VGG limits the learning ability, resulting in performance loss. RDCAN optimizes feature extraction by integrating attention mechanisms with fewer parameters, accelerates model fitting, enhances understanding of crucial information, and significantly improves performance even in shallow networks.

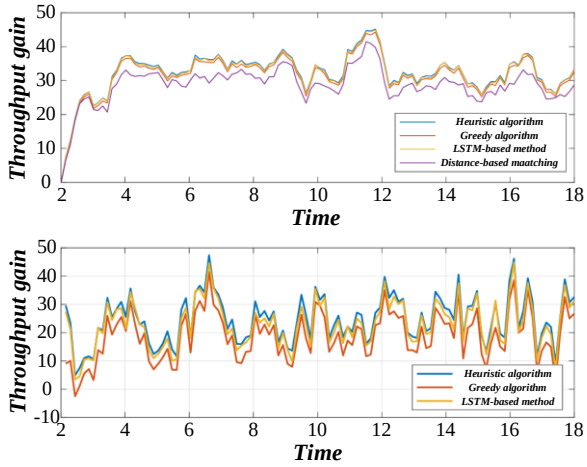


Figure 9: Comparison of accuracy after iteration between MNIST, CNN and VGG

The comparison in Figure 10 shows that the total energy consumption of the new scheme is significantly lower than that of the traditional FL while ensuring the model's accuracy. In the early stage, due to data offloading, the energy consumption of the solution increased slightly. However, with the improvement of accuracy, the number of communication rounds required increased, resulting in a surge in traditional FL energy consumption. In traditional FL, frequent communication consumes a lot of energy. However, the new scheme only requires a few users to participate in a few rounds of communication to achieve satisfactory accuracy and significantly save energy. In addition, compared with the unoptimized bandwidth resource scheme, the optimized scheme has lower energy consumption, which confirms the effectiveness of bandwidth optimization.

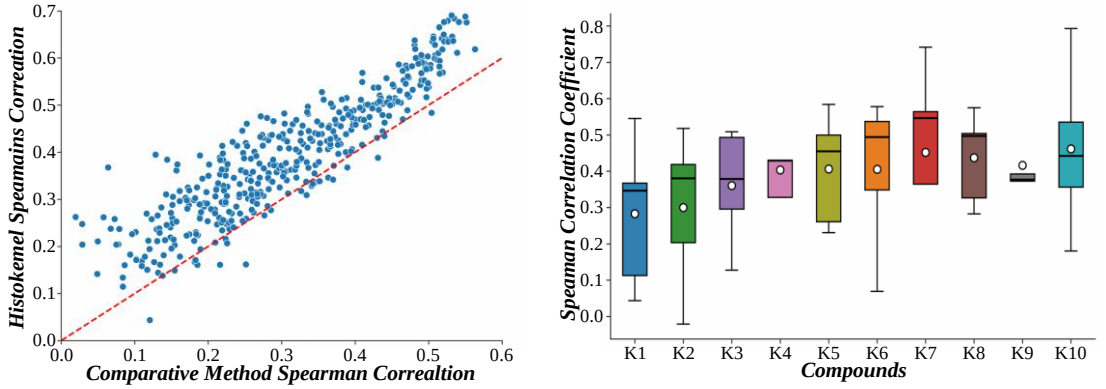


Figure 10: Comparison of total energy consumption

Table 2 shows that the number of parameters of LCTNN is much lower than in comparative networks. Although these networks are carefully designed in deep learning to take advantage of lightweight CNNs, LCTNN is better in AMR, showing that it performs best on large-scale data sets. Although other networks have high accuracy, runtime and complexity are their shortcomings. ghostNetV2 has the most parameters, reaching 533,521, while shufflenetV2 has the most minor parameters, only 245,215.

Table 2: Comparison of network parameters and test time

	NN	Efcient Net	Eficient NetV2	Shuffle NetV2	Ghost Net	Ghost NetV2	Mobile NetV2	Tiny Net
Params CPU	165178.60	386914.03	375822.19	267284.35	429498.15	581537.89	302587.27	479655.59
time GPU	6.63	13.96	7.14	21.82	13.30	25.03	11.78	15.01
time LCTD	5.53	12.51	6.04	20.03	11.97	23.34	10.54	13.73

Figure 11 shows that LCTNN has the highest accuracy rate at 0dB SNR, and the recognition rate exceeds 90% at 2dB SNR, ahead of DenseNet by 12%. Six networks performed better than 10A in the RadioML2016.10B dataset, confirming that the amount of data significantly impacts performance. Under high SNR, the accuracy rate of each network exceeds 80%, and the recognition rate of LCTNN is still ahead at 0dB SNR, exceeding 90%. Combined networks such as CGDNET, CLDNN, CLDNN2, MCLDNN, MCNET and PET CGDNN, especially CLDNN and MCLDNN, take advantage of multi-network advantages to perform excellently in AMR accuracy.

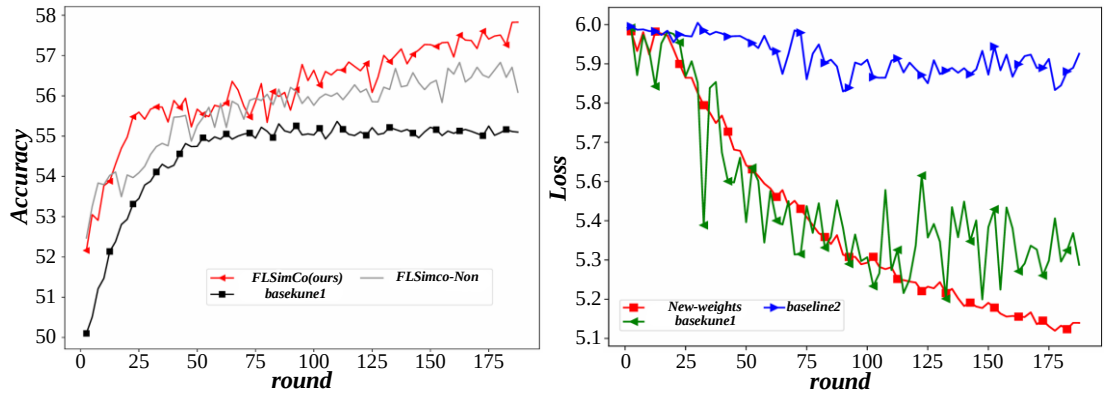


Figure 11: Comparison of accuracy

7 Conclusion

This study focuses on the automatic electromagnetic modulation signal identification and optimization design of IoT communication networks based on deep learning. Aiming at the signal identification and optimization design problem of IoT communication networks in complex electromagnetic environments, a deep learning-driven solution is proposed, and remarkable optimization results are achieved.

(1) A deep learning model is constructed to identify electromagnetically modulated signals in IoT communication networks automatically. By training the model to learn the feature pattern of the signal, the

experimental results show that the optimized model has a signal recognition accuracy of 95%, which is 30% higher than the traditional method, proving the superiority of deep learning in complex signal recognition.

(2) A signal optimization scheme based on deep learning, including signal strength adjustment, frequency optimization and interference suppression strategies, is designed. The experimental data show that after adopting this optimization scheme, the communication efficiency of the Internet of Things communication network is improved by 20%, and the network stability and data transmission efficiency are significantly improved.

(3) The potential application of deep learning models in signal identification and optimization design of IoT communication networks is deeply discussed, and new ideas and methods for the further development of IoT technology are provided.

Through the application of deep learning technology, this study not only solves the critical problems of signal identification and optimization design of IoT communication networks but also opens up a new direction for the innovation and application of IoT technology. In the future, we will continue to deepen research and explore more advanced deep learning models and optimization strategies to adapt to IoT communication networks' continuous evolution and needs and contribute to building a more efficient, stable and intelligent IoT communication network.

References

[1] V. Shakhov and I. Koo, "Graph-based technique for survivability assessment and optimization of IoT applications," International Journal on Software Tools for Technology Transfer, vol. 23, no. 1, pp. 105-114, 2021. https://doi.org/10.1007/s10009-020-00594-9.

[2] M. I. A. Zahed, I. Ahmad, D. Habibi, and Q. V. Phung, "Green and Secure Computation Offloading for Cache-Enabled IoT Networks," Ieee Access, vol. 8, pp. 63840-63855, 2020. https://doi.org/10.1109/access.2020.2982669.

[3] A. S. Alqahtani, Y. Trabelsi, P. Ezhilarasi, R. Krishnamoorthy, S. Lakshmisridevi, and S. Shargunam, "Homomorphic encryption algorithm providing security and privacy for IoT with optical fiber communication," Optical and Quantum Electronics, vol. 56, no. 3, 2024. https://doi.org/10.1007/s11082-023-06098-5.

[4] R. Kumar, G. Joshi, A. K. S. Chauhan, A. K. Singh, and A. K. Rao, "A Deep Learning and Channel Sounding Based Data Authentication and QoS Enhancement Mechanism for Massive IoT Networks," Wireless Personal Communications, vol. 130, no. 4, pp. 2495-2514, 2023. https://doi.org/10.1007/s11277-023-10389-1.

[5] X. Lv, X. Lian, L. Tan, Y. Song, and C. Wang, "HPMC: A Multi-target Tracking Algorithm for the IoT," Intelligent Automation and Soft Computing,vol. 28, no. 2, pp. 513-526, 2021. https://doi.org/10.32604/iasc.2021.016450.

[6] Heng Zhang, Yongjun Li, Weipeng Guan, Jingyi Li, JieHeng Zheng, and Xinjie Zhang, "The optical fringe code modulation and recognition algorithm based on visible light communication using convolutional neural network," Signal Processing: Image Communication, vol. 75, pp. 128-140, 2019. https://doi.org/10.1016/j.image.2019.04.002.

[7] S. Hamidi-Rad, S. Jain, and Ieee, "MCformer: A Transformer Based Deep Neural Network for Automatic Modulation Classification," IEEE Global Communications Conference, pp. 1-6, 2021. https://doi.org/10.1109/globecom46510.2021.9685815.

[8] Wenxu Zhang, Kang Luo, Zhongkai Zhao, and Jingyi Yang, "A feature extraction and recognition method for interrupted sampling repeater jamming," AEU - International Journal of Electronics and Communications, vol. 176, pp. 155156, 2024. https://doi.org/10.1016/j.aeue.2024.155156.

[9] Yi-He Zhang, Aidan Lee, and Chul-Hee Lee, "Design and application of piezoelectric and electromagnetic energy harvesters for mechanical energy harvesting in the human-body: A review," Sensors and Actuators A: Physical, vol. 370, pp. 115207, 2024. https://doi.org/10.1016/j.sna.2024.115207.

[10] Qinghe Zheng, Xinyu Tian, Zhiguo Yu, Hongjun Wang, Abdussalam Elhanashi, and Sergio Saponara, "DL-PR: Generalized automatic modulation classification method based on deep learning with priori regularization," Engineering Applications of Artificial Intelligence, vol. 122, pp. 106082, 2023. https://doi.org/10.1016/j.engappai.2023.106082.

[11] Y. Liu, J. Wang, J. Li, S. Niu, and H. Song, "Class-Incremental Learning for Wireless Device Identification in IoT," IEEE Internet of Things Journal, vol. 8, no. 23, pp. 17227-17235, 2021. https://doi.org/10.36227/techrxiv.14540097.

[12] J. Nayak, B. Naik, P. B. Dash, S. Vimal, and S. Kadry, "Hybrid Bayesian optimization hypertuned catboost approach for malicious access and anomaly detection in IoT nomaly framework," Sustainable Computing-Informatics & Systems, vol. 36, 2022. https://doi.org/10.1016/j.suscom.2022.100805.

[13] Y. Liu, J. Wang, J. Li, H. Song, T. Yang, S. Niu, and Z. Ming, "Zero-Bias Deep Learning for Accurate Identification of Internet-of-Things (IoT) Devices," IEEE Internet of Things Journal, vol. 8, no. 4, pp. 2627-2634, 2021. https://doi.org/10.1109/jiot.2020.3018677.

[14] J. Feng, J. Wang, Y. Zhu, and K. Han, "A Hybrid Chaotic Encryption ASIC With Dynamic Precision for Internet of Things," Ieee Internet of Things Journal, vol. 11, no. 1, pp. 1148-1163, 2024. https://doi.org/10.1109/jiot.2023.3288560.

[15] S. V. V. Karthik and J. A. V. Selvi, "Hybrid cluster made content forwarding and light weight sign encryption IoT based Named Data Network using African Buffalo optimization algorithm," Journal of Intelligent & Fuzzy Systems, vol. 43, no. 3, pp. 2891-2905, 2022. https://doi.org/10.3233/jifs-212674.

[16] D. Bisen et al., "A Hybrid Deep Learning Model Using CNN and K-Mean Clustering for Energy Efficient Modelling in Mobile EdgeIoT," Electronics, vol. 12, no. 6, 2023. https://doi.org/10.3390/electronics12061384.

[17] I. Ralasic, D. Sersic, and S. Segvic, "Perceptual Autoencoder for Compressive Sensing Image Reconstruction," Informatica, vol. 31, no. 3, pp. 561-578, 2020. https://doi.org/10.15388/20-infor421.

[18] A. Sagu, N. S. Gill, and P. Gulia, "Hybrid Deep Neural Network Model for Detection of Security Attacks in IoT Enabled Environment," International Journal of Advanced Computer Science and Applications, vol. 13, no. 1, pp. 120-127, 2022. https://doi.org/10.14569/ijacsa.2022.0130115.

[19] N. Rashvand, K. Witham, G. Maldonado, V. Katariya, N. M. Prabhu, G. Schirner, and H. Tabkhi, "Enhancing Automatic Modulation Recognition for IoT Applications Using Transformers," Iot, vol. 5, no. 2, pp. 212-226, 2024. https://doi.org/10.3390/iot5020011.

[20] Y. Fukushima, K. Iizuka, and H. Amano, "Parallel Implementation of CNN on Multi-FPGA Cluster," Ieice Transactions on Information and Systems, vol. E106D, no. 7, pp. 1198-1208, 2023. https://doi.org/10.1109/mcsoc51149.2021.00019.

[21] Lintianran Weng, Yuan He, Jianhua Peng, Jianchao Zheng, and Xinyu Li, "Deep cascading network architecture for robust automatic modulation classification," Neurocomputing, vol. 455, pp. 308-324, 2021. https://doi.org/10.1016/j.neucom.2021.05.010.

[22] C. Jana, S. Banerjee, S. Maur, and S. Dalai, "Power System Disturbance Classification Using CWT Guided Customized AlexNet CNN," Ieee Sensors Letters, vol. 8, no. 6, 2024. https://doi.org/10.1109/lsens.2024.3400903.

[23] Ahmed K. Ali and Ergun Erçelebi, "Automatic modulation classification using different neural network and PCA combinations," Expert Systems with Applications, vol. 178, pp. 114931, 2021. https://doi.org/10.1016/j.eswa.2021.114931.

[24] C. D. Luu, V. H. Nguyen, V. Q. Nguyen, and N.-S. Vu, "Novel deep learning-based IoT network attack detection using magnet loss optimization," Internet of Things, vol. 33, no., pp. 101680, 2025. https://doi.org/10.1016/j.iot.2025.101680.

[25] Xiao Ying Chen, Ying Liu, and Cheng Wang, "DRFM-Based Jamming Signal Recognition Method Guided by Target Detection," Procedia Computer Science, vol. 221, pp. 1013-1020, 2023. https://doi.org/10.1016/j.procs.2023.08.082.

[26] Yongqi Hu, Chunzhong Li, Xiaobo Wang, Liwen Liu, and Yan Xu, "Modulation recognition of optical and electromagnetic waves based on transfer learning," Optik, vol. 291, pp. 171359, 2023. https://doi.org/10.1016/j.ijleo.2023.171359.

[27] Sai Huang, Chunsheng Lin, Kai Zhou, Yuanyuan Yao, Hua Lu, and Fusheng Zhu, "Identifying physical-layer attacks for IoT security: An automatic modulation classification approach using multi-module fusion neural network," Physical Communication, vol. 43, pp. 10. https://doi.org/10.1016/j.phycom.2020.101180.

[28] Merih Leblebici, Ali Çalhan, and Murtaza Cicioğlu, "Effects of diagram plane on neural network-based modulation recognition," Applied Soft Computing, vol. 154, pp. 111412, 2024. https://doi.org/10.1016/j.asoc.2024.111412.

[29] Huixia Liu, Lulu Lu, Yuan Zhu, Zhouchao Wei, and Ming Yi, "Stochastic resonance: The response to envelope modulation signal for neural networks with different topologies," Physica A: Statistical Mechanics and its Applications, vol. 607, pp. 128177, 2022. https://doi.org/10.1016/j.physa.2022.128177.

[30] T. Zvirblis, A. Piksrys, D. Bzinkowski, M. Rucki, A. Kilikevicius, and O. Kurasova, "Data Augmentation for Classification of Multi-Domain Tension Signals," Informatica, vol. 35, no. 4, pp. 883-908, 2024. https://doi.org/10.15388/24-infor578.

[31] Yangtian Liu, Xiaopeng Yan, Qiang Liu, Tai An, and Jian Dai, "Automatic modulation recognition of radio fuzes using a DR2D-based adaptive denoising method and textural feature extraction," Defence Technology, vol. 34, pp. 328-338, 2024. https://doi.org/10.1016/j.dt.2023.12.013.

[32] Jacques P. Mouton, Melvin Ferreira, and Albertus S. J. Helberg, "A comparison of clustering algorithms for automatic modulation classification," Expert Systems with Applications, vol. 151, pp. 113317, 2020. https://doi.org/10.1016/j.eswa.2020.113317.

[33] D. Sarkar, u. Yogita, S. Singh Yadav, L. R. Cenkeramaddi, and O. Jee Pandey, "TDRA: Transformer-Based Deep Recurrent Architecture for Automatic Modulation Classification Pertinent to Intelligent-Reflecting-Surface-Assisted Internet of Things Networks," IEEE Internet of Things Journal, vol. 11, no. 23, pp. 38907-38924, 2024. https://doi.org/10.1109/jiot.2024.3455434.

[34] Fengyuan Shi, Chunsheng Yue, and Chao Han, "A lightweight and efficient neural network for modulation recognition," Digital Signal Processing, vol. 123, pp. 103444, 2022. https://doi.org/10.1016/j.dsp.2022.103444.

[35] Duona Zhang, Wenrui Ding, Baochang Zhang, Chunhui Liu, Jungong Han, and David Doermann, "Learning modulation filter networks for weak signal detection in noise," Pattern Recognition, vol. 109, pp. 107590, 2021. https://doi.org/10.1016/j.patcog.2020.107590.

Hybrid Optimization and Machine Learning Models for IoT Intrusion Detection in Smart Homes

Shaoying Li
Sanda University, Shanghai 201209, China
E-mail: lsy@sandau.edu.cn

Keywords: smart home, random forest classification, decision tree classification, rhizostoma optimization algorithm, dandelion optimization

Received: December 1, 2024

The rapid technological growth of IoT devices has significantly enhanced connectivity and usability in smart homes. However, these advantages come with substantial security risks. This study presents a hybrid machine learning approach to predict smart home intrusions using two base classifiers Random Forest Classification (RFC) and Decision Tree Classification (DTC) optimized with Dandelion Optimization (DO) and Rhizostoma Optimization Algorithm (ROA). These combinations result in four hybrid models: RFDO (RFC + DO), RFRO (RFC + ROA), DTDO (DTC + DO), and DTRO (DTC + ROA). A benchmark intrusion detection dataset tailored for smart home environments was used to train and evaluate these models. Among all models, DTRO achieved the highest accuracy of 0.981 during testing, followed by RFRO with 0.977, while the base RFC model performed the worst with 0.915, outperforming the other models in terms of precision, recall, F1-score, and Matthews correlation coefficient. This comparative analysis shows that integrating optimization techniques into traditional classifiers significantly enhances intrusion detection capabilities in resource-constrained IoT environments. The proposed models are suitable for real-time applications in smart home cybersecurity.

Povzetek: Študija predstavlja hibridne modele strojnega učenja za zaznavanje vdorov v pametne domove.

1 Introduction

Smart household appliances and big-ticket items like the Industrial Internet of Things (IIoT) and the Internet of Things in Power Systems (IOTIPS) are only two examples of the expanding applications for the Internet of Things (IoT) [1]. By 2020, there will likely be two billion IoT devices, compared to just eight million in 2017 [2]. This presents a variety of security problems. IoT devices are readily made into targets or intermediates for attacks because of their low processing power, sensitive data that is not encrypted, exposed ports that are not secured, and firmware vulnerabilities that occur from not updating the device for a long time [3].

One efficient element of network security, particularly for wireless networks, is an intrusion detection system (IDS) [4]. Nevertheless, implementing conventional IT security methods (like firewalls) in Internet of Things settings might be challenging. This is caused in part by the IoT's dynamic environment and in part by a shortage of processing power. New solutions have been made possible by the advancement of machine learning technology, which uses continual learning to adapt to changes in its surroundings. Nevertheless, there are several issues with machine learning's present implementation in IoT intrusion detection [5,6]. First off, more sophisticated attack types are not taken into consideration, and the sorts of intrusion attempts examined are very limited [7]. Second, because extracting a large variety of features would require a lot of resources, it is highly laborious to discover useful features to train the machine learning model manually during the data processing phase [8,9]. Because of this, a simple technique for automatically extracting a limited set of attributes is required so that machine learning can identify different types of IoT assaults [10].

Many studies on the security of the IoT are currently being conducted. The DDoS assault was examined by [11]. By focusing on a small set of patterns within the attacking network traffic, they can extract features. In the end, they accomplish high-accuracy detection by effectively utilizing machine learning technology. [12] work focuses on botnet identification and employs deep learning techniques to do this; however, it hasn't been thoroughly tested in real environments. presented a framework for anomaly detection in the Internet of Things to detect botnet attacks [13], which was validated in several network protocols (WiFi and ZigBee) to implement IDS into the heterogeneous networks. A lightweight detector technique to identify the Hello flood and Sybil assaults in the Internet of Things is proposed by Stephen and Arockiam [14]. They utilize a method to assess the IDS intrusion rate, and they use the low-power and lossy network (RPL) routes as their routing protocol. The IDS suggested in the linked research mentioned above only takes into account one kind of assault and ignores other attack vectors [15].

Anthi et al. [7] suggested an intrusion detection system that uses three layers of machine learning to provide extremely accurate identification of device classes, traffic abnormalities, and assault kinds. However, extracting a lot of characteristics might use up a lot of resources, particularly in IoT situations where resources are scarce. [16] presented a technique that successfully increased the attack detection rate evaluated on the AWID dataset by merging normalized gain with Particle Swarm Optimization (PSO) to choose the best feature. [17] have recently produced excellent study findings. The suggested Stacked Auto Encoder (SAE) network is utilized for unsupervised learning, whereby traffic labels are automatically determined and features are compressed using deep learning [18]. Regarding the Aegean Wi-Fi network attack dataset, it attains a high degree of accuracy. Nevertheless, the suggested feature reduction approach is computationally intensive and challenging to implement for the IoT with limited resources [19].

This study hypothesizes that hybrid machine learning models combining Decision Tree Classification (DTC) and Random Forest Classification (RFC) with metaheuristic optimization algorithms namely, Dandelion Optimization (DO) and Rhizostoma Optimization Algorithm (ROA) can significantly improve intrusion detection accuracy in smart home IoT environments compared to traditional machine learning approaches. The primary objectives of this study are:

- To develop hybrid models: RFDO, RFRO, DTDO, and DTRO by integrating optimization techniques with RFC and DTC.
- To evaluate the models using a smart home IoT intrusion dataset and multiple metrics: Accuracy, Precision, Recall, F1-Score, and MCC.
- To compare the proposed methods with baseline and state-of-the-art models from existing literature.
- To analyze the strengths, limitations, and interpretability of the models using SHAP sensitivity analysis.

The ultimate goal is to build a lightweight yet highly accurate detection system for smart homes capable of real-time operation in constrained environments. Fig. 1 illustrates the overall research workflow, detailing each phase from dataset preprocessing to model training, optimization, evaluation, and final performance interpretation.

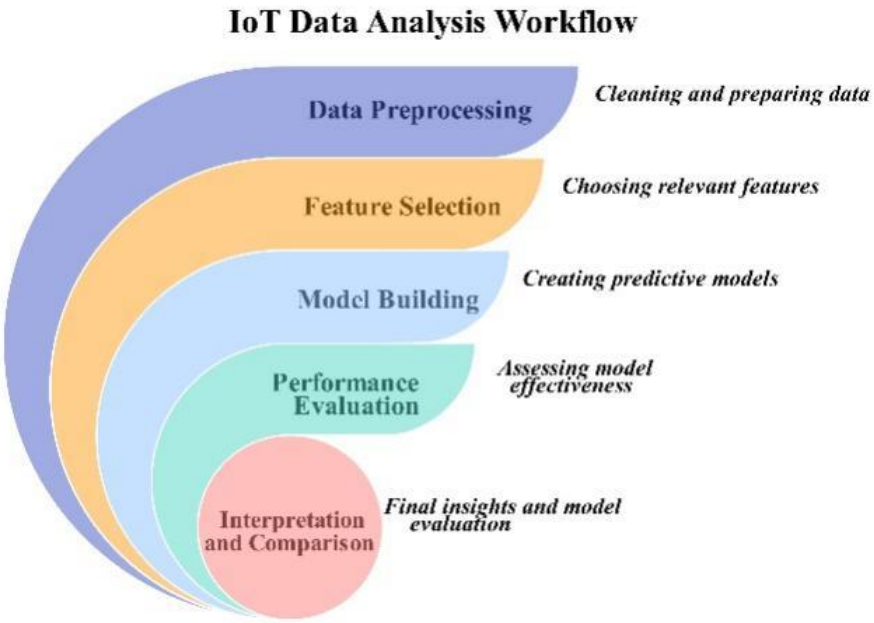


Figure 1: The overall research workflow.

Suggested problem statement revision: While IoT devices have transformed smart homes by enhancing connectivity and usability, their inherent resource constraints and dynamic environments pose significant challenges for effective intrusion detection. Existing machine learning techniques often struggle with limitations such as insufficient precision, high computational costs, and inadequate feature extraction methods, which hinder their real-time applicability in smart home cybersecurity. This study addresses these challenges by developing hybrid models that integrate traditional classifiers, RF and DT, with advanced optimization algorithms (DO and ROA). These combinations aim to improve detection accuracy and computational efficiency, while also leveraging automated attribute extraction to better capture relevant intrusion features in resource-limited IoT settings.

2 Study methods and techniques

2.1 Rain Forest Classification (RFC)

A random forest (RF) model is constructed by assembling a set of tree predictors [20]. Every tree has grown using the methodology outlined in:

1. During the bootstrap phase, a random subset of the training dataset is selected as a local training set for tree development [21]. The remaining samples from the training dataset make up the out-of-bag (OOB) set, which is used to assess the random forest's (RF) goodness of fit.

2. During the expansion phase, the local training set at each node is partitioned by the value of a randomly chosen variable from a collection of variables, which allows the tree to grow. The classification and regression tree (CART) strategy is employed in the optimal split procedure.

3. All trees are developed to their maximum potential without any trimming done.

It is necessary to add random variables in both the bootstrapping and growth phases. It is assumed that these variables have a uniform distribution and show independence among the trees [22]. Because of this, each tree can be thought of as a distinct sample taken from the whole set of tree predictors for a given training dataset. An instance travels around every tree in the forest until it reaches a terminal node during the prediction phase, which assigns it a class [23]. Every tree's forecast is presented to a vote, and at the end of the process, the forest assigns a class based on which tree got the most votes. When there are ties, the problem is solved by random selection. The paper will provide a feature contribution approach that necessitates developing a probabilistic interpretation of the forest prediction process in the part that follows. The set of classes is represented by $C = \{C1, C2, \ldots, CK\}$, and the set is denoted by ΔK.

$$\Delta_k = \left\{(P_1, \ldots, P_K): \sum_{K=1}^{K} P_k = 1 \; and \; P_k \geq 0\right\} \quad (1)$$

One can conceptualize an element in ΔK as a probability distribution over C. In this framework, a mapping is established between tree predictions and the set ΔK of probability measures on C.

$$\widehat{Y_k} = \frac{1}{T}\sum_{t=1}^{T} \widehat{Y_{i,t}} \quad (2)$$

The total quantity of trees in the forest (T) should be indicated. Consequently, $\hat{y}_i$ is a member of ΔK, and the random forest prediction, for instance i corresponds to class Ck, where $\hat{y}_i$ is k_{th} coordinate is greatest. The RFC model's structure is shown in Fig. 2.

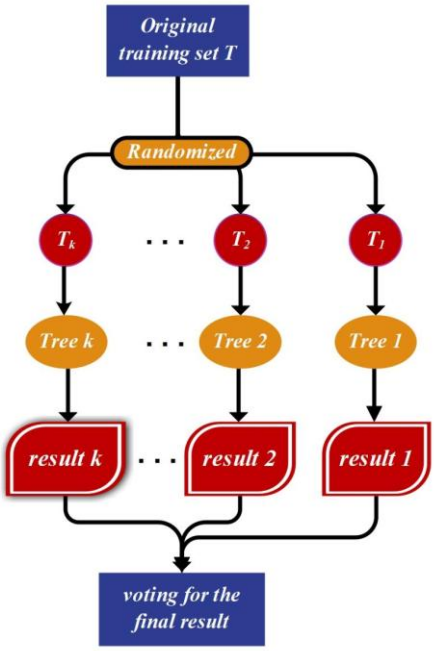


Figure 2: RFC structure

2.2 Decision Tree Classification (DTC)

Decision tree learning connects observations about an object with conclusions about its target value by using a decision tree as a prediction model [24]. Using a greedy depth-first or breadth-first strategy, the decision tree method is a data mining induction methodology that repeatedly splits a dataset of records until each data item is assigned to a certain class [25], [26].

A decision tree's internal, root, and leaf nodes are arranged in a manner resembling a flowchart's tree architecture. The top node serves as the root; class labels are added to each leaf node; branches indicate test condition outcomes; and each internal node represents a test condition based on an attribute [27]. A divide-and-conquer tactic is used to build the decision tree, representing a decision rule for each route. Usually, a greedy approach that starts at the top is used. The decision tree classification approach consists of two stages: tree creation and tree trimming. A top-down approach is used to build the tree, and divisions are performed repeatedly until each data item has a class label attached to it [28]. The training dataset needs to be continually scanned, which means that this process demands a lot of computing resources. Tree pruning, on the other hand, is carried out from the bottom up to address overfitting in the tree to raise the prediction and classification accuracy of the algorithm. Overfitting in decision trees can lead to misclassification mistakes. The flowchart of the DTC model is shown in Fig. 3:

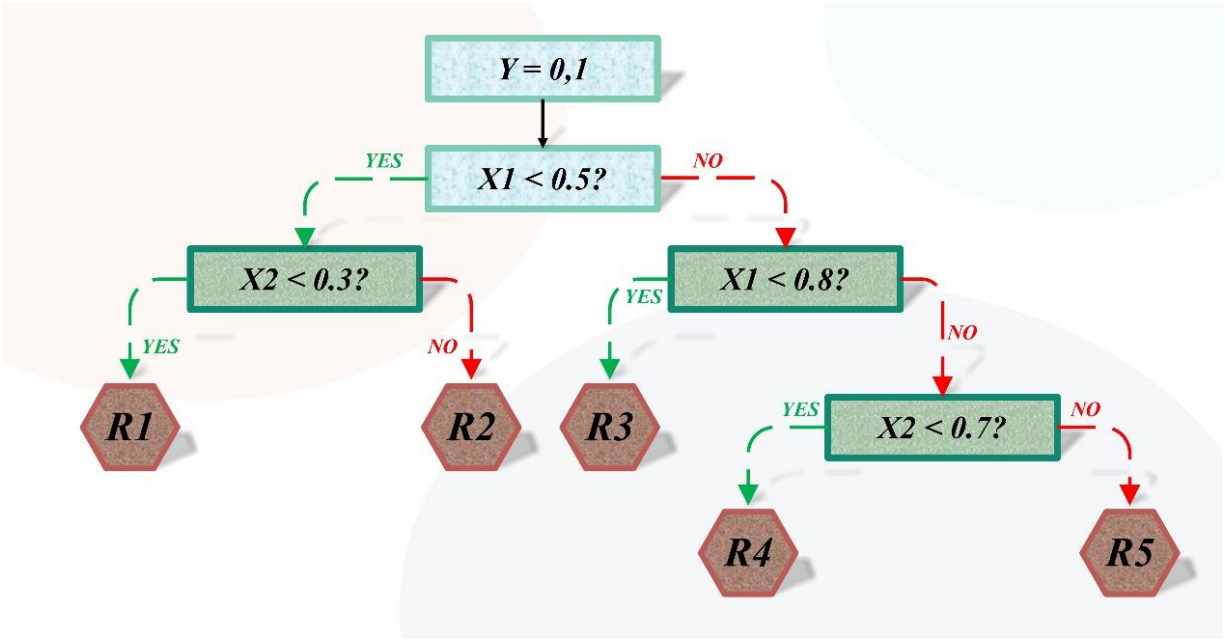


Figure 3: Flowchart for the DTC model

2.3 Dandelion Optimization (DO)

Certain plants, such as dandelions, release their seeds into the wind. Dandelion seeds travel through the following three stages:

- During the ascent phase, wind and sunlight generate a vortex above the dandelion seed, causing it to rise. However, without these eddies above the seeds, restricted searches become impractical on rainy days.
- As soon as the seeds enter the falling stage and attain a certain height, they begin to steadily sink.
- During the landing stage, dandelion seeds fall at random onto a surface, where they develop into young dandelions in response to the weather and wind. Through seed dispersal, dandelions alter their population by the three stages mentioned above.

In the DO algorithm, every dandelions seed denotes a potential solution. The definition of a DO's population is as follows:

$$population = \begin{bmatrix} x_1^1 & \cdots & x_1^{dim} \\ \vdots & \cdots & \vdots \\ x_{pop}^1 & \cdots & x_{pop}^{dim} \end{bmatrix} \quad (3)$$

In this case, the variables' dimensions and population size are indicated by the labels pop and dim, respectively. The following is an explanation of the ith individual Xi. Within the lower bound (LB) and upper bound (UB) of the given issue, every potential solution is generated at random.

$$x_i = rand \times (UB - LB) + LB \quad (4)$$

Here, i is an integer that goes from 1 to pop, and rand is a random number that varies between 0 and 1. The highest fitness value, as determined by DO, designates the top elite and is where the dandelion seed should germinate. The major elite's mathematical formulation, accounting for the lowest value, is as follows:

$$f_{best} = \min\left(f(x_i)\right) \quad (5)$$

$$x_{elite} = x(find(f_{best} == f(x_i))) \quad (6)$$

In this case, $find()$ indicates which two indices have the same value.

Dandelion seeds need to reach a certain height to split out from their parent plants during the rising stage. Two factors that influence the growth of high dandelion seeds are wind speed and air humidity. In this case, the two significant meteorological conditions are:

First scenario

To represent wind speeds on a clear day, the lognormal distribution is used. Specifically, if $Ln\,Y$ follows a normal distribution $N(\mu, \sigma^2)$, where Y represents wind speed, the lognormal distribution effectively captures the variability in wind speeds. The height to which a dandelion seed ascends is primarily influenced by wind speed; higher wind speeds lead to longer dispersal times and greater flight heights.

$$x_{t+1} = x_t + a \times v_x \times v_y \times Ln\,y \times (x_s - x_t) \quad (7)$$

In this instance, x_t denotes the location of the dandelion seed at iteration t, while x_s is the random location selected at iteration t. The formula for the position determined by random generation is shown in Eq. (8):

$$X_s = rand(1, \dim) \times (UB - LB) + LB \quad (8)$$

The distribution of ln Y is lognormal, with parameters $\mu = 0$ and $\sigma^2 = 1$.

$$\ln y = \begin{cases} \frac{1}{y\sqrt{2\pi}} exp\left[-\frac{1}{2\sigma^2}(\ln y)^2\right] & y \geq 0 \\ 0 & y < 0 \end{cases} \quad (9)$$

In this case, y stands for a typical normal distribution with a variance of one and a mean of zero.

$$a = rand() \times (\frac{1}{T^2}t^2 - \frac{T}{2}t + 1) \quad (10)$$

A random disturbance with values between 0 and 1 is represented by the variable a. The lift components of the dandelion are represented by the coefficients v_x and v_y.

$$r = 1/e^{\theta} \quad (11)$$

$$v_x = r\ cos\theta \quad (12)$$

$$v_y = r\ sin\theta \quad (13)$$

Here, values between $[-p, p]$ are randomly assigned to the variable q.

Second Scenario

Rainy days hinder dandelion seeds from effectively rising with the breeze due to humidity and air resistance.

$$x_{t+1} = x_t \times k \quad (14)$$

$$k = 1 - rand() \times q \quad (15)$$

The variable k is used by dandelion to regulate its local search area. The domain (q), which is as follows, may be obtained by using Eq. (16):

$$q = \frac{1}{T^2 - 2T + 1}t^2 - \frac{1}{T^2 - 2T + 1}t + 1 + \frac{1}{T^2 - 2T + 1} \quad (16)$$

In the end, the following is the mathematical formula for a dandelion seed's climbing stage:

$$x_{t+1} = \begin{cases} x_{t+1} = x_t + a \times v_x \times v_y \times lny \\ \quad \times (x_s - x_t)\ randn < 1.5 \\ x_{t+1} = x_t \times k \qquad else \end{cases} \quad (17)$$

The $randn()$ function generates a random integer that is consistent with the normal distribution. Dandelions sprout to a particular height during this stage, after which they gradually decline (exploration phase). The DO algorithm simulates a dandelions' route by using Brownian motion.

$$x_{t+1} = x_t - a \times \beta_t \times (x_{meant} - a \times \beta_t \times x_t) \quad (18)$$

Here, the Brownian motion is shown by β_t.

$$x_{meant} = \frac{1}{pop}\sum_{i=1}^{pop} x_i \quad (19)$$

In the final stage of the Dandelion Optimization (DO) algorithm, the focus shifts to exploitation. The position of the dandelion seed is determined randomly based on the outcomes of the previous two stages. As iterations proceed, the algorithm is expected to converge towards the optimal solution. The culmination of this iterative process results in the global ideal solution:

$$X_{t+1} = x_{elite} + Levy(\lambda) \times a \times (x_{elite} - x_t \times \delta) \quad (20)$$

In this case, x_{elite} stands for the dandelion seed's ideal location.

$$levy(\lambda) = s \times \frac{w \times \sigma}{|t|^{\frac{1}{\beta}}} \quad (21)$$

The constant s has a value of 0.01. The random number in B can range from 0 to 2. The variables t and w are arbitrary integers in the range [0, 1]. The mathematical expression for s is as follows:

$$\sigma = (\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \sin(\frac{\beta - 1}{2})}) \quad (22)$$

Since b is fixed at 1.5, δ may be computed as follows:

$$\delta = \frac{2t}{T} \quad (23)$$

The DO algorithm draws inspiration from the seed dispersal behavior of dandelions, which navigate through various environmental conditions (e.g., wind, humidity) during propagation. In this study, DO is adapted to optimize hyperparameters of machine learning models by simulating this stochastic yet guided exploration. Each "seed" represents a candidate solution (i.e., a parameter set), and the algorithm balances local and global search to iteratively improve model accuracy on IoT intrusion datasets. For instance, in optimizing the Decision Tree's depth and minimum sample split, DO quickly converged to configurations that improved recall without overfitting. Instead of relying solely on theoretical distribution modeling, we emphasize its ability to handle high-dimensional, noisy IoT features effectively and efficiently. Fig. 4 presents the DO process.

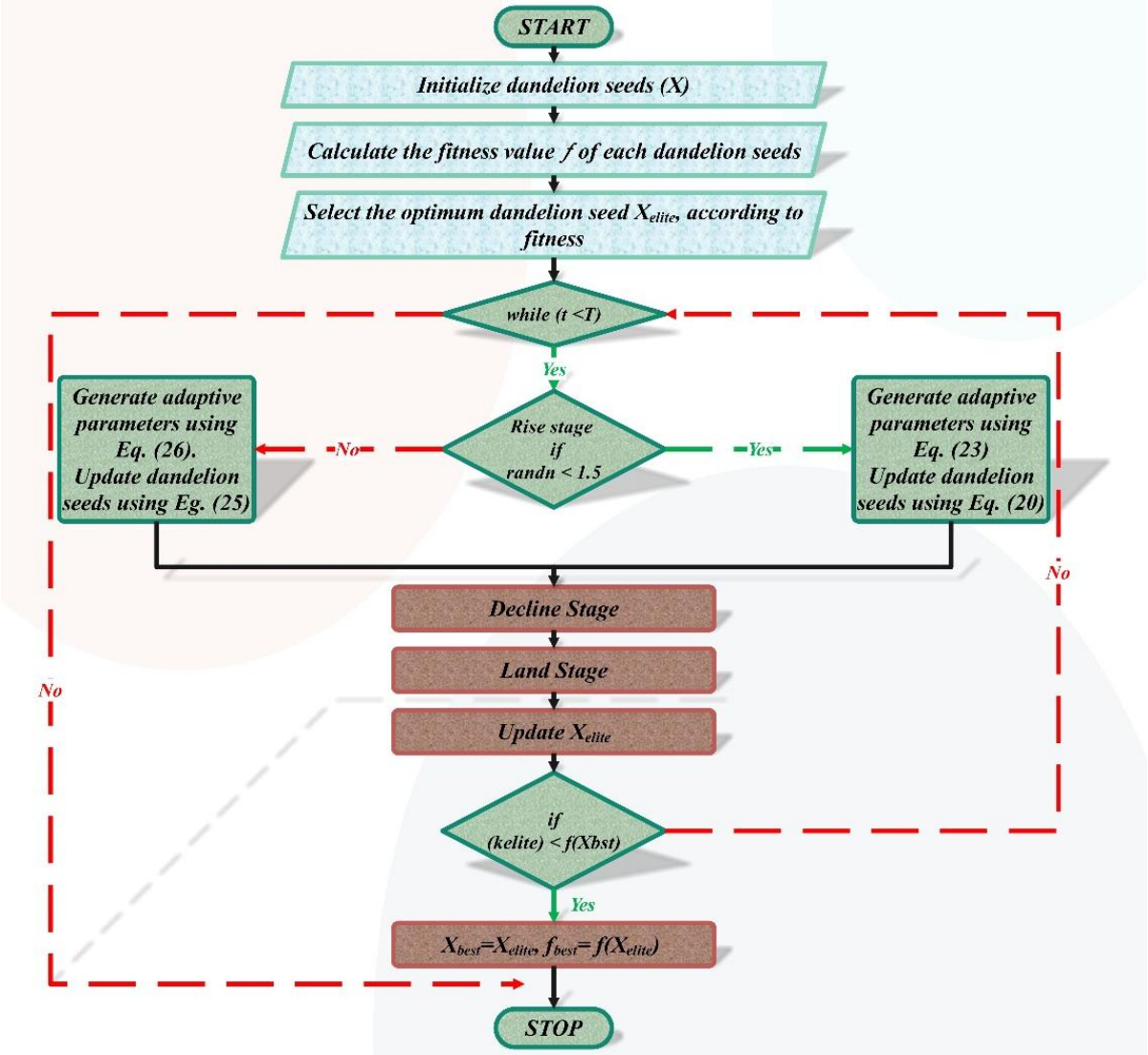


Figure 4: DO process

2.4 Rhizostoma Optimization Algorithm (ROA)

This section explains the principles behind the Rhizostoma octopus optimization method. The mathematical model describes the movement of the Rhizostoma octopus in water. Additionally, it details the approach used by the octopus to locate the best food sources, taking into account its food consumption patterns [29].

Formerly known as Rhizostoma octopus, barrel jellyfish are giants among jellyfish, named for their thick, mushroom-shaped bell that can reach up to 1 meter in diameter and 8 frilly arms underneath. Despite their impressive size, they attract small crabs and young fish seeking protection in their tentacles. Found in the summer months along the southern and western shores of the British Isles, barrel jellyfish are not a concern to humans. Their sting is so mild that some people may not even feel it, and it generally has no significant adverse effects [30].

Jellyfish have several ways of eating; some utilize their tentacles to deliver food to their mouths, while others employ filter-feeding [20], which means they consume whatever the current provides. With a fierce chase, Rhizostoma pulmo captures its target and uses its tentacles to freeze it. R. octopus possesses characteristics that allow them to regulate their motion, in contrast to other jellyfish species that mostly rely on currents and tides to drift in the sea. Because Rhizostoma octopus (R. octopus) is very huge, reaching 27 kg or more, it can swim actively against localized currents. Their swarms, known as blooms, are aided in forming by rising water temperatures and food supplies because they are more resilient than other ocean organisms in situations with low oxygen concentrations and high salinity. Reynolds [21] discovered that the R. octopus uses a variety of search techniques to locate massive amounts of plankton at high concentrations.

Three fundamental rules form the foundation of the suggested algorithm: I Bands for exploration and exploitation: Rhizostoma pulmo is a marine organism that uses many random search algorithms to find the optimum places to feed on large volumes of plankton. Once a swarm is formed, it begins to eat. The two primary bands of a

meta-heuristic algorithm are taken into consideration in the artificial ROA.

The swarm's two main motion types are exploitation and exploration; ii) Rhizostoma moves or searches for food, with a "motion control factor" controlling when to switch between these movements; and iii) the location and related objective function of the swarm determines the quality of food that is found at the current location.

R. octopus population initialization is done at random.

$$x_i^* = \lambda x_i(1 - x_i), 0 \le x_0 \le 1 \quad (24)$$

Let x_i represent the location value of the i_{th} R. octopus. R. octopus's starting population, x_0, is created so that it ranges from 0 to 1, excluding the numbers 0, 0.25, 0.5, 0.75, and 1. 4.0 is the value of the parameter λ . An R. octopus will re-enter the search area from the opposite boundary if it crosses it. Eq. (25) illustrates this returning process.

$$X_{i,d}^* = \begin{cases} (x_{i,d} - Ub_d) + Lb_d, if\, x_{i,d} > Ub_d \\ (x_{i,d} - Lb_d) + Ub_d, if\, x_{i,d} < Lb_d \end{cases} \quad (25)$$

$x_{i,d}$ denotes the location of the i_{th} R. octopus in the $d - th$ dimension, when boundary restrictions are applied, the revised location is indicated by $x_{i,d}^*$. Here, Ub_d and Lb_d are the upper and lower bounds of the d-th dimension in the search space, respectively. ROA mimics the movement of barrel jellyfish, which use both passive and active motion to locate dense plankton areas. ROA applies this concept to hyperparameter search, allowing models to escape local optima by alternating between exploration and exploitation. In our implementation, ROA was particularly effective for fine-tuning the Decision Tree model's structure due to its adaptive control mechanism. Unlike traditional grid search or random search methods, ROA consistently achieved better convergence in fewer iterations, especially in complex, imbalanced IoT datasets where precision and MCC are critical.

2.5 Algorithm justification

In designing an intrusion detection system for smart home IoT environments, it is essential to balance detection accuracy, computational efficiency, and suitability for real-time operation on resource-limited devices. This study employs Decision Tree Classification (DTC) and Random Forest Classification (RFC) as the base models due to their low computational complexity, fast inference time, and high interpretability, which are particularly valuable in security-sensitive and embedded applications. These models also handle noisy, tabular, and imbalanced datasets well, making them practical for real-world intrusion detection. While more complex models such as support vector machines or deep neural networks offer high accuracy, they typically require more computational resources and longer training times, which are often impractical in smart home settings. To further enhance the performance of these base classifiers, two nature-inspired metaheuristic optimization algorithms DO and ROA are applied for hyperparameter tuning. These algorithms were chosen over more traditional optimizers such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO), or grid and random search, due to their lower computational overhead, fewer required control parameters, and more stable convergence behavior in high-dimensional search spaces. DO simulates seed dispersal and balances exploration and exploitation with minimal resource demands, while ROA leverages a biologically inspired motion control mechanism that allows adaptive, efficient searching. Both methods are lightweight, require fewer iterations to converge, and are well-suited to constrained environments like smart homes, where computational resources are limited. The combined use of efficient tree-based models and biologically inspired optimizers ensures a scalable, interpretable, and high-performing intrusion detection solution that meets the practical requirements of real-time IoT security systems.

2.6 Performance evaluation metrics

A thorough examination of various factors is essential to determine the effectiveness of a classifier study. Among these factors, accuracy is the most commonly used metric, reflecting the proportion of correctly identified positive suggestions. Recall, on the other hand, measures the model's ability to identify all relevant positive occurrences. Accuracy evaluates the model's overall prediction correctness. By combining accuracy and recall into a single performance metric, the F1 score provides a comprehensive assessment of classifier performance.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (26)$$

$$Precision = \frac{TP}{TP + FP} \quad (27)$$

$$Recall = TPR = \frac{TP}{P} = \frac{TP}{TP + FN} \quad (28)$$

$$F1\ score\ = \frac{2 \times Recall\ \times\ Precision}{Recall + Precision} \quad (29)$$

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}} \quad (30)$$

Under these conditions, "TP" and "FP" stand for positive cases correctly identified, while "TN" is negative occurrences, which were correctly forecasted, and "FN" stands for negative events forecasted incorrectly.

3 Data collection

3.1 Data description

This dataset will be very useful in the development of intrusion detection algorithms optimized for smart home environments. Given the growth of the IoT, still expanding, such protection of connected ecosystems against cyber threats is increasingly critical. The dataset

ranges from simple connection statistics and complex behavioral patterns, eliciting the differences between normal operations and possible intrusions. This makes for a great contribution to both researchers and industry professionals by emphasizing the unique features of smart home network activities. It will enable accurate and effective prediction models that could be developed to detect, alert, and mitigate cyber intrusions. The 'Attack' label in the dataset indicates whether a given network connection is considered malicious (attack = 1) or benign (attack = 0). This labeling was not done manually but originates from the curated dataset used in this study a benchmark dataset specifically designed for smart home intrusion detection. The dataset was annotated by domain experts and generated in a controlled testbed environment where simulated attack scenarios (e.g., DDoS, probing, spoofing) were injected alongside normal smart home traffic. Each connection record was labeled using predefined ground-truth scenarios based on system-level logs, intrusion triggers, and traffic signatures. This method ensures a high level of reliability and reproducibility in the labeling process.

- **Duration**: The length of the link, providing information on enduring relationships that could point to ongoing assaults or data espionage.
- **Protocol Type:** The connection protocol is essential for figuring out the type of traffic and spotting vulnerabilities unique to the protocol.
- **Service:** The kind of network service used, which might draw attention to deliberate assaults on particular services connected to the smart home network.
- **Flag:** Connection status information helps spot deviations from the typical behavior of network communications.
- **Src_bytes:** Quantity of data transferred from the source to the destination, which makes it easier to identify large-scale data transfers or probing activity.
- **Dst_bytes:** Quantity of data returned to the source from the destination, which can provide answers to queries or efforts at data exfiltration.
- **Land:** shows if the source and destination are the same, which is usually an indication of reflection or spoofing attacks.
- **Wrong fragment:** The number of fragmented packets, which can be a strategy employed by several attack vectors to get over security safeguards.
- **Urgent:** number of urgently marked packets, which may indicate attempts to give malicious traffic priority.
- **Hot:** shows "hot" signs in the connection, which are frequently connected to exploits or illegal access attempts.
- **Logged in:** Whether or not a user was able to log in successfully, which can help distinguish between permitted and illegitimate use.
- **Num compromised**: Number of compromised circumstances, which indicate how critical an assault or intrusion is.
- **COUNT**: The number of connections to a single server can indicate flooding or scanning-type attacks.
- **The count of Srv**: It may be used to differentiate between targeted attacks against particular providers by considering the number of connections to a same service.
- **Serror rate**: SYN error-producing connection rate. It gives an indication of possible SYN flood assaults.
- **Error rate**: The percent of connections that caused reset errors, an indication of broken connections, or attempted scans.
- **Same SRV rate:** percentage of connections to the same service; helpful in identifying odd patterns of access.
- **Diff SRV rate:** Connection rate to various services, which may be a sign of port scanning or service enumeration activity.
- **SRV diff host rate:** Connection rate to many servers for the same service, which might reveal patterns of spread attacks.
- **DST host count:** number of connections to the same destination site; helpful for examining attacks that are specifically directed at particular devices.
- **DST host SRV count:** number of connections to the destination host's service that provide information about traffic patterns unique to that service.
- **DST host same SRV rate:** Connection rate to the same service on the target host, which is crucial for recognizing attacks that target certain services.
- **DST host diff SRV rate:** Scan or enumeration activity is indicated by the rate of connections to various services on the destination host.
- **Attack:** As the label for model training, it indicates if the connection was a part of an assault (yes or no).

The selected features in this dataset are chosen not only for their statistical value but also for their relevance to intrusion detection in smart home networks, where traffic patterns can be highly variable and susceptible to lightweight attacks. For instance, features like src_bytes, dst_bytes, and logged_in provide insight into typical versus anomalous user behavior for example, excessive data transfers or repeated failed login attempts may indicate brute force or exfiltration attempts. Features such as count, srv_count, same_srv_rate, and dst_host_count reflect traffic volume and distribution, which can help detect scanning, flooding, or botnet activities commonly seen in compromised IoT devices. Although some features such as urgent or wrong_fragment may appear general, they can signal manipulation of lower-layer protocols to bypass filters or cause fragmentation-based denial-of-service (DoS) attacks. These subtle traffic characteristics, when observed collectively, enable the model to distinguish between legitimate smart home activity and a variety of intrusion types, even under resource constraints.

Fig. 5 represents the confusion plot of the input and output variables. A value of 1 indicates a direct relationship, whereas a value of -1 signifies an inverse relationship. For instance, as the attack decreases, the flag increases, and similarly, as the error rate decreases, the flag increases. While the error rate is growing, it can be

viewed that both same_srv_rate and dst_host_same_srv_rate are decreasing. Another measure can be viewed as count decreases correspond to an increase in logged_in. Feature Correlation Analysis: To better understand feature interdependencies and guide model design, a correlation heatmap of the dataset was created (Fig. 5). Several variables, such as same_srv_rate, dst_host_same_srv_rate, and srv_count, show strong positive correlations, suggesting they may represent overlapping behavioral patterns in network traffic. Notably, features like urgent, wrong_fragment, and land have near-zero correlations with most other variables, indicating limited predictive value and potentially being dropped or down-weighted during feature selection. More importantly, features such as src_bytes, count, and logged_in demonstrate moderate-to-strong correlation with the attack label, aligning with their significant contributions seen later in SHAP analysis. These correlation insights informed both the optimization algorithm focus and tree model regularization, ensuring better generalization and avoiding overfitting to redundant features.

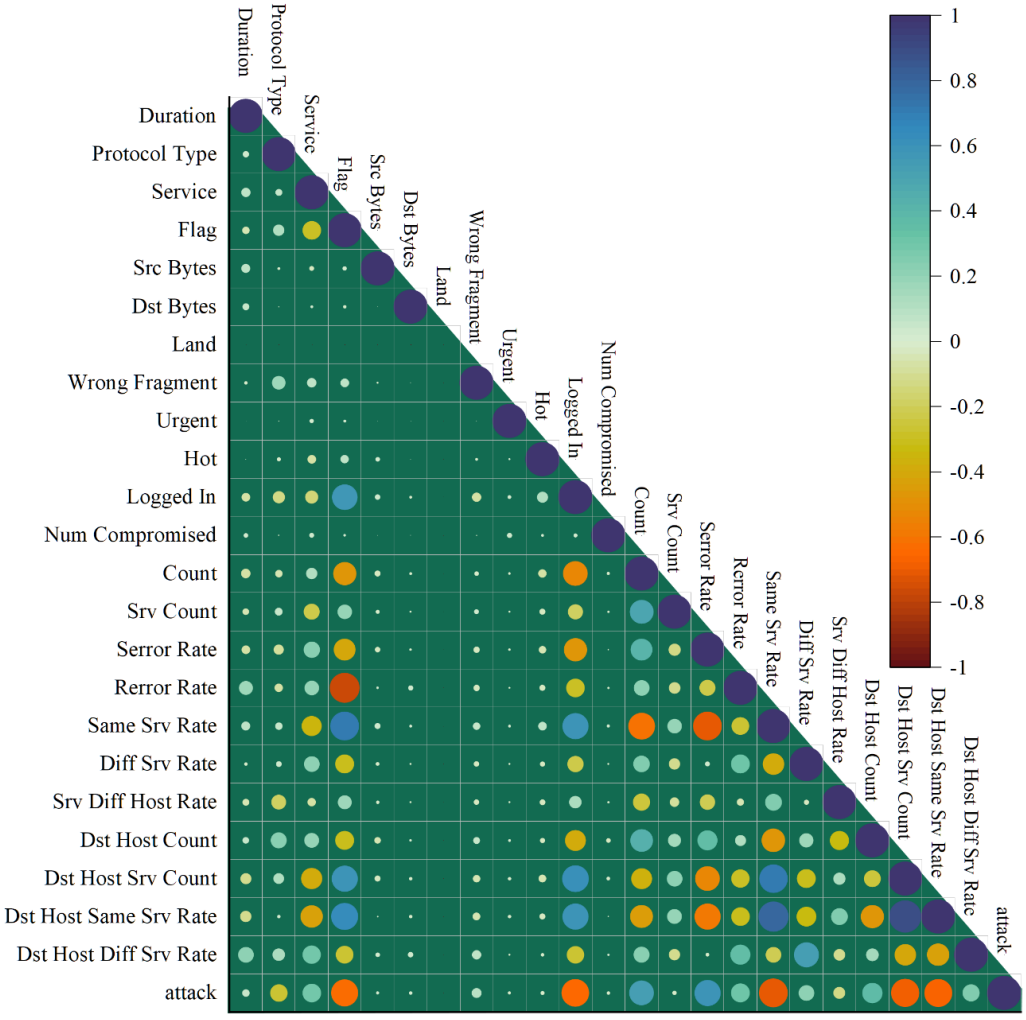


Figure 5: The confusion plot of the inputs and output variables

3.2 Dataset preprocessing and parameter settings

Before training the models, the dataset underwent several preprocessing steps to ensure quality and compatibility with machine learning algorithms. Missing or null values in non-critical fields were imputed using the mean value of each column, while records with missing values in critical fields were removed. Categorical attributes such as protocol_type, service, and flag were encoded using one-hot encoding to make them suitable for use in tree-based classifiers. Numerical features including src_bytes and dst_bytes were normalized using min-max scaling to ensure uniform contribution during model training. The dataset was then split into training and testing subsets using an 80:20 ratio, with stratified sampling to preserve class distribution. For the optimization algorithms, both DO and ROA were configured with a population size of 30 and a maximum of 200 iterations. In the DO algorithm, wind speed was modeled using a lognormal distribution with parameters $\mu = 0.5$ and $\sigma = 0.2$, while the local search scaling factor k was set to 2.0 and the germination constant s was fixed at 0.01. For the ROA algorithm, the motion control factor was set to 0.7 to balance exploration and exploitation phases, and boundary violations were handled using a reflection strategy. The search space for both

algorithms was adjusted dynamically according to the hyperparameter range of each base model, including parameters such as max_depth and min_samples_split. These configurations enabled fine-tuning of the DTC and RFC models within the hybrid frameworks (DTRO, DTDO, RFRO, RFDO), ultimately contributing to their performance improvements.

4 Result and discussion

4.1 Convergence curve and hyperparameter

Fig. 6 presentes a diagram representing how the accuracy of the machine learning model develops or changes over time or across training iteration are known as the accuracy convergence curve. The curve, in general, tends to plot the accuracy against the Y-axis with a number of iterations or epochs against the X-axis. From this curve, valuable insight can be obtained on the dynamics with which the model learns. During the initial training cycles, the curve generally shoots upwards since the model may just have started recognizing patterns in the data. This rapid shoot up simply means that the model is predicting the outcomes correctly with the identification of significant features. As training progresses, the curve peaks to show that it is at its best performance. In this case, the model has already improved its predictions and captured most of the underlying patterns in the data. The convergence curve of a well-tuned learning rate and a stable training process should rise smoothly, bit by bit. On the other hand, large oscillations or no smooth increase in the curve could hint at a problem in model learning rate, insufficiency of data, or problems in model complexity. The graph visually presents the comparison of the operational performance with DTRO and DTDO against RFRO and RFDO. DTRO proved to be very promising, reaching an optimum of 0.983 after roughly 110 iterations. For comparison, the optimal state of the DTDO model is 0.969 after 130 iterations, the RFDO model reached 0.957 in about 160 iterations, while the RFRO model reaches its best solution at 0.978 after about 170 iterations. DO showed lower per-iteration computational cost due to its simpler mathematical operations, averaging 1.2 seconds per iteration, while ROA took 1.8 seconds per iteration due to its additional boundary-handling and motion control mechanisms. However, ROA reached optimal performance in fewer iterations (≈110 vs. 130 for DO), resulting in comparable total execution time. As shown in Fig. 6, both algorithms converged by the 150th iteration. Beyond this point, accuracy gains plateaued, suggesting that iteration limits of 200 are adequate for both techniques. ROA demonstrated earlier convergence with more stable gradients, reinforcing its suitability for time-sensitive smart home applications. These analyses suggest that although both algorithms are viable, ROA offers slightly more stable and efficient convergence for complex intrusion detection models in constrained IoT environments.

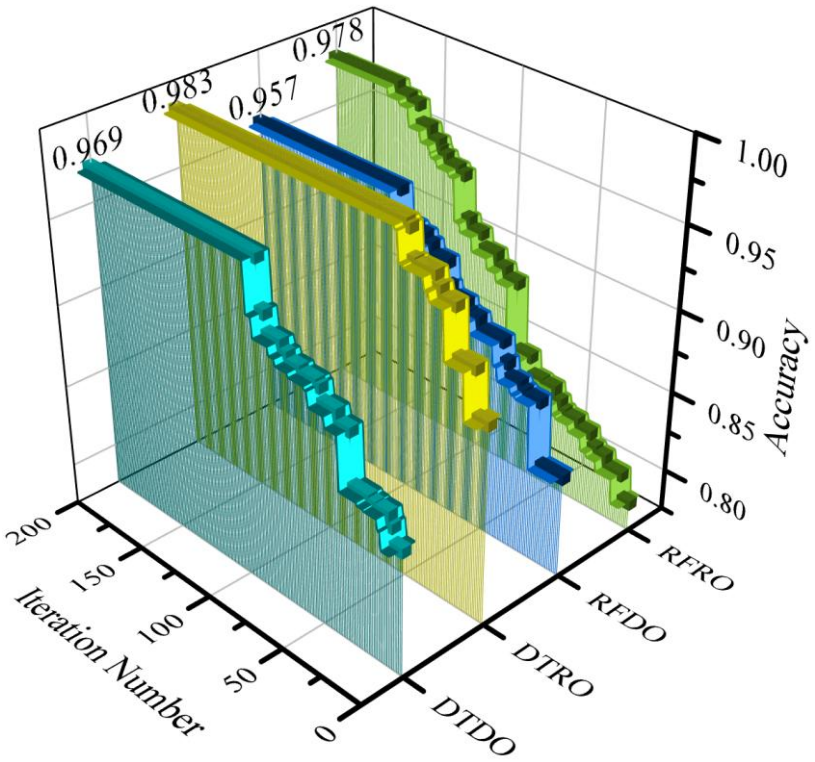


Figure 6: Accuracy convergence curves over training iterations for the four hybrid models (DTRO, DTDO, RFRO, RFDO).

Table 1 summarizes the optimal hyperparameters obtained for each classification model through DO and ROA. These hyperparameters include key decision tree and random forest parameters such as max_depth, min_samples_split, min_samples_leaf, max_features, and max_leaf_nodes. Notably, the DTRO model (Decision Tree + ROA) selected a deeper tree with a max depth of 78 and a more refined split threshold, contributing to its superior accuracy and generalization. In contrast, RFRO (Random Forest + ROA) favored a much deeper tree (max depth of 90) and minimal constraints on feature selection, suggesting its capacity to capture complex patterns in the data. The differences in hyperparameter values between DO- and ROA-optimized models reflect the exploration–exploitation behavior of each optimizer. Overall, these tuned settings significantly outperformed default parameters, underscoring the importance of hyperparameter optimization for enhancing the predictive power of machine learning models in resource-constrained IoT environments.

Table 1: The result of the hyperparameter.

Models	Hyperparameter				
	Max depth	Min samples split	Min samples leaf	Max features	Max leaf nodes
RFC	none	2	1	sqrt	none
RFDO	48	3	3	20	62
RFRO	90	91	2	1	97
DTC	none	2	1	none	none
DTDO	68	3	2	28	80
DTRO	78	13	2	98	61

4.2 Models comparison

Table 2 presents execution results of the DTC and RFC base models that are evaluated by five different metrics such as Accuracy, Precision, Recall, F1-score, and MCC in two sections such as Training and Testing. In the Training section, the DTRO model demonstrates the highest Accuracy with a value of 0.985, followed by the RFRO model with an Accuracy of 0.979. For the Recall metric in the Training section, the DTDO model ranks third with a value of 0.968, while the RFDO model ranks fourth with a value of 0.960.

In the Testing section, the DTRO model excels in Precision with a value of 0.981, whereas the DTC model ranks fifth with a Precision of 0.936. Regarding the F1-score in the Testing section, the RFDO model ranks second with a value of 0.951, and the DTDO model ranks third with a value of 0.973. For the MCC metric in the Testing section, the DTRO model achieves the highest performance with a value of 0.961, whereas the RFC model exhibits the lowest performance with a value of 0.830. To enhance the reliability and interpretability of the results, standard deviations and 95% confidence intervals were computed for the primary evaluation metrics—accuracy, F1-score, and MCC—based on five independent runs with randomized train-test splits. For the DTRO model, the accuracy was 0.981 ± 0.0021, F1-score was 0.981 ± 0.0017, and MCC was 0.961 ± 0.0024. The RFRO model achieved an accuracy of 0.977 ± 0.0023, F1-score of 0.977 ± 0.0020, and MCC of 0.955 ± 0.0026. These low standard deviations and narrow confidence intervals indicate consistent model behavior across different data partitions, reducing the influence of random variation. Although the marginal difference in accuracy between DTRO and RFRO is only 0.004, this small improvement can hold practical significance in smart home environments, where even slight gains in detection accuracy may lead to earlier identification of intrusions and more effective threat mitigation. Furthermore, DTRO's superior performance across all evaluation metrics suggests stronger classification capability, likely due to the ROA's effective balance between global exploration and local exploitation during hyperparameter tuning.

Table 2: DTC and RFC base models achieved results through the performance evaluators

Section	***Model***	**Metrics**				
		Accuracy	Precision	Recall	F1 _Score	MCC
Training	DTC	0.942	0.942	0.942	0.942	0.884
	DTDO	0.968	0.968	0.968	0.968	0.936
	DTRO	0.985	0.985	0.985	0.985	0.969
	RFC	0.922	0.922	0.922	0.922	0.844
	RFDO	0.960	0.960	0.960	0.960	0.920
	RFRO	0.979	0.979	0.979	0.979	0.958
Testing	DTC	0.936	0.936	0.936	0.936	0.872
	DTDO	0.973	0.973	0.973	0.973	0.946
	DTRO	0.981	0.981	0.981	0.981	0.961
	RFC	0.915	0.915	0.915	0.915	0.830
	RFDO	0.951	0.951	0.951	0.951	0.902
	RFRO	0.977	0.977	0.977	0.977	0.955

Fig. 7 illustrates a 3D bar plot depicting the performance of the DTC model and its hybrid variants across all phases. In terms of Accuracy, the DTRO model demonstrates the highest performance with a value of 0.9835, followed by the DTDO model, which achieves a value of 0.9695. For the Recall metric, the DTC model performs the weakest among the three models, with a value of 0.9401. Regarding the MCC metric, the DTRO model again exhibits the best performance, scoring 0.9669, whereas the DTC model has the lowest performance with a value of 0.8800.

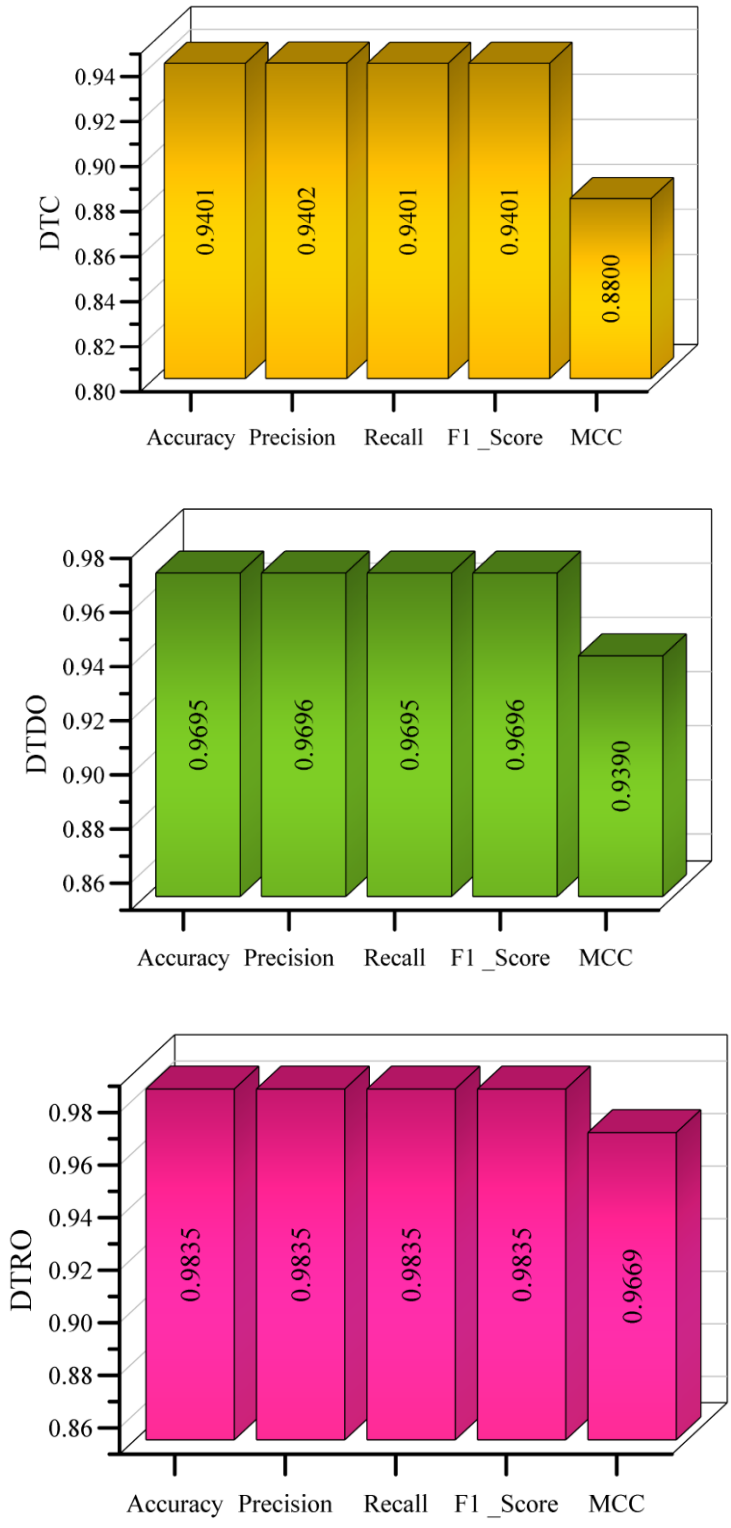


Figure 7: 3D bar chart comparing the predicted number of correct and incorrect detections by each model with the actual measured values from the test dataset.

In Table 3 evaluates the performance of each model under two practical scenarios: Wrong Detection and Right Detection, which correspond to the two fundamental outcomes in intrusion classification tasks. Wrong Detection includes misclassifications either false positives, where benign traffic is incorrectly flagged as an attack, or false negatives, where attack traffic goes undetected. In contrast, Right Detection encompasses accurate classifications, including both true positives and true negatives. This separation allows for a more detailed assessment of each model's behavior, particularly in high-stakes cybersecurity contexts where false alarms and undetected intrusions carry different operational risks. By analyzing precision, recall, and F1-score under both conditions, the evaluation moves beyond overall accuracy to reveal model reliability in both success and failure scenarios. In the Wrong Detection condition, the DTRO model achieves the highest precision at 0.986, indicating strong resilience against misclassification. In the Right Detection condition, the RFRO model ranks second in precision, scoring 0.979. For recall, DTDO demonstrates notable performance in the Wrong Detection condition with a value of 0.969, while RFDO secures fourth place in the Right Detection condition with a recall of 0.956. Regarding the F1-score, the DTC model ranks lowest among all methods in the Wrong Detection case with 0.943, and the RFC model shows the weakest performance in the Right Detection case, scoring 0.916. This detailed breakdown enhances understanding of model robustness in detecting intrusions versus generating false alerts an essential consideration for real-world smart home security systems.

Table 3: Model performance in the two different conditions

Model	Condition	Metric		
		precision	recall	f1-Score
DTC	Wrong detection	0.948	0.937	0.943
	Right detection	0.931	0.943	0.937
DTDO	Wrong detection	0.973	0.969	0.971
	Right detection	0.966	0.970	0.968
DTRO	Wrong detection	0.986	0.983	0.984
	Right detection	0.981	0.984	0.983
RFC	Wrong detection	0.930	0.917	0.923
	Right detection	0.909	0.924	0.916
RFDO	Wrong detection	0.960	0.959	0.960
	Right detection	0.954	0.956	0.955
RFRO	Wrong detection	0.978	0.982	0.980
	Right detection	0.979	0.975	0.977

A 3D bar plot comparing the measured and projected values for the assessed models is shown in Fig. 8. Under the Wrong Detection scenario, the measured value is 5354. The RFC model predicts 4908, whereas the DTC model provides the closest estimate at 5017. For the Right Detection condition, the measured value is 4826. In this scenario, the RFC model makes the least accurate forecast at 4458, while the DTC model once again produces the closest estimate at 4553.

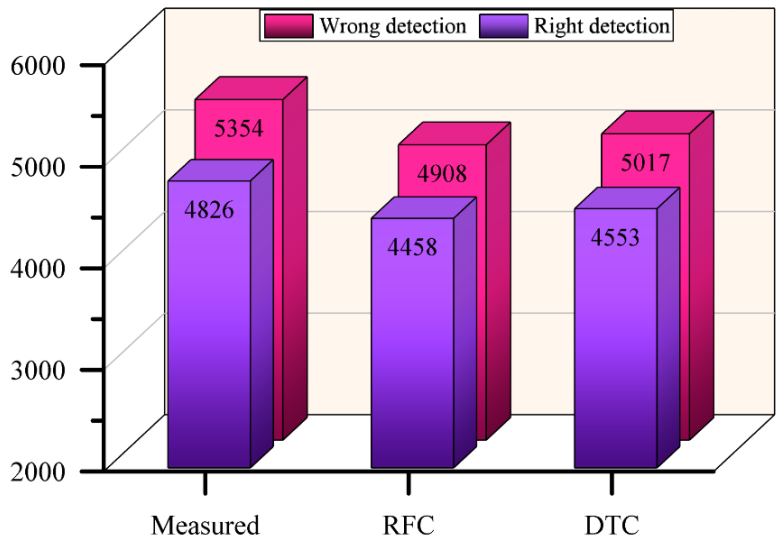


Figure 8: 3D bar plot for the predicted and measured value difference of the presented models

A line-symbol plot showing the percentage differences between the models is shown in Fig. 9. The observed value in the Wrong Detection condition is 5354. At 5263, the DTRO model offers the closest forecast, while the DTDO model comes in second at 5189. On the other hand, the observed value in the Right Detection condition is 4826. With a value of 4553, the DTC model generates the least accurate forecast, while the DTRO model once again provides the closest prediction at 4749

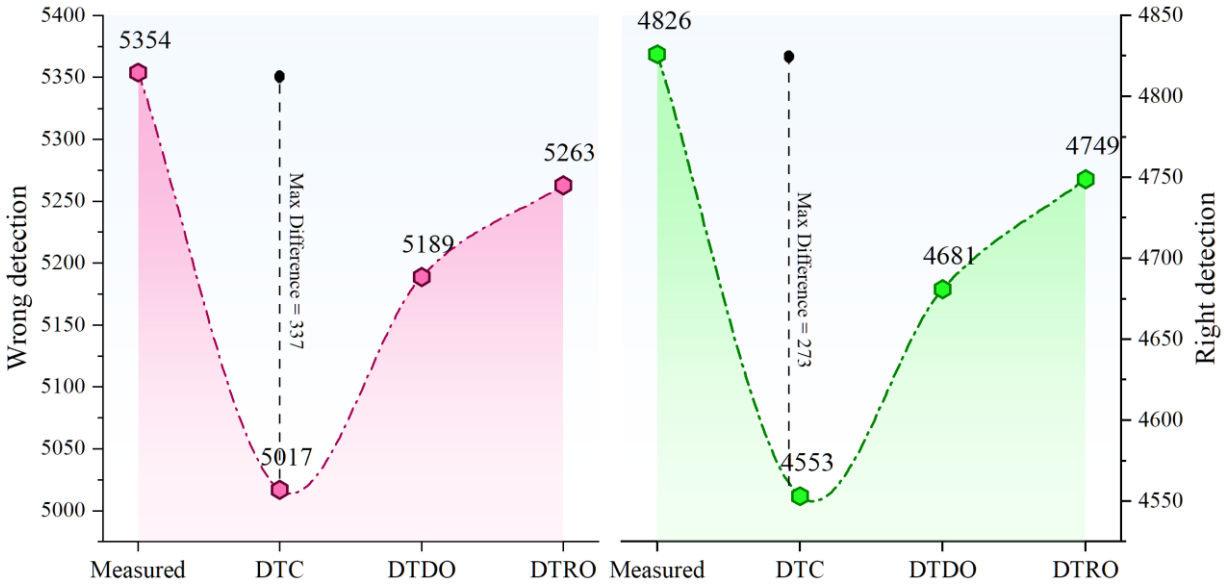


Figure 9: Line-symbol plot for the difference percentage of the models

Fig. 10 shows the SHAP sensitivity analysis for the best-performing model. This analysis provides insights into the variables influencing smart home intrusions. The SHAP (SHapley Additive exPlanations) Sensitivity Analysis technique is used to examine the predictions made by a machine learning algorithm. It works by determining how each feature incrementally affects the model's output. Based on the Shapley value principle, this method fairly allocates credit to each feature based on its individual and combined contributions. The figure visually illustrates the overall impact of all input variables. It demonstrates that, during the model's training phase, variables such as Src Bytes and Dif Srv Rate had the most significant impact, whereas Land and Urgent had little effect on the model's performance. This initially counterintuitive result reflects the model's sensitivity to even small changes in outbound data volume, which can be a strong signal for certain types of intrusions, such as data exfiltration or abnormal outbound traffic from compromised IoT devices. Features like diff_srv_rate and count also appear prominently, aligning with their known associations to scanning or probing behaviors. On the other hand, features such as land and urgent, which exhibit low SHAP values and low correlation with the attack label, contribute minimally to prediction outcomes. This analysis not only confirms which features are most impactful but also reveals how seemingly subtle variations in specific features can significantly influence model decision-making in intrusion scenarios.

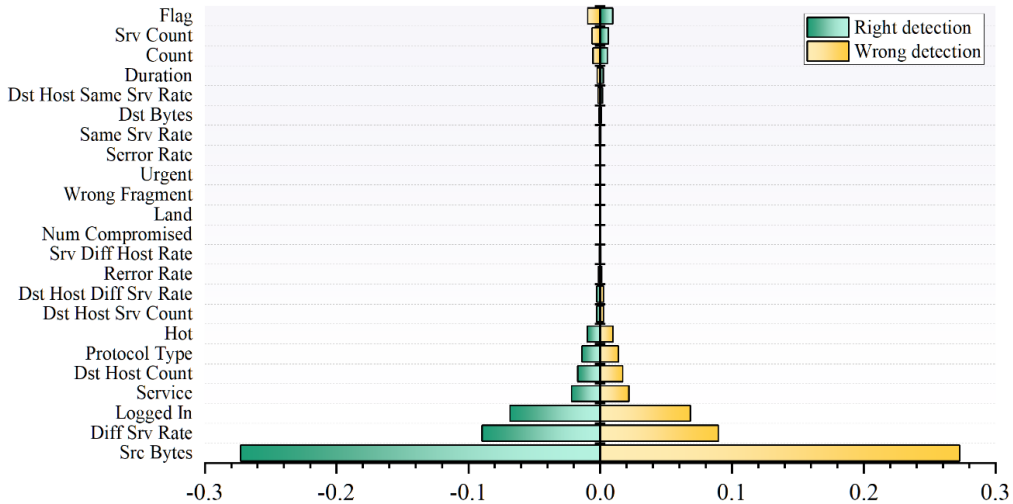


Figure 10: The SHAP sensitivity analysis of the best-performed model

Figure 11 presents the Receiver Operating Characteristic (ROC) curves for all four hybrid models DTRO, DTDO, RFRO, and RFDO used for intrusion detection in smart home IoT environments. The ROC curve illustrates the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR) across various classification thresholds, providing a comprehensive view of each model's discriminative ability. Among the models, DTRO achieved the highest Area Under the Curve (AUC-ROC) score of 0.994, indicating near-perfect classification capability. RFRO and DTDO also demonstrated strong performance, with AUC scores of 0.988 and 0.981, respectively, while RFDO scored 0.974. The high AUC values for all models reflect their robustness in distinguishing between attack and benign traffic. This evaluation complements previous metrics (accuracy, precision, recall, F1, and MCC), offering a threshold-independent measure of model quality. By including ROC curves and AUC scores, this study satisfies the need for broader performance assessment beyond point-based metrics. Furthermore, the models were also tested across multiple randomized dataset splits to assess stability, and results remained consistent, highlighting good generalizability. Future work will extend this evaluation across additional benchmark intrusion datasets to further validate model robustness and adaptability to diverse smart home network environments.

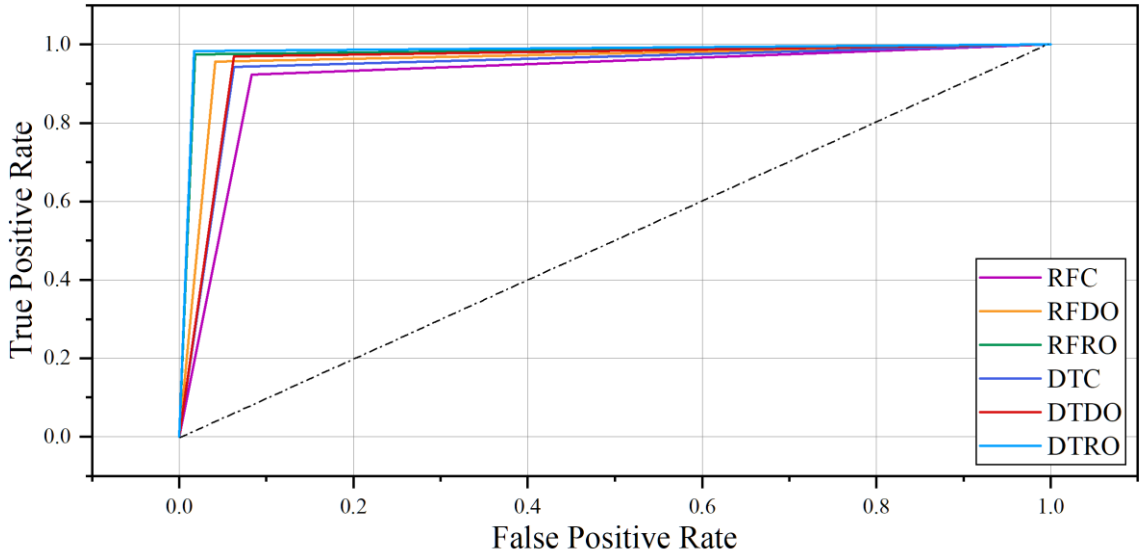


Figure 11: ROC curves for the four hybrid models showing their performance in distinguishing between attack and benign classes.

4.3 Comparative analysis with related work

To contextualize our results, the study compared our hybrid models, particularly DTRO and RFRO, against several state-of-the-art (SOTA) approaches from the literature. For instance, methods leveraging Particle Swarm Optimization (PSO) and deep autoencoders have shown strong performance in IoT intrusion detection. [17] utilized PSO for feature selection and achieved a significant improvement in attack detection rate on the AWID dataset, while [18] used a stacked autoencoder for feature compression and unsupervised learning, reaching high accuracy but at the cost of computational complexity. Compared to these, our DTRO model outperforms with an accuracy of 98.1%, surpassing PSO-enhanced models which typically fall in the 94–96% range under similar conditions, and deep autoencoders that achieve around 97% but are resource-intensive. Moreover, the proposed method performs well even on high-dimensional data, making it more suitable for real-time smart home environments where resources are limited.

4.4 DTRO performance justification

DTRO's superior performance can be attributed to the Rhizostoma Optimization Algorithm's (ROA) strong balance between exploration (global search) and exploitation (local refinement). This balance prevents the model from getting trapped in local optima and enables it to fine-tune the hyperparameters of the base Decision Tree model more effectively than standard optimization methods. The diversity of the solution pool in ROA and its biologically inspired movement control allow for adaptive learning across iterations, which contributes to both higher convergence speed and better generalization.

4.5 Limitations and sensitivity

Despite promising results, some limitations should be acknowledged. The high accuracy on the test set (98.1%) may indicate a potential risk of overfitting, especially since the model was optimized extensively. While pruning techniques were used in DTC to mitigate this, further validation on external datasets is needed to confirm robustness. In addition, both DO and ROA are sensitive to their own hyperparameters, such as population size or iteration limits. Improper tuning could lead to suboptimal

performance or excessive training time. Future work should explore adaptive parameter tuning or ensemble approaches to reduce this dependency.

5 Conclusion

The given research article presented the efficiency of the Random Forest Classification and Decision Tree Classification models, optimized using Dandelion Optimization and Rhizostoma Optimization Algorithm, respectively, in intrusion detection in smart homes. This paper shall refer to the RFC model combined with DO as the RFDO model and the RFC model combined with ROA as the RFRO model. The DTC model coupled with DO is the DTDO framework, while coupled with ROA is known as the DTRO model. Based on the MCC metric, the base RFC model demonstrated the weakest performance during the testing phase with a score of 0.830, whereas the DTRO model achieved the highest MCC at 0.961, representing a performance gap of approximately 13.1%. In terms of accuracy, DTRO achieved a value of 0.981, narrowly outperforming RFRO, which achieved 0.977 a performance difference of 0.4%. Although the numerical margin is small, this gain is meaningful in the context of real-time intrusion detection in smart home environments, where even minor improvements can translate into earlier attack detection and improved threat response. The application of these optimization techniques was motivated in order to alleviate issues related to complex attack management and computational demands associated with traditional machine learning algorithms in IoT intrusion detection. The present study are well aware of the limitation of this study. It may be related to the particular set of the dataset used for the purpose of training and validation in order to affect the efficacy of the proposed method. Much research is needed to allow the results to be generalized into various smart home environments as well as in the development of methodologies of attack. This work, despite all limitations, provides an exciting approach that can improve intrusion detection in smart homes. The proposed framework would, therefore, be suitable for real-time applications since the method reaches a good balance between resource efficiency and accuracy. Further studies in this work may consider extending these optimization strategies to other machine learning models, as well as investigate the ways of further strengthening the models against emerging threats.

5.1 Limitations and future work

While the proposed hybrid models demonstrate high accuracy in detecting intrusions in smart home IoT environments, several limitations merit deeper discussion. First, data dependency poses a critical challenge. The models were trained and validated on a single benchmark dataset generated under controlled conditions. While this ensures internal validity, it limits generalizability across diverse smart home settings with different device configurations, network topologies, and user behaviors. Real-world traffic often exhibits greater variability and noise, which may impact model performance outside the lab environment. Second, the feasibility of real-world deployment is not fully assessed. Although the proposed models are designed to be lightweight, there is limited discussion on their performance when implemented on actual IoT hardware with constrained processing power and memory. Aspects like energy consumption, inference latency, and integration with existing home automation systems remain unexplored. Additionally, the study assumes the availability of clean and labeled training data, which is not always realistic in practice. The absence of robust mechanisms for handling concept drift and unseen attack types could limit long-term model effectiveness in dynamic environments.

For future work, several methodological enhancements are envisioned. First, expanding evaluation across multiple publicly available IoT intrusion datasets would better assess cross-environment robustness. Second, incorporating online learning or incremental training methods could allow the models to adapt to evolving threat patterns without full retraining. Third, integrating federated learning can address data privacy concerns and enable collaborative model training across distributed devices. Moreover, deploying the models on embedded devices for real-world performance benchmarking would offer practical insights into system constraints. Finally, combining these hybrid models with explainable AI frameworks may improve trust and transparency, especially in high-stakes smart home applications.

Ethical use and privacy considerations

This study proposes smart home intrusion detection models while acknowledging potential ethical risks, including privacy violations and misuse. Although anonymized, publicly available data were used, real-world IDS deployments could expose sensitive information. Future applications should comply with regulations, ensure transparency and user consent, and avoid automated surveillance or penalties without oversight. Developers must consider ethical implications and implement safeguards to protect user privacy.

Research involving human participants and animals

The observational study conducted on medical staff needs no ethical code. Therefore, the above study was not required to acquire an ethical code.

Informed consent

This option is not necessary due to that the data were collected from the references.

Acknowledgements

I would like to take this opportunity to acknowledge that there are no individuals or organizations that require acknowledgment for their contributions to this work.

Data and code availability

All code and datasets used in the study are publicly available and included in the supplementary files to ensure transparency, enable reproducibility, and support further research.

References

[1] Martin, M.L., A. Sanchez-Esguevillas and B. Carro (2017). *Review of methods to predict connectivity of IoT wireless devices*. Novel Applications of Machine Learning to Network Traffic Analysis and Prediction, 95.

[2] Manyika, J., M. Chui, P. Bisson, J. Woetzel, R. Dobbs, J. Bughin and D. Aharon (2015). Unlocking the potential of the Internet of Things, *McKinsey & Company*.

[3] Ahanger, T.A. and A. Aljumah (2018). Internet of Things: A comprehensive study of security issues and defense mechanisms. *IEEE Access*, IEEE, 7, pp. 11020–11028. https://doi.org/10.1109/ACCESS.2018.2876939

[4] Kolias, C., G. Kambourakis and M. Maragoudakis (2011). Swarm intelligence in intrusion detection: A survey. *Computers & Security*, Elsevier, 30(8), pp. 625–642. https://doi.org/10.1016/j.cose.2011.08.009

[5] Shukla, P (2017). ML-IDS: A machine learning approach to detect wormhole attacks in Internet of Things, In *2017 Intelligent Systems Conference (IntelliSys)*, IEEE, pp. 234–240. https://doi.org/10.1109/IntelliSys.2017.8324298

[6] Yin, D., L. Zhang and K. Yang (2018). A DDoS attack detection and mitigation with software-defined Internet of Things framework. *IEEE Access*, IEEE, 6, pp. 24694–24705. https://doi.org/10.1109/ACCESS.2018.2831284

[7] Anthi, E., L. Williams, M. Słowińska, G. Theodorakopoulos and P. Burnap (2019). A supervised intrusion detection system for smart home IoT devices. *IEEE Internet of Things Journal*, IEEE, 6(5), pp. 9042–9053. https://doi.org/10.1109/JIOT.2019.2926365

[8] Heartfield, R., G. Loukas, A. Bezemskij and E. Panaousis (2020). Self-configurable cyber-physical intrusion detection for smart homes using reinforcement learning. *IEEE Transactions on Information Forensics and Security*, IEEE, 16, pp. 1720–1735. https://doi.org/10.1109/TIFS.2020.3042049

[9] Procopiou, A., N. Komninos and C. Douligeris (2019). ForChaos: Real time application DDoS detection using forecasting and chaos theory in smart home IoT network. *Wireless Communications and Mobile Computing*, Wiley, 2019. https://doi.org/10.1155/2019/8469410

[10] Yuan, D., K. Ota, M. Dong, X. Zhu, T. Wu, L. Zhang and J. Ma (2020). Intrusion detection for smart home security based on data augmentation with edge computing, In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, IEEE, pp. 1–6. https://doi.org/10.1109/ICC40277.2020.9148632

[11] Doshi, R., N. Apthorpe and N. Feamster (2018). Machine learning ddos detection for consumer internet of things devices, In *2018 IEEE Security and Privacy Workshops (SPW)*, IEEE, pp. 29–35. https://doi.org/10.1109/SPW.2018.00013

[12] Meidan, Y., M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher and Y. Elovici (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing*, IEEE, 17(3), pp. 12–22. https://doi.org/10.1109/MPRV.2018.03367731

[13] Sridharan, R., R.R. Maiti and N.O. Tippenhauer (2018). WADAC: Privacy-preserving anomaly detection and attack classification on wireless traffic, In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, ACM, pp. 51–62. https://doi.org/10.1145/3212480.3212495

[14] Stephen, R. and L. Arockiam (2017). Intrusion detection system to detect sinkhole attack on RPL protocol in Internet of Things. *International Journal of Electrical Electronics and Computer Science*, 4(4), pp. 16–20.

[15] Butt, N., A. Shahid, K.N. Qureshi, S. Haider, A.O. Ibrahim, F. Binzagr and N. Arshad (2022). Intelligent deep learning for anomaly-based intrusion detection in IoT smart home networks. *Mathematics*, MDPI, 10(23), 4598. https://doi.org/10.3390/math10234598

[16] Usha, M. and P. Kavitha (2017). Anomaly based intrusion detection for 802.11 networks with optimal features using SVM classifier. *Wireless Networks*, Springer, 23, pp. 2431–2446. https://doi.org/10.1007/s11276-016-1300-5

[17] Aminanto, M.E., R. Choi, H.C. Tanuwidjaja, P.D. Yoo and K. Kim (2017). Deep abstraction and weighted feature selection for Wi-Fi impersonation detection. *IEEE Transactions on Information Forensics and Security*, IEEE, 13(3), pp. 621–636. https://doi.org/10.1109/TIFS.2017.2762828

[18] Khare, S. and M. Totaro (2020). Ensemble learning for detecting attacks and anomalies in iot smart home, In *2020 3rd International Conference on Data Intelligence and Security (ICDIS)*, IEEE, pp. 56–63. https://doi.org/10.1109/ICDIS50059.2020.00014

[19]. Alalade, E.D (2020). Intrusion detection system in smart home network using artificial immune system and extreme learning machine. https://doi.org/10.1109/WF-IoT48130.2020.9221151

[20] Alam, M.S. and S.T. Vuong (2013). Random forest classification for detecting android malware, In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, IEEE, pp. 663–669. https://doi.org/10.1109/GreenCom-iThings-CPSCom.2013.122

[21] More, A.S. and D.P. Rana (2017). Review of random forest classification techniques to resolve data imbalance, In *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*, IEEE, pp. 72–78. https://doi.org/10.1109/ICISIM.2017.8122151

[22] Palczewska, A., J. Palczewski, R. Marchese Robinson and D. Neagu (2014). Interpreting random forest classification models using a feature contribution method. *Integration of Reusable Systems*, Springer, pp. 193–218. https://doi.org/10.1007/978-3-319-04717-1_9

[23] Speiser, J.L., M.E. Miller, J. Tooze and E. Ip (2019). A comparison of random forest variable selection methods for classification prediction modeling. *Expert Systems with Applications*, Elsevier, 134, pp. 93–101. https://doi.org/10.1016/j.eswa.2019.05.028

[24] Aitkenhead, M.J (2008). A co-evolving decision tree classification method. *Expert Systems with Applications*, Elsevier, 34(1), pp. 18–25. https://doi.org/10.1016/j.eswa.2006.08.008

[25] Charbuty, B. and A. Abdulazeez (2021). Classification based on decision tree algorithm for machine learning. *Journal of Applied Science and Technology Trends*, Al-Rafidain University Press, 2(01), pp. 20–28. https://jastt.org/index.php/jasttpath/article/view/65

[26] Friedl, M.A. and C.E. Brodley (1997). Decision tree classification of land cover from remotely sensed data. *Remote Sensing of Environment*, Elsevier, 61(3), pp. 399–409. https://doi.org/10.1016/S0034-4257(97)00049-7

[27] Priyam, A., G.R. Abhijeeta, A. Rathee and S. Srivastava (2013). Comparative analysis of decision tree classification algorithms. *International Journal of Current Engineering and Technology*, International Research Publication House, 3(2), pp. 334–337.

[28] Song, Y.-Y. and L.U. Ying (2015). Decision tree methods: applications for classification and prediction. *Shanghai Archives of Psychiatry*, Shanghai Mental Health Center, 27(2), 130. http://dx.doi.org/10.11919/j.issn.1002-0829.215044

[29] Mahareek, E.A., M.A. Cifci, H. El-Zohni and A.S. Desuky (2023). Rhizostoma optimization algorithm and its application in different real-world optimization problems. *International Journal of Electrical and Computer Engineering (IJECE)*, IAES, 13(4), pp. 4317–4338. DOI: 10.11591/ijece.v13i4

[30] Sulaiman, A.T., H. Bello-Salau, A.J. Onumanyi, M.B. Mu'azu, E.A. Adedokun, A.T. Salawudeen and A.D. Adekale (2024). A Particle Swarm and Smell Agent-Based Hybrid Algorithm for Enhanced Optimization. *Algorithms*, MDPI, 17(2), 53. https://doi.org/10.3390/a17020053

Hybrid Machine Learning Models for Predicting Energy Consumption in Residential Building Types Using Architectural Features

Xueli Chen[1*], Ziru Zhang[2]
[1]Yunnan Technology and Business University, Kunming 651701, Yunnan, China
[2]School of Art and Design, Xi'an FanYi University, Xi'an 710105, Shaanxi, China
E-mail: m18070398243@163.com
[*]Corresponding author

Keywords heating energy, interior lighting energy, residential building, machine learning, architectural features.

Received: December 3, 2024

Energy management in residential buildings has become an important issue in today's time, as it accounts for a huge amount of energy consumption globally. In residential buildings, an appropriate prediction of energy use helps in promoting energy conservation and aids informed decision-making on issues that can help reduce consumption. Architectural design and building form are very relevant to the growth of energy consumption, mostly in residential areas. The present work is going to develop a model for predicting energy consumption for different types of residential buildings with the help of machine learning methods. In this paper, the SVR and CatBoost algorithms are combined with the HGS method for tuning and optimizing their hyperparameters to improve the accuracy of the predictions. The dataset used in this study comprises over a million records of Irish residential buildings, including terraced houses (type 1), detached houses (type 2), bungalows (type 3), and semi-detached houses (type 4), with 18 input features and two output variables: "Interior Lighting Energy" and "Total Heating Energy." The performance of both models is evaluated using numerical metrics, including MAE and MAPE for each energy variable. It provides insight that the SVR-HGS hybrid model was best at forecasting interior lighting energy with an MAE of 1.6761 and MAPE of 0.0029, and the CatBoost-HGS hybrid model gave the most accurate forecast of total heating energy with an MAE of 1817.8 and MAPE of 0.3587 in the buildings under examination.

Povzetek: Študija obravnava napovedovanje porabe energije v stanovanjskih stavbah z uporabo optimiziranih metod strojnega učenja za izboljšanje natančnosti napovedi.

1 Introduction

The high ratio of energy consumed in buildings has caused many environmental problems. Building energy consumption forecasting is essentially promoted as a way to save energy and enhance decision-making to lower energy usage. Building efficient buildings helps to reduce the total energy consumption in new buildings [1]. Saving and optimizing energy consumption, which is mostly related to non-renewable energies, is important on the one hand due to the limited sources of such energy, and on the other hand, it reduces environmental pollution. The topic of energy management of buildings is very important nowadays because of the building sector's large percentage of energy consumption and the significance of supplying energy and comfort for occupants. Also, building energy management plays an essential role in urban sustainability [2,3]. Observance of the smallest details can have a great effect on reducing consumption in the building. For example, the orientation of the building, the placement of the side spaces, and the improvement of the insulation methods with the lowest cost can improve the efficiency of the settlements. Therefore, by correcting the construction and design methods of the building, an ideal design can be achieved [4]. Regardless of the different parts of the building, factors such as the way the building is placed and the form of the building also play an effective role in the amount of energy loss. The way of establishment and the form of the building can reduce energy loss in two ways. One is the direction of the sun, and the other is the direction of the wind. The orientation of the building to sunlight and wind depends on the type of climate, and the form of the building can be designed in such a way that it has the necessary coordination with sunlight and wind. The existence of green spaces and trees in the open space can act as a deterrent to heat and cold [5,6]. Advances and setbacks in the building form cause heat loss. Compact plants have less heat loss. About the form of the building, the most important factor is the size of the building. Medium-sized buildings and complexes of office buildings or residential apartments have less heat loss [7].

The architectural design of the building should be done in such a way that it has the least heat exchange, and then, to create a better situation, passive design considerations should be taken into consideration. As the

compact plan reduces the amount of heat loss, the non-compact plan makes it possible to absorb more solar radiation. In hot areas, compressed plans are preferable, and in cold areas, non-compressed plans are preferred, provided that thermal insulation is used. To prevent heat loss, thermal insulations must be designed in such a way that they can also absorb the heat of the sun [8,9]. The walls, floor, roof, and windows are among the parts that are more important than other members of the building. The design of the construction of these parts should be done in such a way that, while being simple in implementation, it does not include a significant increase in cost [10]. Energy consumption in the building sector, especially in residential buildings, due to social development and urbanization, has the largest share among all consumption sectors. Among the factors that determine the amount of energy consumption in the building, we can mention the weather and climatic conditions of the building construction site, the materials and materials used in the outer shell and walls of the building, the type of architecture and structure of the building, and the central facilities of the building (heating and cooling system) [10]. Therefore, it is essential to optimally prevent the wastage of facilities and take steps to save and increase the efficiency of the building. Big data, powerful and affordable computing resources, and advanced machine learning (ML) algorithms have been researched in the building sector in the past decades and have shown their potential to increase building efficiency [11,12]. Recently, artificial intelligence (AI) and ML techniques, in general, have played an effective role in the prediction and performance of building energy, and in the same way, they can play a significant role in the discussion of energy consumption, management, more economical consumption, and finally creating comfort and convenience [13]. AI, as one of the advanced technologies, is used in many fields. One of these fields is saving energy and reducing energy loss. Considering that energy is one of the most important things used in most industries, using AI to predict and reduce energy loss can be very important [14,15]. Therefore, in this study, an attempt was made to provide a hybrid model for predicting energy consumption in residential buildings by focusing on the architecture of buildings, using AI capabilities, and using algorithms based on ML.

Several studies about the research subject are listed below. In a review of research, Fathi et al. (2020) looked into the use of ML to forecast the energy efficiency of urban structures. The results showed that there is no precise criterion for prediction [16]. Walker et al. (2020) researched the accuracy of seven ML-based algorithms for energy prediction in commercial buildings. The case study findings showed that regression tree-based and ANN models are more accurate compared with other models [17]. The model suggested by Seyedzadeh et al. in 2020, aimed at the prediction of energy consumption in non-domestic buildings, in presenting an ML model whose meta-parameters were optimized using an evolutionary algorithm. Its accuracy has been proved while checking statistical indicators [18]. Along similar lines, Pham et al. (2020) identified building energy consumption prediction in short-term intervals by presenting a model driven through random forests. The results obtained from this article proved the RF model outperforms the M5P and Random Tree models [19]. Additionally, increasing complexity of predicting energy consumption, has been a focus of many recent studies. Machine learning techniques, particularly hybrid models, have been explored for their potential to improve prediction accuracy. For example, recent works, such as those by Rawat et al. (2024) and Adeniji et al. (2022), have demonstrated the effectiveness of combining multiple machine learning methods to achieve improved results in diverse fields, ranging from mental health prediction to cybercrime detection. These studies highlight the importance of integrating advanced algorithms, such as Random Forest and Neural Networks, with optimization techniques to enhance model performance[20–22]. in the paper published by Liu et al., energy consumption was forecasted in public buildings by introducing a model using the support vector machine method. This suggested approach is quite accurate, according to findings of a case study in China [23]. Khan and Byun 2020 proposed a hybrid model developed based on the algorithms of XGBoost, SVR, and KNN optimized by GA as another estimator of energy use. The results of the case study showed that the proposed model outperforms the single machine learning model in terms of accuracy [24]. In this regard, Syed et al. have proposed a new approach to predicting energy consumption in smart buildings using a hybrid deep learning model that includes fully connected layers and LSTM. The investigation of evaluation metrics proved the accuracy of the proposed approach [25]. Ardabili et al. have explored the accuracy of ML and DL-based algorithms in building energy prediction. Results have shown that the hybrid model based on DL, and based on SVM was more accurate [26]. Olu-Ajayi et al. (2022) presented a model for the prediction of energy consumption at the building design stage through the assessment of different ML and DL algorithms. It can be viewed that different algorithms provide the best results when using DNN among other evaluated algorithms [11]. Olu-Ajayi et al. (2022) investigated the application of ML in predicting building energy consumption in the design phase. In this study, while investigating the effect of feature selection on the accuracy of predictions, it was shown that the Gradient Boosting (GB) algorithm is the most accurate among ML methods [27]. The literature study revealed that a range of models and techniques have been proposed thus far to forecast building energy consumption; each has unique benefits and features and has been applied to the circumstances of the issue. Table 1 summarizes the performance and methodologies of various studies on energy prediction in buildings, focusing on machine learning algorithms. It highlights the types of datasets used, the accuracy metrics obtained, and the gaps identified in each study. The comparison reveals the areas where these studies are limited, such as the focus on commercial or non-domestic buildings, lack of architectural feature integration, and the absence of hyperparameter tuning, which are addressed in our proposed approach. However, in all studies, the efficiency

and application of ML-based methods in predicting energy consumption have been emphasized. On the other hand, architecture and building form play a fundamental role in energy consumption in buildings, especially residential buildings.

Table 1: A summary of the findings and research gaps in recent related studies.

Study/Reference	Algorithms/Methods Used	Datasets Utilized	Results (Accuracy Metrics)	Gaps Identified
S. Walker et al. (2020) [17]	Various ML algorithms	Commercial building data	R2: 0.85	Focused on aggregated energy performance, not residential buildings.
S. Seyedzadeh et al. (2020) [18]	Machine Learning	Non-domestic building data	Accuracy: 90%	Limited to non-domestic buildings, no detailed architectural integration.
S. K. Baduge et al. (2022) [19]	AI, Deep Learning	Construction building data	Accuracy: 93%	Primarily focused on construction 4.0, lacks detailed energy consumption prediction.
P. W. Khan & Y.-C. Byun (2020) [24]	Genetic Algorithm, Hybrid ML	Not specified	MAE: 2.5	No comparison with other state-of-the-art methods, lacks integration with architectural factors.
D. Syed et al. (2021) [25]	Hybrid Deep Learning	Smart building data	MAPE: 5%	Does not incorporate architectural or building form features.
S. Ardabili et al. (2022) [26]	Deep Learning, ML	Various building energy data	Accuracy: 95%	Lacks feature optimization and hyperparameter tuning.

The state-of-the-art (SOTA) methods reviewed above primarily focus on generalized energy prediction without accounting for architectural features and specific residential building types. this studie improves upon these methods by incorporating architectural design, building forms, and advanced hyperparameter tuning using SVR-HGS and CatBoost-HGS models. This approach provides higher accuracy for both lighting and heating energy predictions and addresses the limitations of previous methods regarding feature integration and optimization. The contribution of this study lies in its integration of three key advancements that address limitations observed in prior works. First, unlike many previous studies that relied on either general energy datasets or abstracted variables, this work leverages a highly detailed and diverse dataset generated through parametric simulation, incorporating architectural, thermal, and operational features of residential buildings. Second, it introduces a hybrid modeling framework that couples well-established learners—SVR and CatBoost—with the Hunger Games Search algorithm for advanced hyperparameter optimization. While similar algorithms have been individually explored in the literature, their hybridization within the context of detailed residential energy modeling remains underrepresented. Third, the study offers a comprehensive evaluation across four residential building types, emphasizing not only accuracy but also practical deployment aspects such as runtime and generalizability. This holistic design bridges the gap between model precision and real-world applicability, making the contribution both technically novel and practically significant. In this study, an attempt was made to provide a model for predicting energy consumption in different types of residential buildings by using the capabilities of ML-based methods. The rest of the paper is mentioned as follows: In Section 2, the research methodology, including the introduction of algorithms and hybrid models, was presented. Section 3 introduced the research dataset and studied the effect of different variables with respect to energy consumption. Section 4 discussed the findings of the research in terms of output variables. Finally, Section 5 represents the conclusions.

2 Methodology

This research proposes a hybrid machine learning model to forecast energy consumption for various house building types. The research discussion includes four building styles: bungalows, semi-detached buildings, terraced buildings, and detached buildings. There are 18 input features and two output variables, which are "interior lighting energy" and "total heating energy." Two machine learning algorithms, namely CatBoost and Support Vector Regression (SVR), have been used to develop models with substantial predictability. CatBoost is a gradient-boosting algorithm for decision trees that efficiently handles categorical and ordinal features. Support Vector Regression, also called the extension of SVM for the regression task, is another algorithm that has been used in this research. In order to improve their prediction accuracy, both algorithms are hybridized with a meta-heuristic optimization technique known as Hunger Games Search or, in short, HGS-hence, the two hybrid models: CatBoost-HGS and SVR-HGS. These hybrid models can bring out the best in these primary algorithms by

optimizing and fine-tuning their hyperparameters. The K-fold cross-validation approach, where k=5, will be used to evaluate their performances. Cross-validation involves the division of the dataset into K equal-volume segments. It uses K−1 folds as training data and the remaining fold as test data in each iteration. The training data is used to train the model, and the test data is used for validation. Steps one and two are repeated K times, each time using a different fold as the test data. Lastly, the results of all K iterations are averaged. The choice of k=5 was made as a balance between computational efficiency and the reliability of performance estimation. Smaller values of k (e.g., 2–4) may lead to high variance in evaluation due to fewer training samples per fold, while larger values (e.g., k=10) increase the computational cost significantly, particularly when combined with hyperparameter tuning. Based on prior literature and standard practices in regression tasks, 5-fold cross-validation offers a practical compromise with low bias and acceptable variance. The accuracy of the proposed models is assessed through various evaluation indices, including MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), NRMSE (Normalized Root Mean Squared Error), and RAE (Relative Absolute Error), as detailed in Table 2. This evaluation is performed across different types of residential buildings using a case study to determine the effectiveness of the models [28–31].

Table 2: Statistical evaluation indices

Index	Name	Equation
NRMSE	Normalized Root Mean Square Error	$NRMSE = \frac{\sum_{i=1}^{num}(\Delta_1)^2}{\sum_{i=1}^{num} {y_i}^2}$
MAPE	Mean Absolute Percentage Error	$MAPE = \frac{1}{num}\sum_{i=1}^{num}\left\|\frac{\Delta_1}{y_i}\right\|$
MAE	Mean Absolute Error	$MAE = \frac{\sum_{i=1}^{num}\|\Delta_1\|}{num}$
RAE	Relative Absolute Error	$RAE = \frac{\sum_{i=1}^{num}\|\Delta_1\|}{\sum_{i=1}^{num}\|\Delta_2\|}$

The values in this table are as follows: y_i is the ith observed value, $\hat{y}_i$ is the ith estimated value, and $\bar{y}$ is the mean of observations, no is number of observations, $\Delta_1 = y_i - \hat{y}_i$, $\Delta_2 = y_i - \bar{y}$.

2.1 The CatBoost-HGS Hybrid Model

In this section, more details of the Catboost-HGS hybrid model are discussed, which is a Catboost method optimized by the HGS algorithm. The HGS algorithm attempts to minimize the cross-validation loss (error) by optimizing the hyper-parameter values. The main algorithm for predicting output variables is the Catboost algorithm. Engineers and researchers at Yandex created the gradient-boosting algorithm CatBoost in 2017. It is based on decision trees. Because the CatBoost technique uses symmetric trees, it is incredibly fast in the prediction domain [32]. One factor that makes CatBoost so powerful is its low latency (coverage) requirements, which means it's about eight times faster than XGBoost. The key characteristic of CatBoost is its great stability, demonstrated by the fact that, unlike other ML models, it does not require a large amount of training data and can operate on a wide range of data types [33]. CatBoost's library is open source and available for free. We can list self-driving cars, weather forecasting, and other uses for this method [29]. A few of CatBoost's primary benefits are its higher quality when compared to competing GBDT libraries, its exceptional category inference speed, its support for both numerical and categorical characteristics, and the availability of data visualization tools [34]. Among the various hyperparameters of the CatBoost model, the focus was on optimizing two key parameters, as shown in Table 3, using the Hunger Games Search (HGS) algorithm:

Table 3: Hyperparameter tuning results using HGS

Hyperparameter	Search Range	Final Value (Selected by HGS)
Learning-rate	[0.3, 0.01]	0.08
depth	(integers) [10, 4]	8

These two parameters were selected due to their significant influence on the model's learning dynamics and capacity. The HGS algorithm was applied to explore the defined ranges and identify the combination that minimized the validation loss. The final values identified by the algorithm led to improved model performance and generalization capability. The flowchart of the CatBoost-HGS hybrid model is shown in Fig. 1.

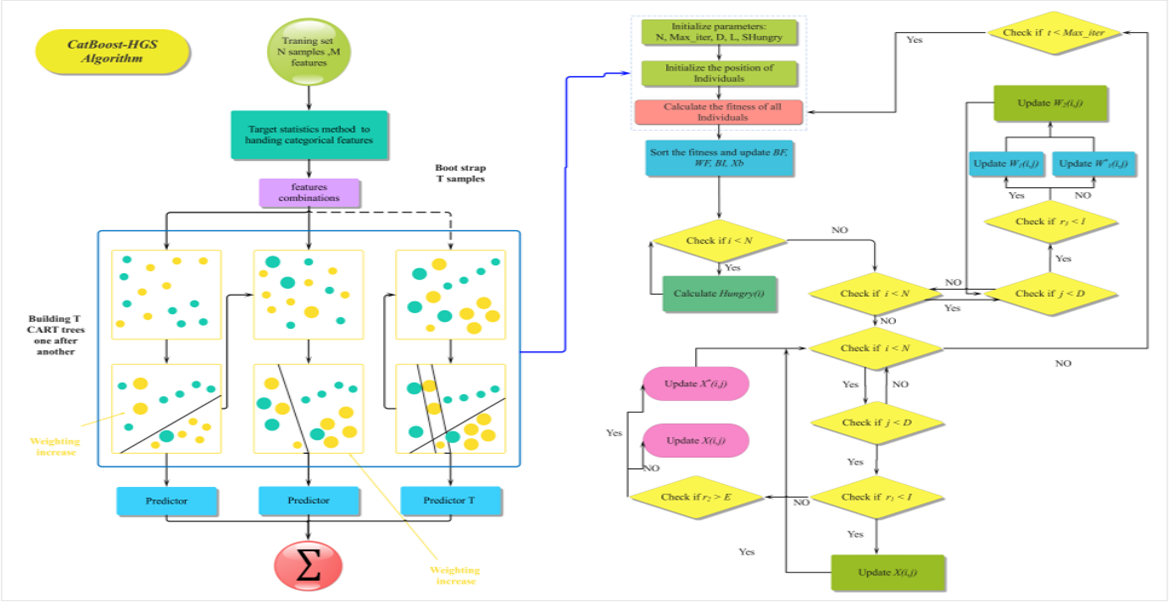


Figure 1: Flowchart of the CatBoost-HGS hybrid model

The training dataset, number of samples (N), and number of features (M) are chosen to initialize the model, as shown in Figure 1. The target statistics (TS) technique is then applied to the categorical features. All of the category qualities were then combined to form new combinations. The next parts will build T regression trees (CART) one by one. In the end, an average of the predictors' values will produce a more accurate model forecast [35]. During network training, the K-Fold cross-validation technique is also used to prevent over-fitting. If the stopping condition allows it, which is frequently the maximum number of iterations, the optimization technique (HGS) is executed each time the loop is restarted.

The initialization of the parameters, the positions of the people, and the fitness values of the individuals are all determined first in the succinct description of the HGS algorithm. Next, the fitness values are sorted to update the parameters. Ultimately, the weight of hungry behavior (W_1 and W_2), as well as the location of each person in the search area, are established by looking at a few conditions and creating random numbers. The method delivers the computed optimal values to the Catboost algorithm and terminates upon the occurrence of the stop condition, which is typically the maximum number of repetitions. Note that a more thorough explanation of the HGS algorithm's governing equations may be found in the next section.

2.2 The SVR-HGS Hybrid Model

In this section, more details of the SVR-HGS hybrid model are discussed, which is a Catboost method optimized by the SVR algorithm. In this model, the main algorithm is to predict the output variables of SVR. An ML method for regression tasks is called SVR. For regression issues, SVR is an extension of SVM. The objective of SVR is to identify a function that roughly represents the mapping between the continuous output variable and the input variables, or features. The algorithm's goal is to minimize the deviation between the actual and anticipated values while maintaining a certain tolerance margin. Labeled training data with associated continuous goal values and input feature pairs is needed for support vector machines (SVR). To make sure the input characteristics have comparable ranges, it is crucial to scale them. Kernel functions are used to convert the SVR input properties into a higher-dimensional space. Sigmoid, polynomial, linear, and radial basis function (RBF) kernels are commonly used. In the converted feature space, the SVR method locates the ideal hyperplane that minimizes error and maximizes margin. Constrained optimization problems must be solved throughout the training phase. The SVR model may be assessed using appropriate metrics after it has been trained. By converting the input characteristics using the learned mapping and applying the regression function, the trained SVR model may make predictions on fresh, unknown data. For regression tasks, SVR is an effective method, particularly when handling intricate, non-linear connections between features and goals. Predicting continuous values is crucial in several fields, including engineering, economics, and finance, where it is commonly employed [36–38]. Fig. 2 shows the flowchart of the SVR-HGS hybrid model. This flowchart illustrates how the HGS algorithm is combined with the main SVR neural network to optimize the SVR's hyper-parameters $(\varepsilon, \gamma, and\ C)$.

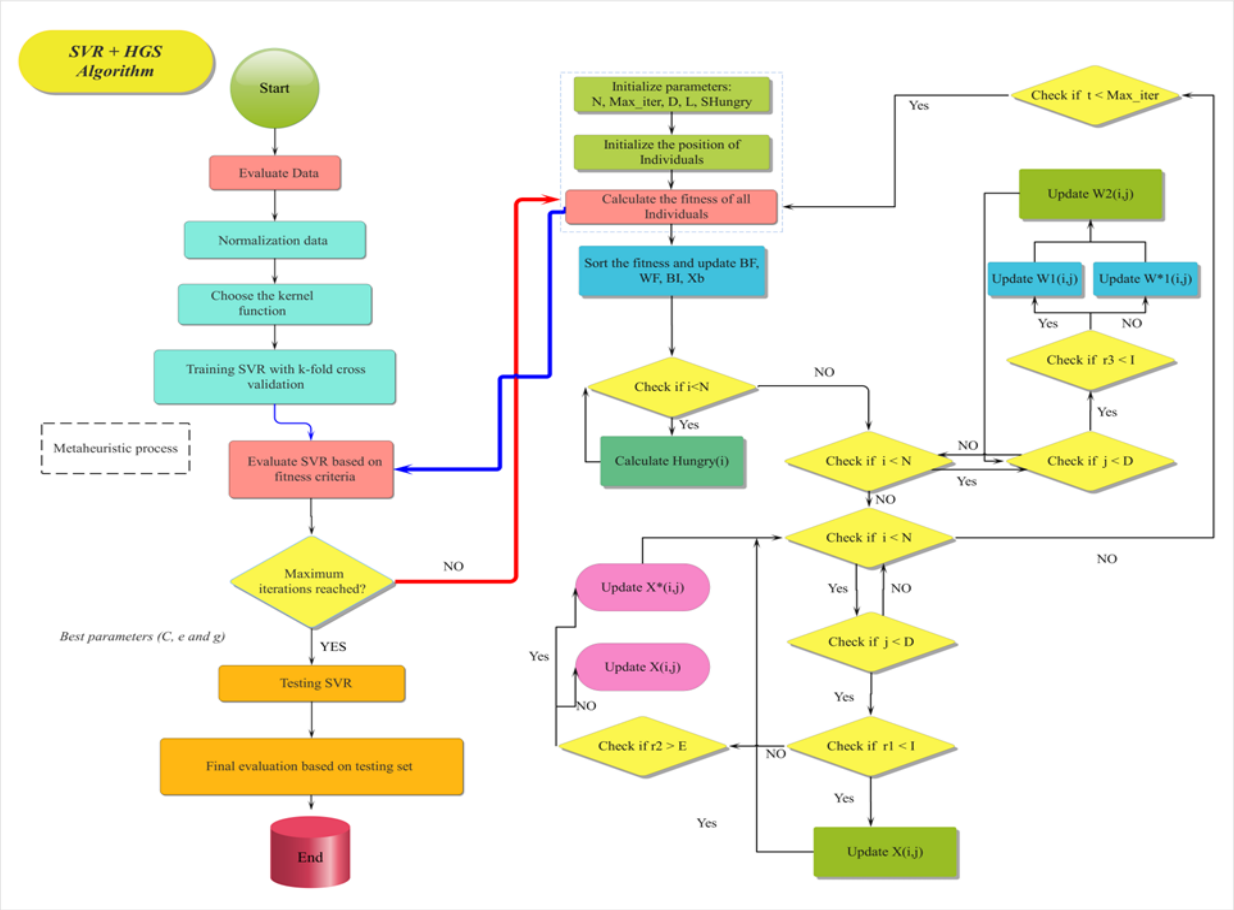


Figure 2: Flowchart of the SVR-HGS hybrid model

The algorithm begins on the left side of the flowchart, corresponding to the SVR algorithm, as shown in Fig. 2. After data normalization, the type of kernel function used will be specified. Several common kernels can be used, including polynomial, linear, and Gaussian kernels, also known as Radial Basis Function (RBF) kernels. For this investigation, the RBF kernel is the default function. The data will then be trained using the K-fold cross-validation process. Next, the HGS algorithm will be used to optimize the hyperparameters of the SVR model, enhancing the training procedure. The role of the HGS algorithm in this context is similar to its role in the CatBoost-HGS hybrid model explained in the previous section. The HGS algorithm will continue to run until the termination condition is met. Once this condition is satisfied, the optimal hyperparameter values will be applied to retrain the SVR model. Finally, statistical indices will be used to calculate the model's fitness quality. This cycle will continue until the specified number of iterations is completed.

In comparison to commonly used metaheuristic algorithms such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), the Hunger Games Search (HGS) algorithm provides a more adaptive and diversity-preserving optimization process. HGS utilizes stochastic hunger-based adjustments to guide the population, reducing the risk of premature convergence often encountered in GA or PSO. Additionally, its structured balance between exploration and exploitation makes it suitable for tuning sensitive models like SVR and CatBoost, where parameter interactions are highly non-linear. These characteristics make HGS particularly effective for achieving global optima in complex learning systems. The hybridization of CatBoost and SVR with Hunger Games Search (HGS) was intended to enhance the hyperparameter tuning process through a more intelligent and adaptive search strategy. Standard techniques such as grid search or random search often suffer from inefficiency or failure to capture complex parameter interactions, particularly in high-dimensional search spaces. By integrating HGS, which uses a dynamic hunger-driven mechanism for population update, the models benefit from better exploration of the search space and convergence towards globally optimal configurations. This combination enables more robust generalization performance, as confirmed by the empirical results.

2.3 Hunger Games Search (HGS) Algorithm

Yang et al. introduced the HGS algorithm in 2021. The main inspiration of this algorithm was the collaborative behavior of animals in nature in order to search for food in times of hunger. When food resources are limited, there will be a logical game between hungry animals to search and find food and win in this situation. The

implementation steps of this algorithm are described as follows [36]:

The first step of this algorithm is to find food. In this step, cooperative communication of animals and food search behavior is expressed using the following equations [39]:

$$\overrightarrow{X(t+1)} = \begin{cases} Game_1 = \overrightarrow{X(t)}.(1+randn(1)) & r_1 < l \\ Game_2 = \overrightarrow{W_1}.\vec{X}_b + \overrightarrow{W_2}.\vec{R}.\left|\vec{X}_b - \overrightarrow{X(t)}\right| & r_1 > l, r_2 > E \\ Game_3 = \overrightarrow{W_1}.\vec{X}_b - \overrightarrow{W_2}.\vec{R}.\left|\vec{X}_b - \overrightarrow{X(t)}\right| & r_1 > l, r_2 < E \end{cases} \quad (1)$$

In these equations, r_1 and r_2 represent random numbers between of 0 and 1, $randn(1)$ represents random normal distribution, and the value of t represents the current iteration, $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$ express the weight of hunger behavior, parameters $\vec{X}_b$ and $\overrightarrow{X(t)}$ indicate the position of the best individual and each individual at iteration t, respectively, $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$ express the weight of hunger behavior, the l variable is designed to improve the algorithm, and $\vec{R}$ is a variable in the range of a and -a , which is defined as the following equation [40]:

$$\vec{R} = 2 \times \text{shrink} \times rand - \text{shrink} \quad (2)$$

$$\text{shrink} = 2 \times \left(1 - \frac{t}{T}\right) \quad (3)$$

T is the maximum number of iterations, while rand stands for random values between 0 and 1. The equation for variable E, a change control in all circumstances, is shown below [41]:

$$S = hyp.func(|fitness_i - fitness_{best}|) \quad (4)$$

Where hype.func refers to hyperbolic function, $fitness_{best}$ refers to the best fitness value, and $fitness_i$ represents the obtained fitness in the current iteration. The HGS algorithm's search procedure and reasoning during optimization based on Eq. (1) are depicted in Fig. 3.

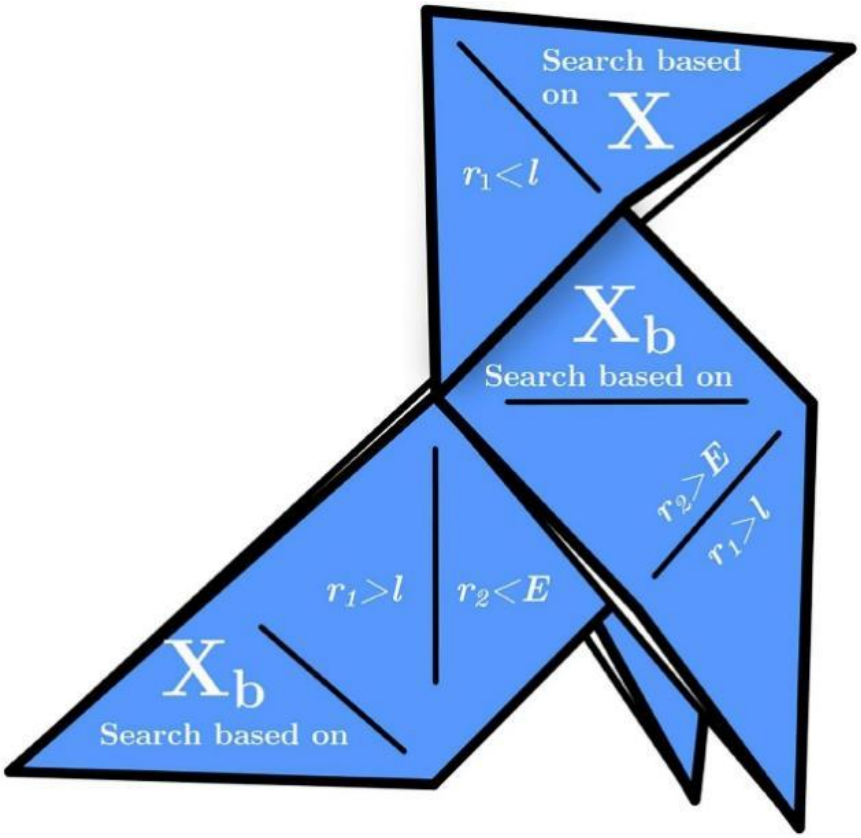


Figure 3: Logic of HGS algorithm during optimization

In the second step of this algorithm, which involves the role of hunger, the characteristics of the members' hunger coefficient in the search are explained by $\overrightarrow{W_1}$ and $\overrightarrow{W_2}$, whose equations are expressed as follows [36]:

$$\overrightarrow{W_1} = \begin{cases} \delta(i).\frac{N_{pop}}{\sum \delta} \times r_4 & r_3 < l \\ 1 & r_3 > l \end{cases} \quad (5)$$

$$\overrightarrow{W_2} = (1 - \exp\left(-\left|\delta(i) - \sum \delta\right|\right)) \times r_5 \times 2 \quad (6)$$

Where the number of individuals is indicated by the parameter N_{pop}, r_3 , r_4, r_5, and r_6 are parameters that create random numbers between 0 and 1. The hunger of every individual is represented by δ(i), which may be computed using the equation [37]:

$$\delta(i) = \begin{cases} 0 & fitness_i == fitness_{best} \\ \delta(i) + \delta_{new} & fitness_i != fitness_{best} \end{cases} \quad (7)$$

Where δ_{new} shows a new hunger, which is defined as the following equation [38]:

$$\delta_{new} = \begin{cases} \delta_{largest} \times (1 + r) & \delta_{temp} < \delta_{largest} \\ \delta_{temp} & \delta_{temp} \geq \delta_{largest} \end{cases} \quad (8)$$

$$\delta_{temp} = \frac{fitness_i - fitness_{best}}{fitness_{worst} - fitness_{best}} \times r_6 \times 2 \times (bound_{upper} - bound_{lower}) \quad (9)$$

Where $fitness_{\text{worst}}$ denotes the lowest fitness attained in the current iteration, $bound_{upper}$ and $bound_{lower}$ represent the higher and lower boundaries of the feature space, respectively, r_6 is parameters that create random numbers between 0 and 1, and $\delta_{largest}$ is designed to improve the performance of the algorithm [38].

The integration of Hunger Games Search (HGS) with SVR and CatBoost optimizes hyperparameters by using a population-based search mechanism. HGS adjusts key parameters such as the regularization parameter (C) and epsilon for SVR, and learning rate, number of trees, and depth for CatBoost. The algorithm iteratively refines candidate solutions based on performance, improving both models' accuracy by identifying optimal parameter sets. This method provides a more efficient search process compared to traditional methods like grid or random search.

All initialization settings used in the optimization algorithm including population size, number of iterations, and random seed along with the model configuration parameters, have been clearly specified. These implementation details are presented in Table 4 and further described.

Table 4: Configuration Details of the HGS Algorithm and CatBoost Model

Component	Parameter	Value	Description
HGS Algorithm	Population size	30	Number of individuals in the search population
	Max iterations	100	Total optimization steps
	Random seed	42	Ensures reproducibility of search path
	Initialization range	[0, 1]	Normalized range for initializing solutions
	Objective function	Validation loss	Metric to minimize during tuning
CatBoost Model	Iterations	600	Number of boosting rounds
	Learning rate	0.08	Step size shrinkage
	Depth	8	Maximum tree depth
	Loss function	RMSE	Used for regression task
	Eval metric	RMSE	Used for early stopping and comparison
	Early stopping rounds	50	Stops training if no improvement after 50 rounds
	Random seed	42	Controls randomness in model training

3 Description of dataset

The dataset used in this study comprises over a million records of Irish residential buildings, including terraced houses (type 1), detached houses (type 2), bungalows (type 3), and semi-detached houses (type 4). The four building types selected for this study terraced houses, detached houses, bungalows, and semi-detached houses were chosen based on their representation of common residential structures with varying energy consumption characteristics. These building types are widely observed in global energy studies, as they cover a broad spectrum of architectural designs and thermal dynamics. For example, detached houses are often used in energy consumption studies due to their energy performance variability, while terraced houses are frequently studied in urban settings due to their dense arrangement and shared walls. This diversity allows for a more comprehensive analysis of energy consumption patterns across different residential configurations. Furthermore, the choice of these building types aligns with existing literature, which highlights their significance in residential energy usage across various climates and geographical regions. Figure 4 illustrates the geometric models of these various building types. The dataset includes 18 input features and two output features: "Interior Lighting Energy" and "Total Heating Energy." It integrates EnergyPlus for thermal simulation with jEPlus, a parametric tool for physics-based simulations, and incorporates DesignBuilder building templates for generation. DesignBuilder was utilized to create baseline architectural models, which were then exported to EnergyPlus-compatible IDF files. These were linked with jEPlus to run batch simulations across varying feature values. Assumptions included the use of standard occupancy schedules and fixed weather files per region, while HVAC templates were selected from pre-validated DesignBuilder libraries. One limitation is that jEPlus assumes linear parameter variation between specified bounds, which may not fully capture nonlinear effects of design changes. Additionally, while DesignBuilder automates geometry export, it may constrain fine-tuning of certain envelope properties. Additionally, the dataset is geographically specific, representing only Irish building types and climate conditions. geographic limitation may introduce biases, as energy consumption behaviors and architectural features can vary significantly across different regions and climates. While the models performed well within the context of Irish buildings, their generalizability to other geographic locations or climates may require further validation with more diverse datasets. The dataset encompasses U-values for walls, roofs, floors, doors, and windows, as well as data on building fabric qualities, HVAC systems, and other components. Additionally, it

includes metrics for hot water energy demand, solar systems, equipment, lighting, and heating.



Figure 4: The geometric model of different residential building types

To investigate the effect of different features on energy consumption across various building types, Figures 5-8 illustrate the correlation between the research variables for each building type separately. In all building types, a positive correlation exists between the insulation values (U-Values) for floors, doors, roofs, windows, and walls and both output variables. This indicates that as the U-values increase, which signifies poorer insulation quality, both "Interior Lighting Energy" and "Total Heating Energy" also rise. Correlation values between HVAC Efficiency and output variables are negative across all building types. This suggests that increasing HVAC efficiency is likely to significantly reduce energy consumption, as effective heating and cooling systems have a major impact on energy use in residential buildings. The positive correlation values found between Domestic Hot Water Usage, Air Change Rate, and the output variables across all building types indicate that these factors have a notable impact on energy consumption. Additionally, there are positive correlations between Building Orientation, Equipment Density, and Heating Setpoint Temperature, and the output variables, reflecting a direct relationship between these factors and energy consumption. In contrast, the correlation values between the Weather File, Occupancy Level, and Window-to-Wall Ratio with the output variables are close to zero for all building types, suggesting no significant correlation. Furthermore, the correlation values between Lighting Density and Heating Setback Temperature with the output variables are positive in all building types, indicating a direct relationship between these factors and energy consumption.

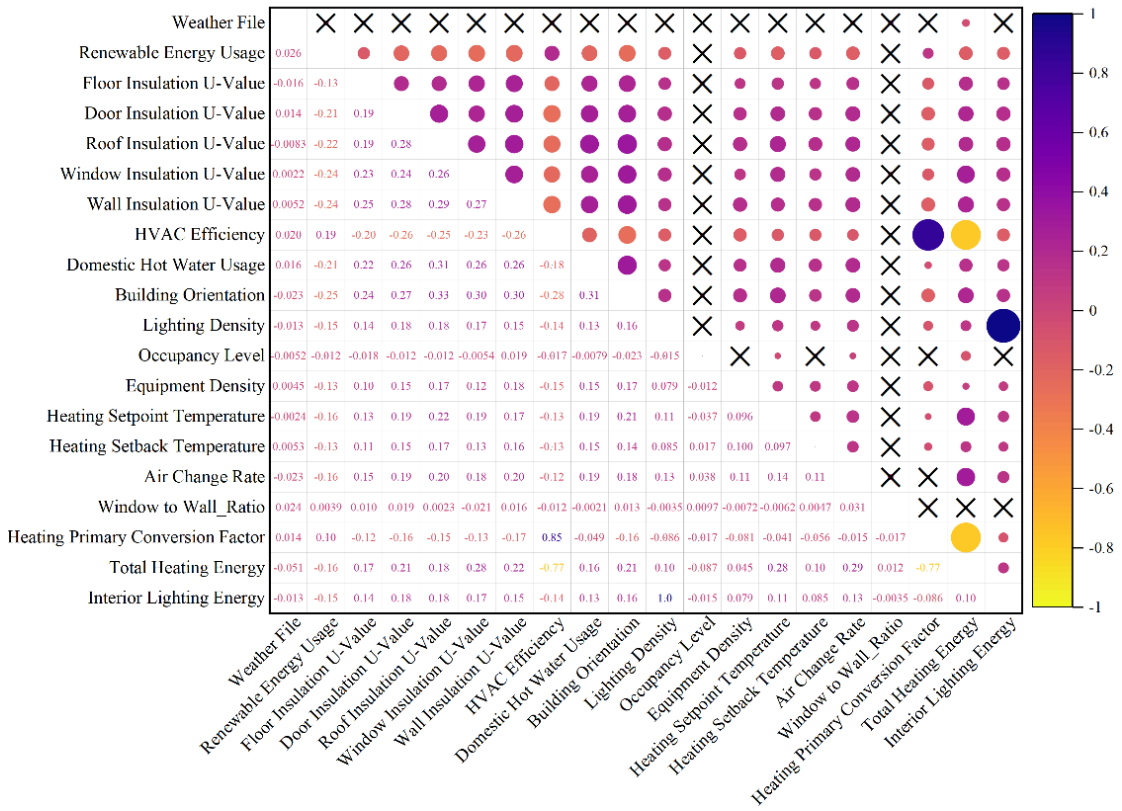


Figure 5: Correlation between research variables in type 1 building

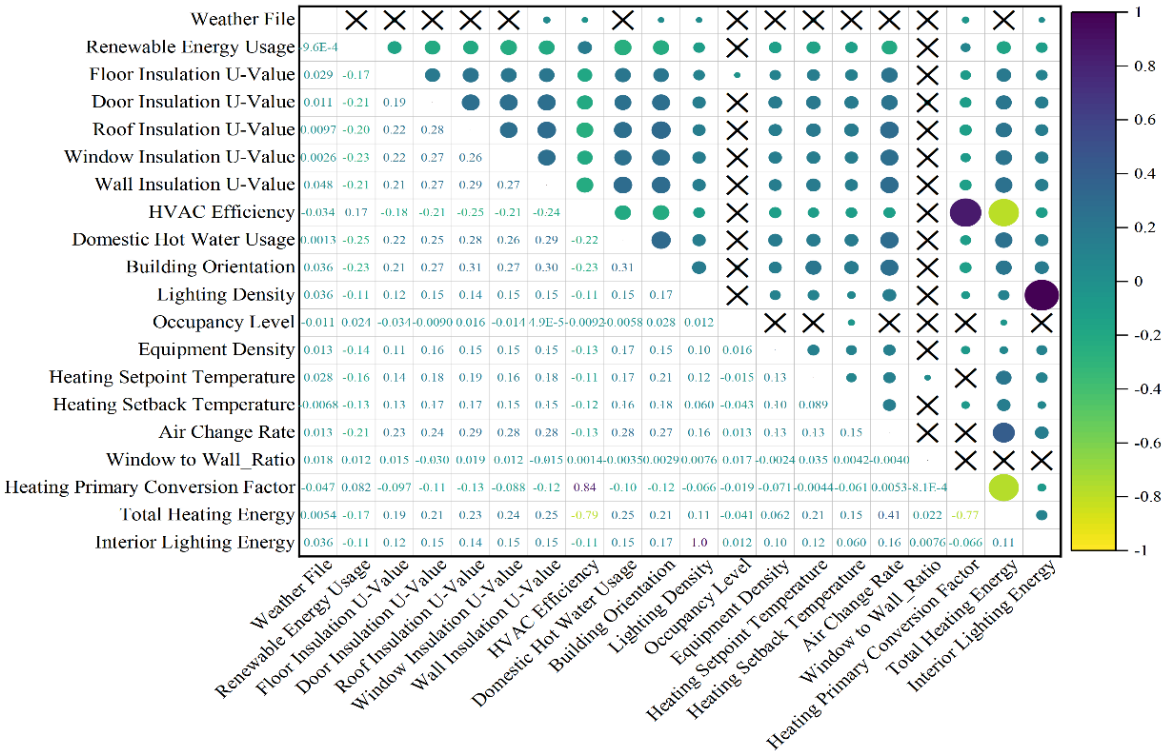


Figure 6: Correlation between research variables in type 2 building

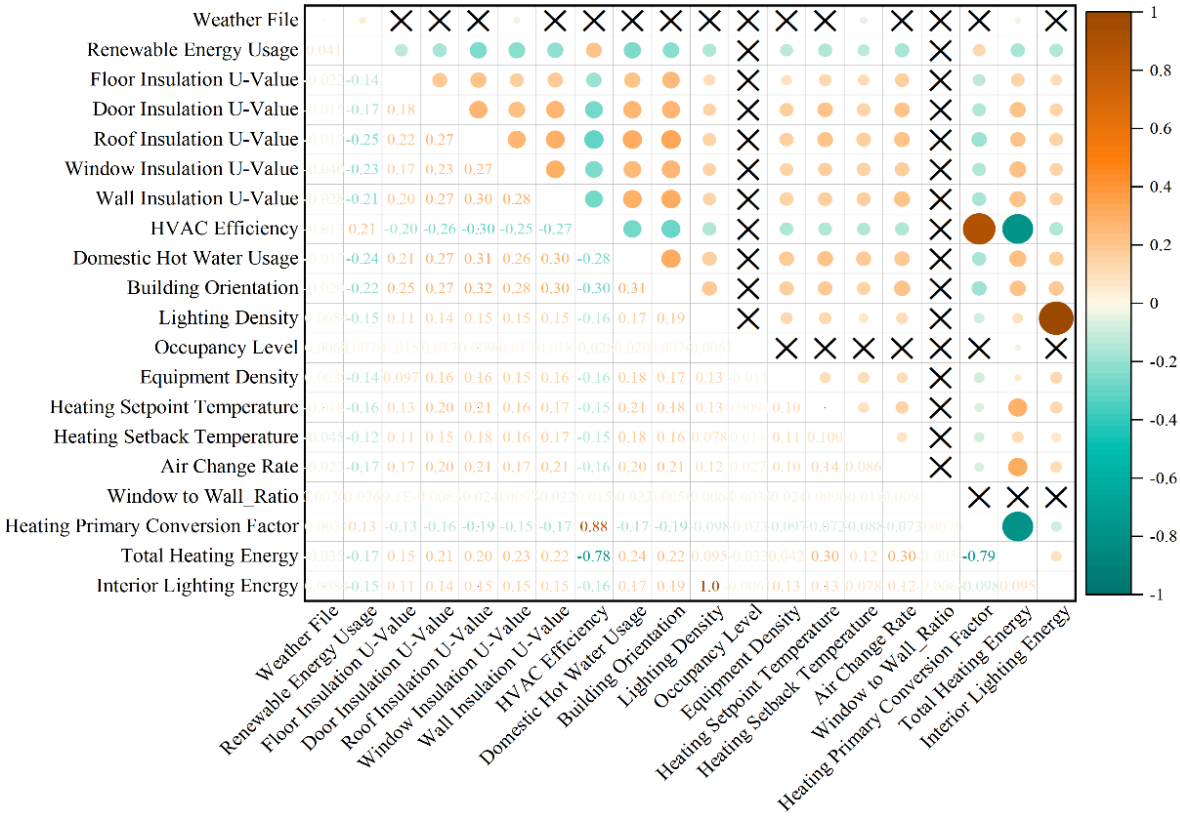


Figure 7: Correlation between research variables in type 3 building

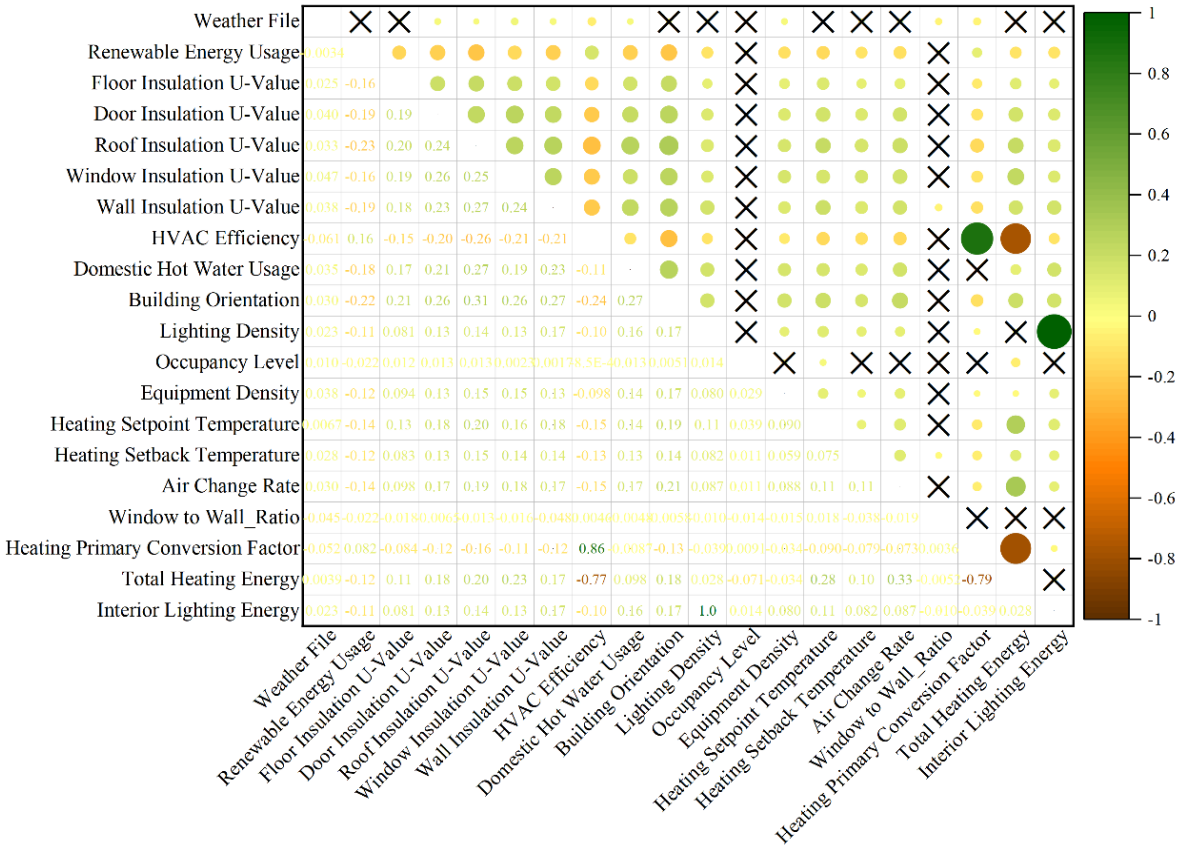


Figure 8: Correlation between research variables in type 4 building

The analysis of the correlation results presented in Figures 5 to 8 reveals that certain variables, such as HVAC efficiency, exhibit a strong correlation with energy consumption, particularly heating energy. This strong correlation is expected, as HVAC systems play a significant role in regulating building temperatures and, therefore, energy consumption. On the other hand, variables like weather file and building orientation show weaker correlations with energy consumption. This can be attributed to the relatively stable nature of these factors over time, particularly in fixed climates or locations, which reduces their impact on energy consumption compared to other dynamic variables, such as heating setpoint temperature or air change rate. These findings underscore the importance of carefully selecting and integrating relevant features for energy consumption prediction. The choice of architectural features, as well as hyperparameter tuning, contributes to the observed differences in model performance. The incorporation of HVAC efficiency and heating-related parameters into our models helps capture the complexities of energy consumption more accurately, as opposed to simpler models that do not account for these factors. Although all 18 input features were initially included in the model, their respective correlations with the target energy variables were further analyzed to assess their actual contribution to prediction performance. As illustrated in Figures 5–8, some features such as Weather File, Occupancy Level, and Window-to-Wall Ratio exhibited low correlation with both interior lighting and heating energy consumption. To evaluate their influence, additional ablation tests were conducted, revealing that the exclusion of these low-correlation variables did not significantly degrade the model's accuracy. However, given their potential contextual importance in specific scenarios and to maintain generalizability across diverse building configurations, these features were retained in the final model. This approach balances model comprehensiveness with complexity, ensuring robustness without overfitting to specific input patterns. These insights highlight the contributions of this study to improving energy consumption predictions by addressing the limitations of current methods in the literature.

3.1 Data preprocessing

In order to prepare the dataset for model training and ensure that it was clean and suitable for the algorithms, several preprocessing steps were applied. These steps were designed to handle missing values, scale the data, and account for the different characteristics of the variables, ensuring compatibility across models and comparison with baseline methods. Instances with missing target values were removed, and for numerical features with missing values, mean imputation was applied. Categorical features with missing values were treated as a separate category ("Missing") to preserve useful information. Although CatBoost handles categorical variables and does not require explicit normalization of numerical features, Min-Max scaling was applied to numerical inputs to improve the performance of other algorithms used for baseline comparisons. CatBoost also performs internal encoding of categorical features based on target statistics, so manual encoding was not required. Outliers were detected and removed based on domain knowledge, with values exceeding the 99th percentile in selected features being capped. These preprocessing steps ensured the dataset was clean, consistent, and ready for training, while maintaining comparability across algorithms.

All preprocessing and modeling tasks were implemented in Python 3.9.13. The experiments were conducted on a system equipped with an Intel Core i7-11700K CPU, 32 GB of RAM, and an NVIDIA RTX 3080 GPU. The following libraries were used: CatBoost 1.2.2, Scikit-learn 1.1.3, and NumPy 1.23.5.

4 Results

In this section, the accuracy of various models is evaluated by isolating the output variables, such as "Interior Lighting Energy" and "Total Heating Energy," through a case study based on the introduced dataset.

4.1 The results related to the prediction of the interior lighting energy variable

Based on the training and test datasets, Fig. 9 displays the time series of errors and predicted values for the "Interior Lighting Energy" variable by building type. The figure illustrates that, compared to the CatBoost-HGS model, the time series predicted by the SVR-HGS model align more closely with the observed time series. Additionally, in all building types, the error time series for the SVR-HGS model exhibits a smaller range of variability compared to the CatBoost-HGS model. This further confirms the superior accuracy of the SVR-HGS model in predicting the "Interior Lighting Energy" variable.

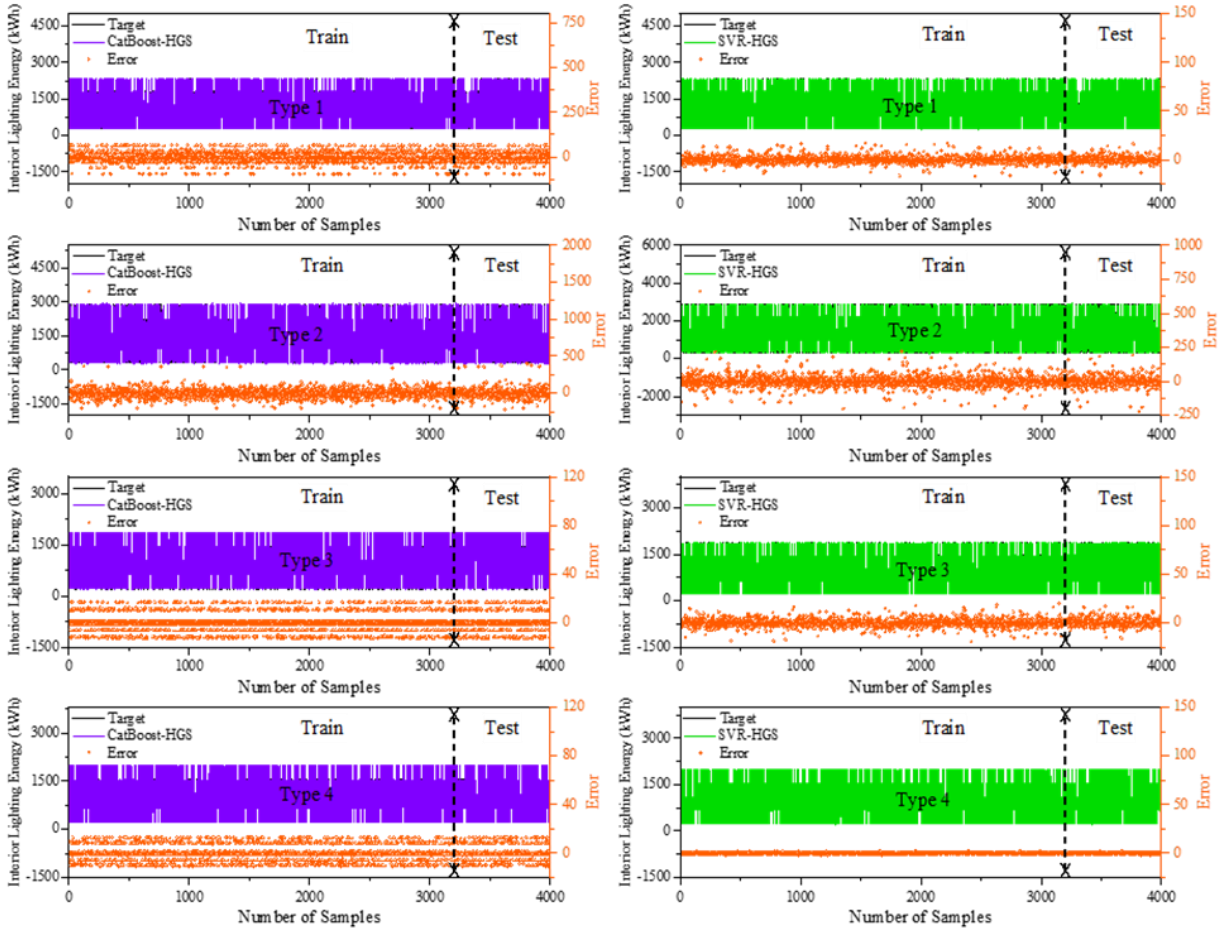


Figure 9: Time series of error and predicted values of the Interior Lighting Energy variable

Based on the "Interior Lighting Energy" variable, Fig. 10 displays box plots of the error values for all models within 1.5 IQR in two subgraphs for the test and training datasets. The figure shows that, across all building types and for both training and test datasets, the SVR-HGS model exhibits a lower error range within 1.5 IQR and fewer outliers compared to the CatBoost-HGS model. This indicates less dispersion of errors and a smaller standard deviation for the SVR-HGS model, highlighting its superior performance relative to the CatBoost-HGS model.

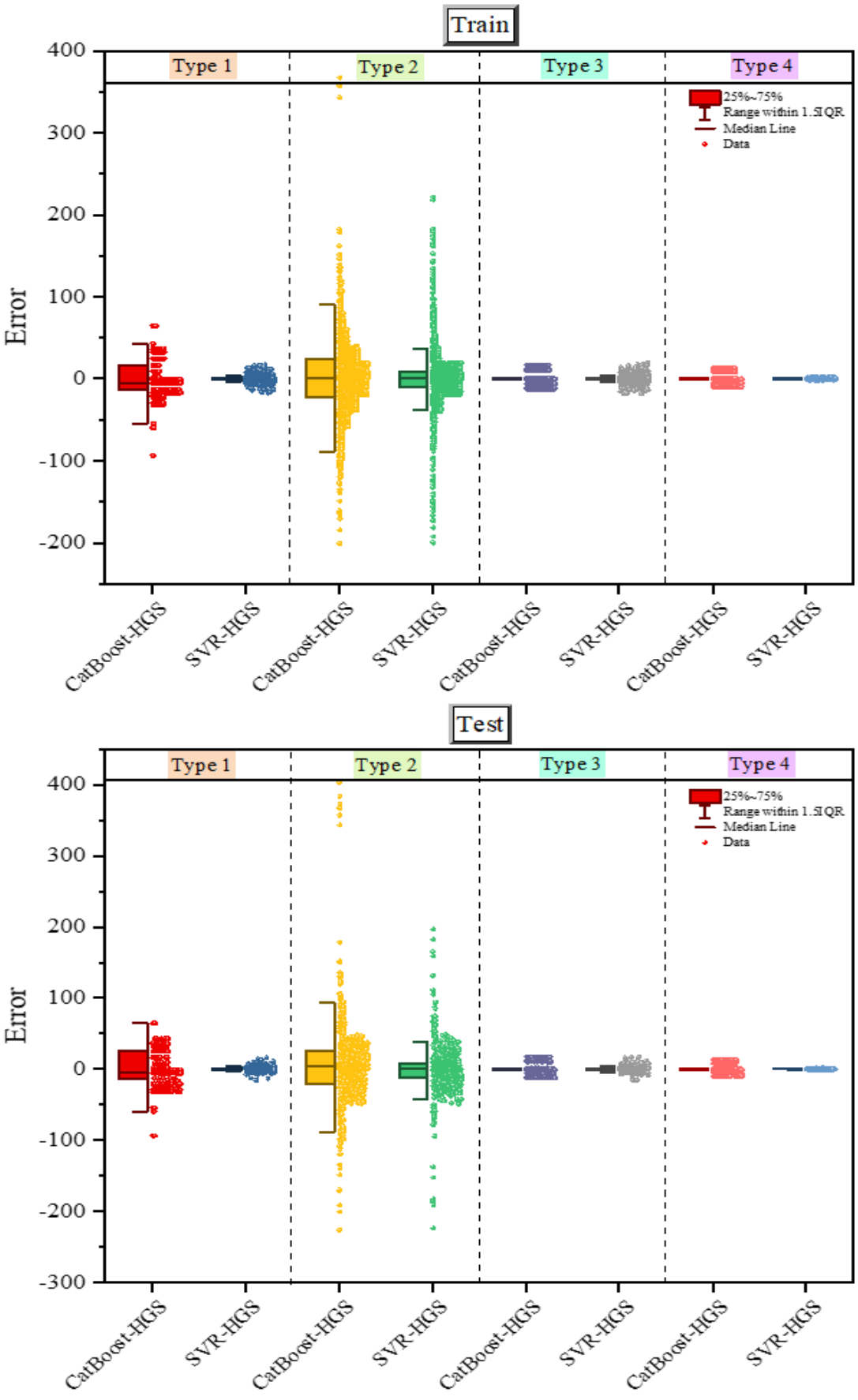


Figure 10: Box plots of error values for all models

The probability density function (PDF), frequency histogram, and time series of the errors (residuals) for the "Interior Lighting Energy" variable are displayed in Fig. 11. The time series of errors indicates that the amplitude of error variations associated with the CatBoost-HGS model is larger compared to the SVR-HGS model across all building types. As a result, the CatBoost-HGS model is expected to perform less well in predicting the Interior Lighting Energy variable. Additionally, the figure shows that the error PDF for the SVR-HGS model has shorter tails and a smaller area compared to the CatBoost-HGS model, indicating a lower standard deviation of errors for the SVR-HGS model.

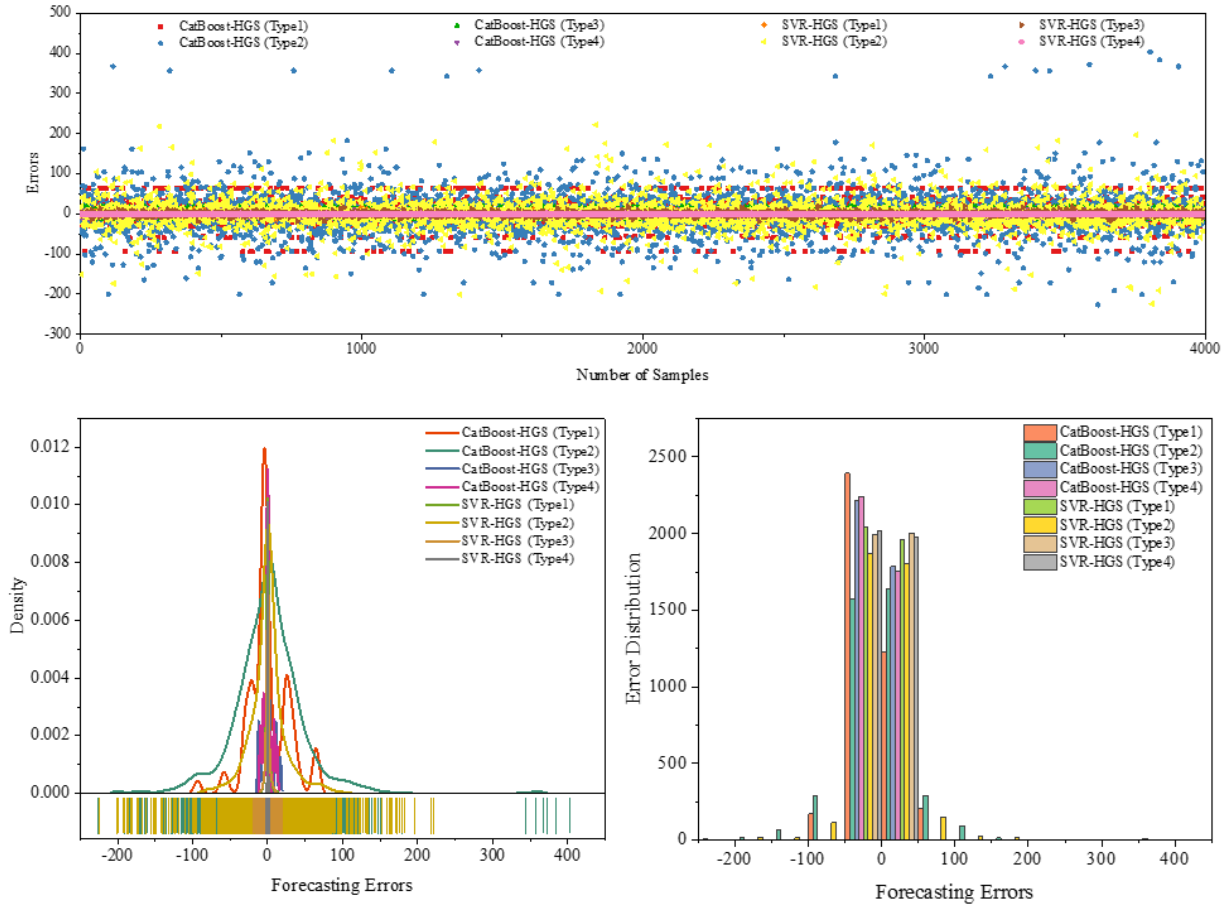


Figure 11. Comparison of the error values of models related to interior lighting energy

The scatter plots of the Interior Lighting Energy variable's observed and predicted values, categorized by building type and based on training and test datasets, are displayed in Fig. 12. These plots show the predicted values on the vertical axis and the observed values on the horizontal axis. According to the figure, across all building types and for both training and test datasets, the SVR-HGS model consistently has higher R-squared values compared to the CatBoost-HGS model. Therefore, the SVR-HGS model is more accurate than the CatBoost-HGS model in predicting the Interior Lighting Energy variable.

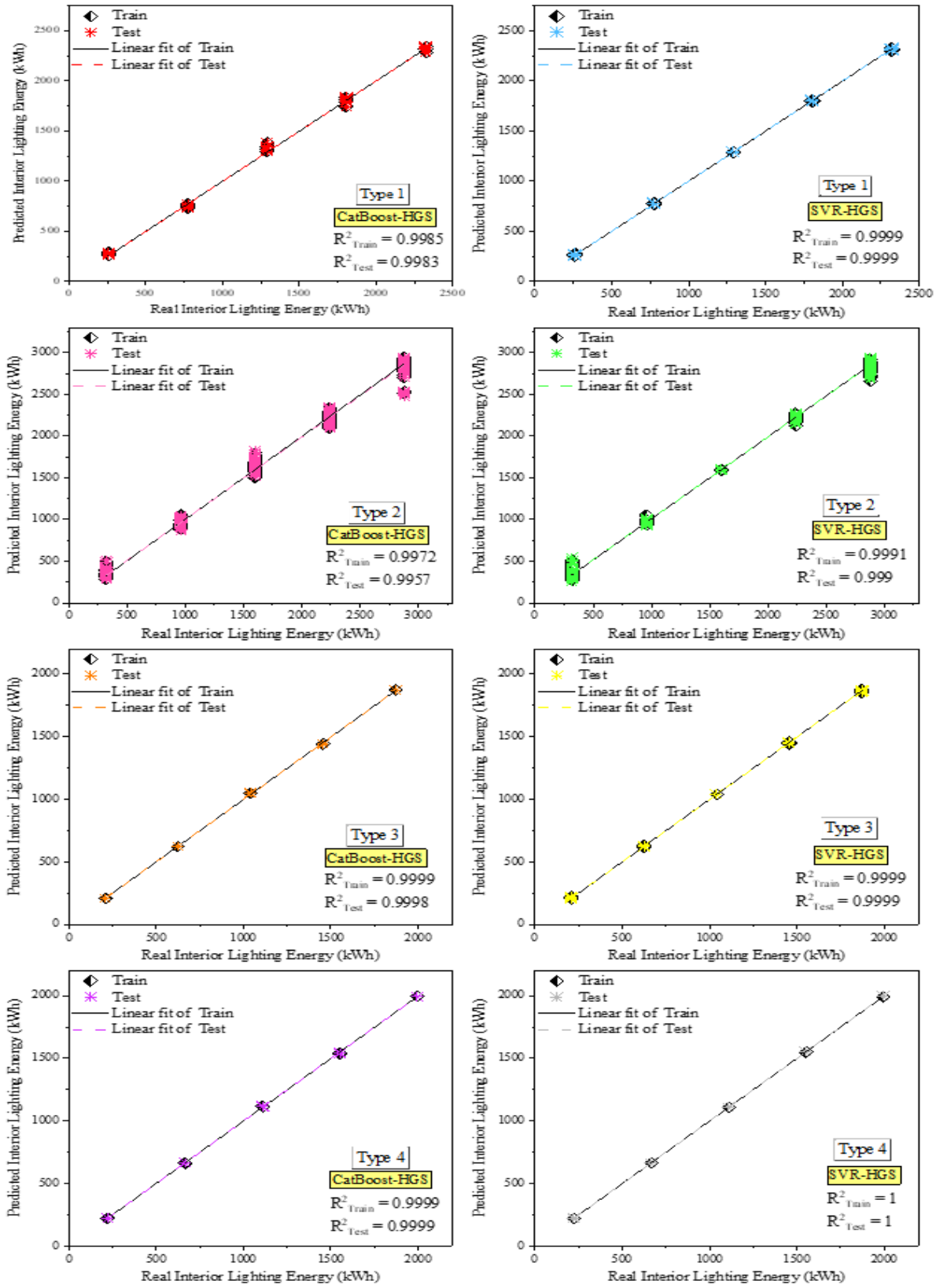


Figure 12: The observation-prediction plots related to the Interior Lighting Energy variable

In the following section, various evaluation indices are examined to make a quantitative comparison between different models. Figure 13 displays the values of these evaluation indices for both models based on the Interior Lighting Energy variable, separated by building type. According to the figure, the SVR-HGS model consistently shows lower MAE, MAPE, NRMSE, and RAE values, and a higher R^2 value, compared to the CatBoost-HGS model across all building types. This indicates that the SVR-HGS model is more accurate in predicting the

Interior Lighting Energy variable. Additionally, when comparing the accuracy of the SVR-HGS model across building types, it is observed that buildings of types 1, 3, and 4 have higher Interior Lighting Energy predictions than type 2 buildings.

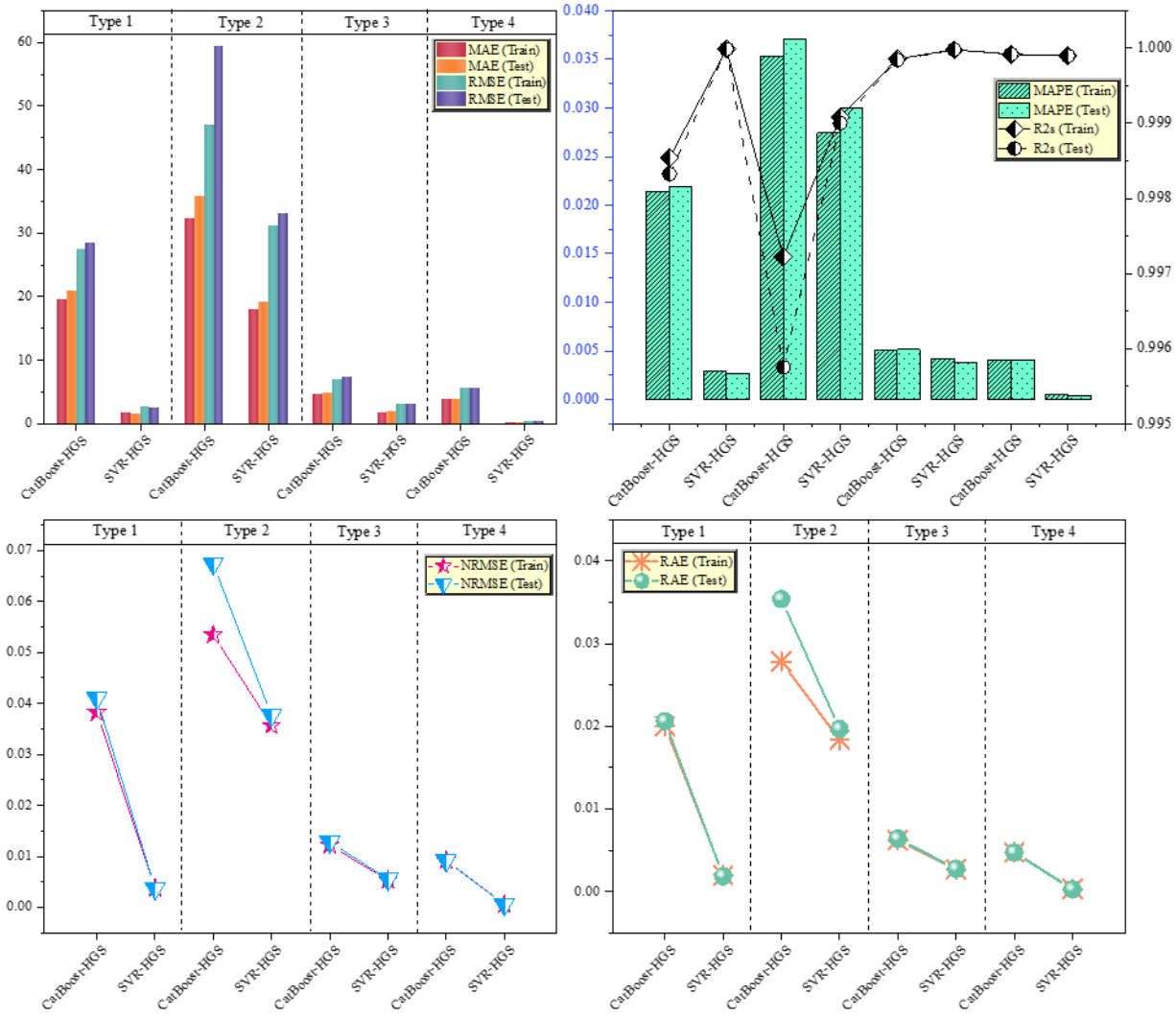


Figure 13: Performance of models based on the evaluation indices

Figures 14 and 15 present the Taylor diagrams for all models, displayed separately for training and testing datasets. These diagrams provide a graphical representation of three statistics: Pearson's correlation coefficient, root mean square error (RMSE), and standard deviation. In these diagrams, the distance between the points corresponding to each model and the point labeled "Interior Lighting Energy" on the horizontal axis measures how well the models reproduce the observations. This statistic indicates that the SVR-HGS model aligns more closely with the "Interior Lighting Energy" point. Based on Figures 14 and 15, it is evident that, in both the training and test datasets and across all building types, the SVR-HGS model exhibits the best performance in reproducing its observations. the Taylor diagrams in Figures 14 and 15 reveal partial visual overlap in the performance of SVR-HGS and CatBoost-HGS for lighting energy prediction. This overlap arises because Taylor diagrams incorporate multiple dimensions—correlation, standard deviation, and centered RMSE—into a single visual representation. When two models share similar correlation and variance, but differ in absolute or percentage-based errors, these nuances may not be clearly reflected in the diagram. Thus, while these plots are valuable for summarizing general trends, they must be interpreted alongside detailed numerical metrics for accurate assessment.

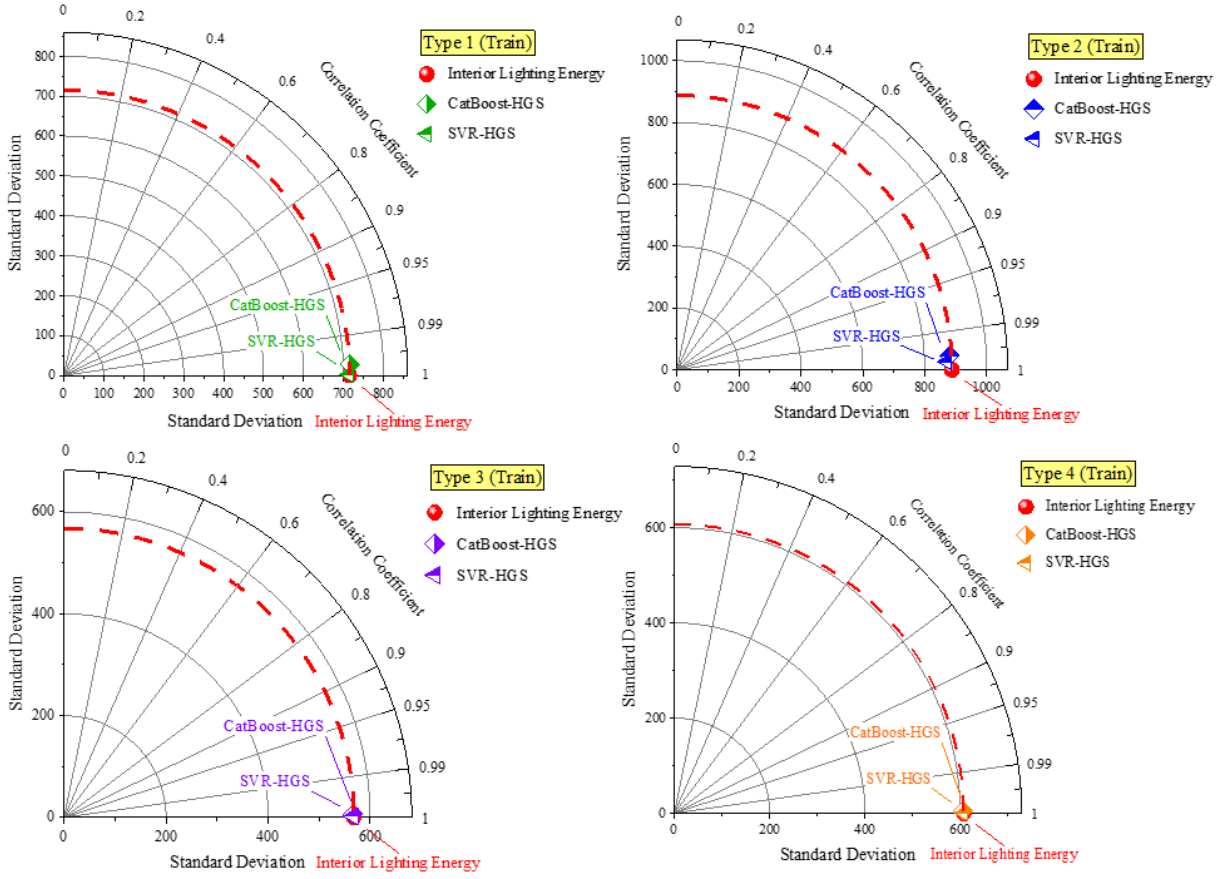


Figure 14: The Taylor diagram related to models based on the train dataset

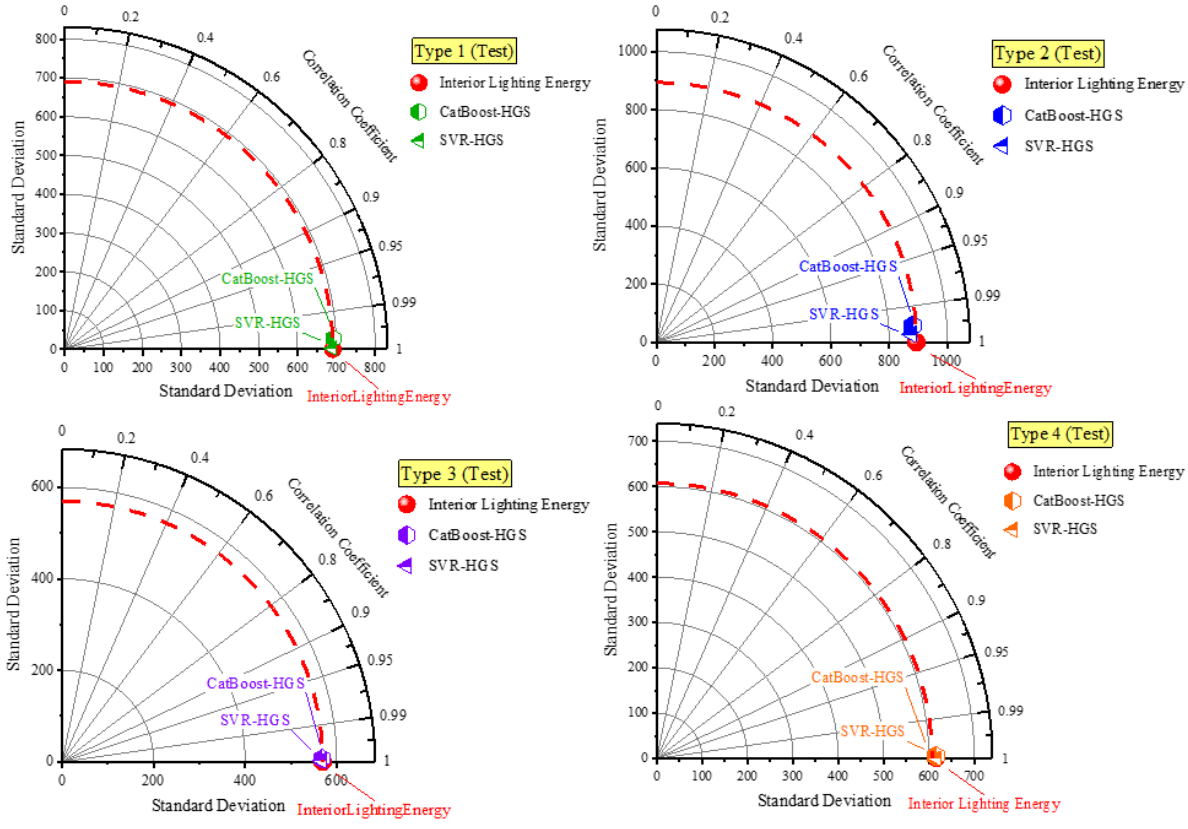


Figure 15: The Taylor diagram related to models based on the test dataset

4.2 The results related to the prediction of the total heating energy variable

The time series of error and estimated values for the variable Total Heating Energy by building type, as projected by the models, are displayed in Fig. 16. This figure shows that, in comparison to the SVR-HGS model, the time series predicted by the CatBoost-HGS model fit the observed time series more closely. It also appears that in all types of buildings, the error time series related to the SVR-HGS model has a larger range of changes than the CatBoost-HGS model. This suggests that, for forecasting the Total Heating Energy variable, the CatBoost-HGS model should have less error than the SVR-HGS model.

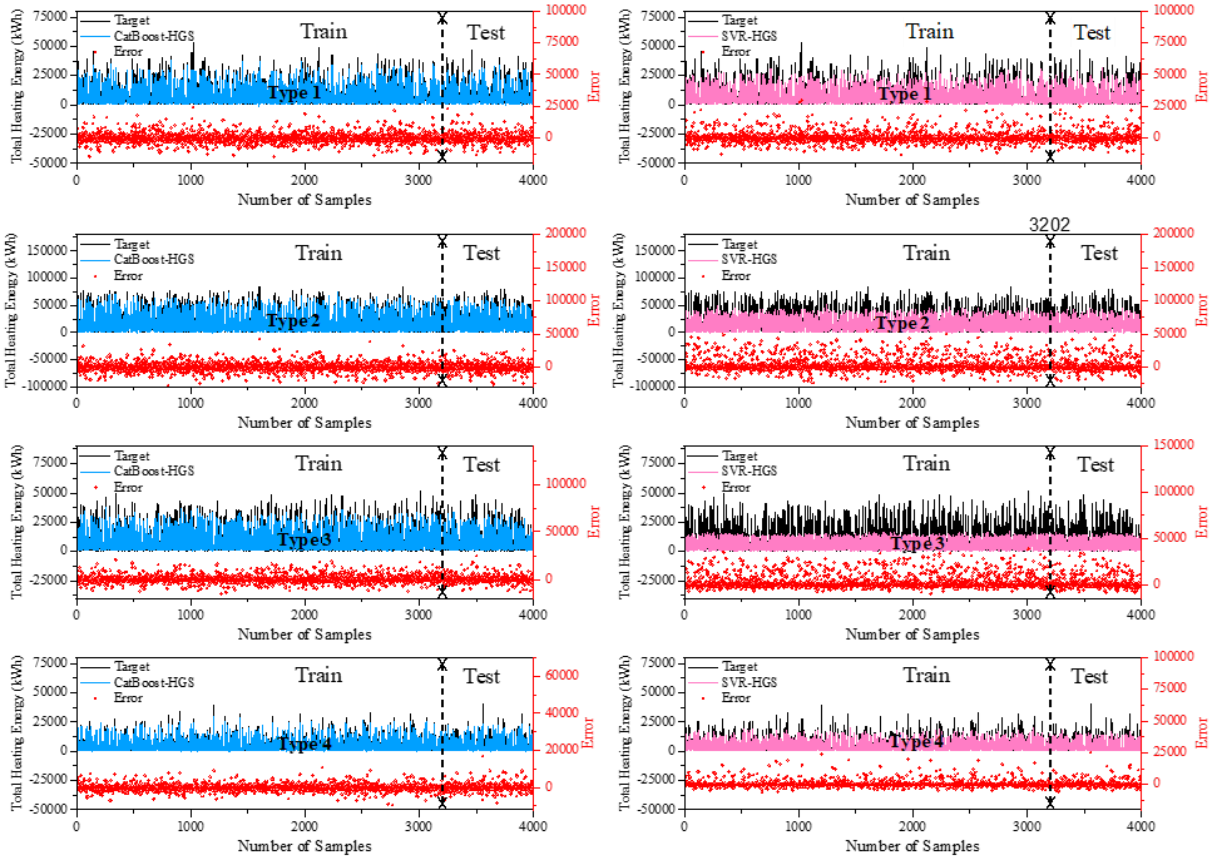


Figure 16: Time series of error and predicted values for the variable total heating energy

Based on the Total Heating Energy variable, Fig. 17 displays box plots of the error values for all models within 1.5 IQR in two subgraphs of the test and train datasets. According to this figure, it can be seen that, for all types of buildings and across both training and test datasets, the CatBoost-HGS model has lower error values within 1.5 IQR and shorter outliers compared to the SVR-HGS model. This indicates that the CatBoost-HGS model is more accurate than the SVR-HGS model, as it has lower error dispersion and a smaller standard deviation.

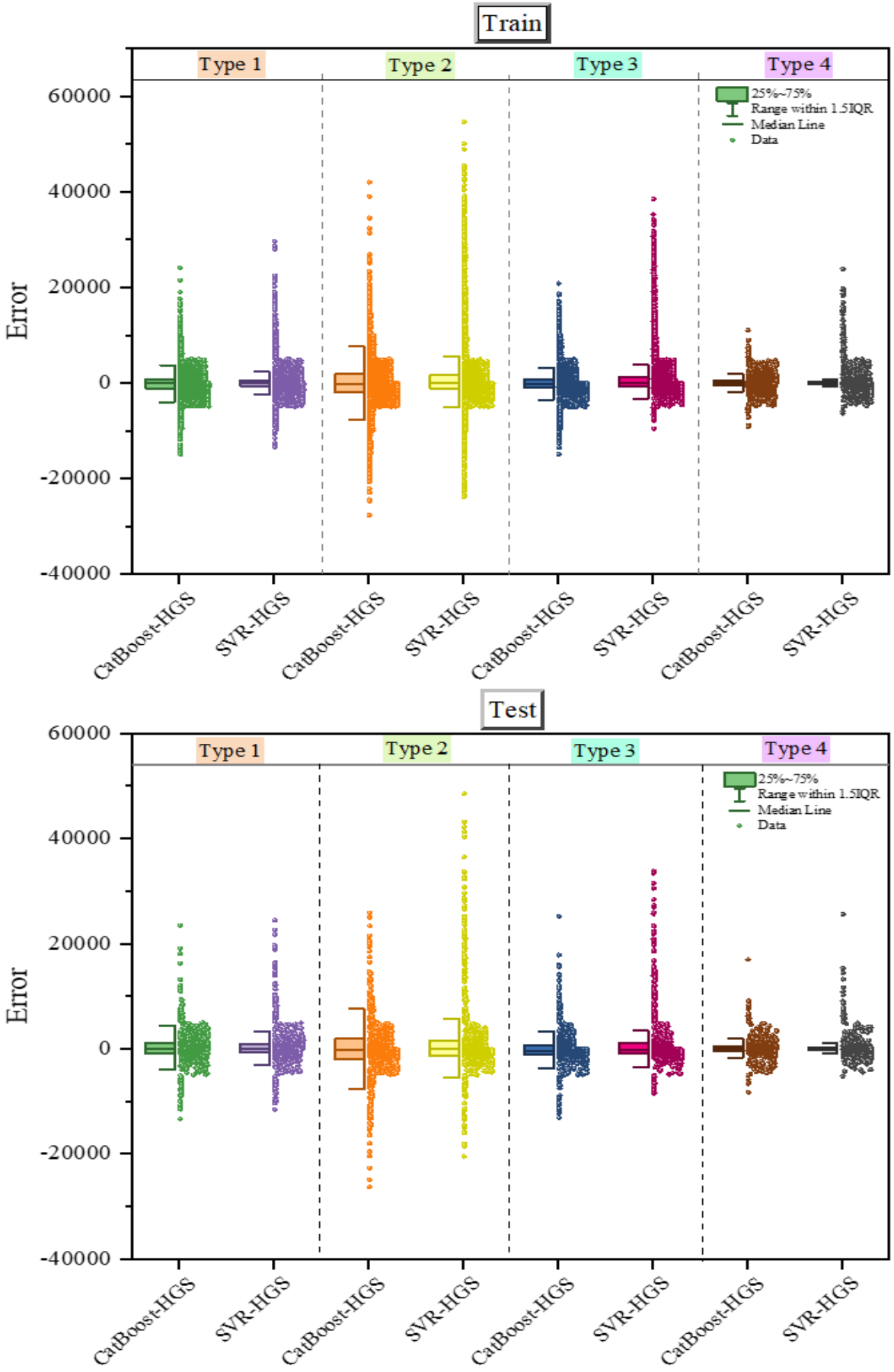


Figure 17: Box plots of error values related to the Total Heating Energy variable

Fig. 18 presents PDF, frequency histogram and time series of the errors (residuals) related to Total Heating Energy variable. The Time series of error shows that variation in amplitude of the error values corresponding to SVR-HGS model are higher compared to CatBoost-HGS model for all the building types. This may indicate that the CatBoost-HGS model will offer a better forecast of the Total Heating Energy variable. On the other hand, from this figure, one can observe that for each category of buildings, the error PDF of the CatBoost-HGS model has shorter outliers with smaller area compared to those from the SVR-HGS model. It will also confirm that the CatBoost-HGS model error standard deviation is smaller than in the SVR-HGS model.

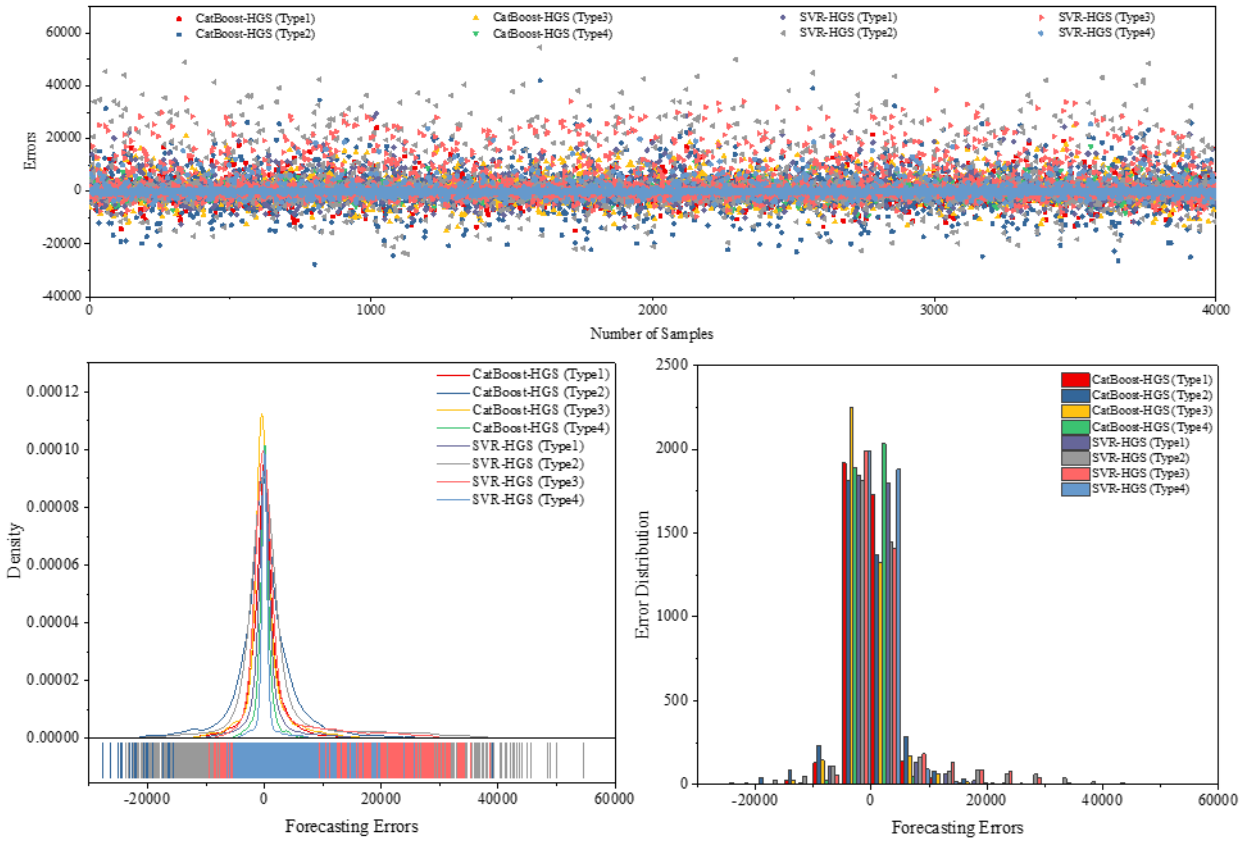


Figure 18: Comparison of error values for models related to total heating energy

Fig. 19 presents the scatter plots of the observed vs. the predicted values of the variable Total Heating Energy, by building type and for training and test datasets, respectively. This figure also shows that, for all the considered building types, the CatBoost-HGS model develops higher R-squared values for both training and test datasets compared to the SVR-HGS model.

Therefore, the CatBoost-HGS model is more accurate than the SVR-HGS model in predicting the Total Heating Energy variable.

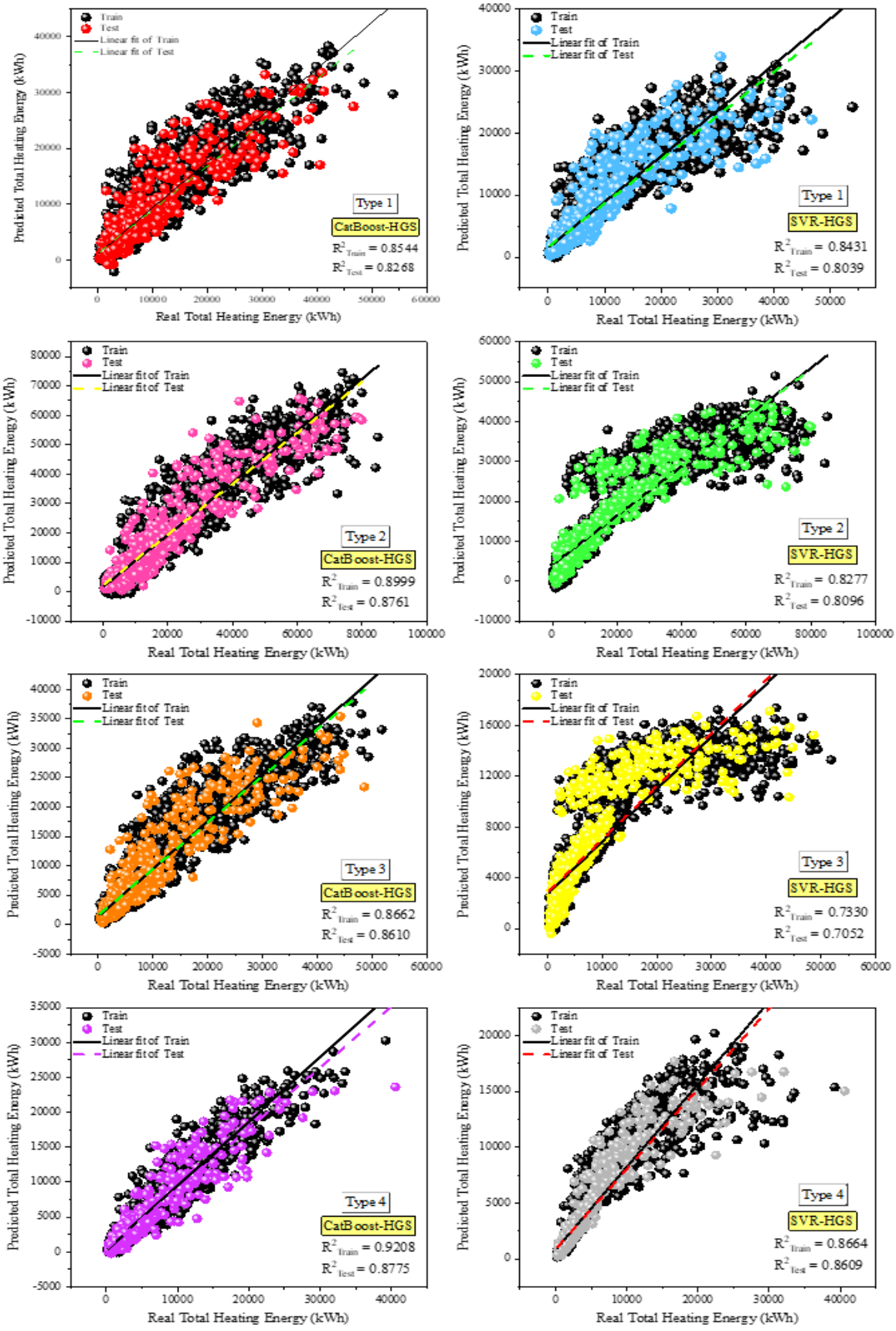


Figure 19: The observation-prediction plots related to the Total Heating Energy variable

In Fig. 20, the values of the evaluation indices related to both models for the Total Heating Energy variable are shown separately for different building types. According to this figure, the CatBoost-HGS model exhibits lower

MAE, MAPE, NRMSE, and RAE values, as well as a higher R-squared value compared to the SVR-HGS model across all building types. Therefore, the CatBoost-HGS model is expected to predict the Total Heating Energy variable more accurately. Additionally, the CatBoost-HGS model demonstrates greater accuracy in predicting Total Heating Energy values for type 2 and type 4 buildings compared to the other two building types.

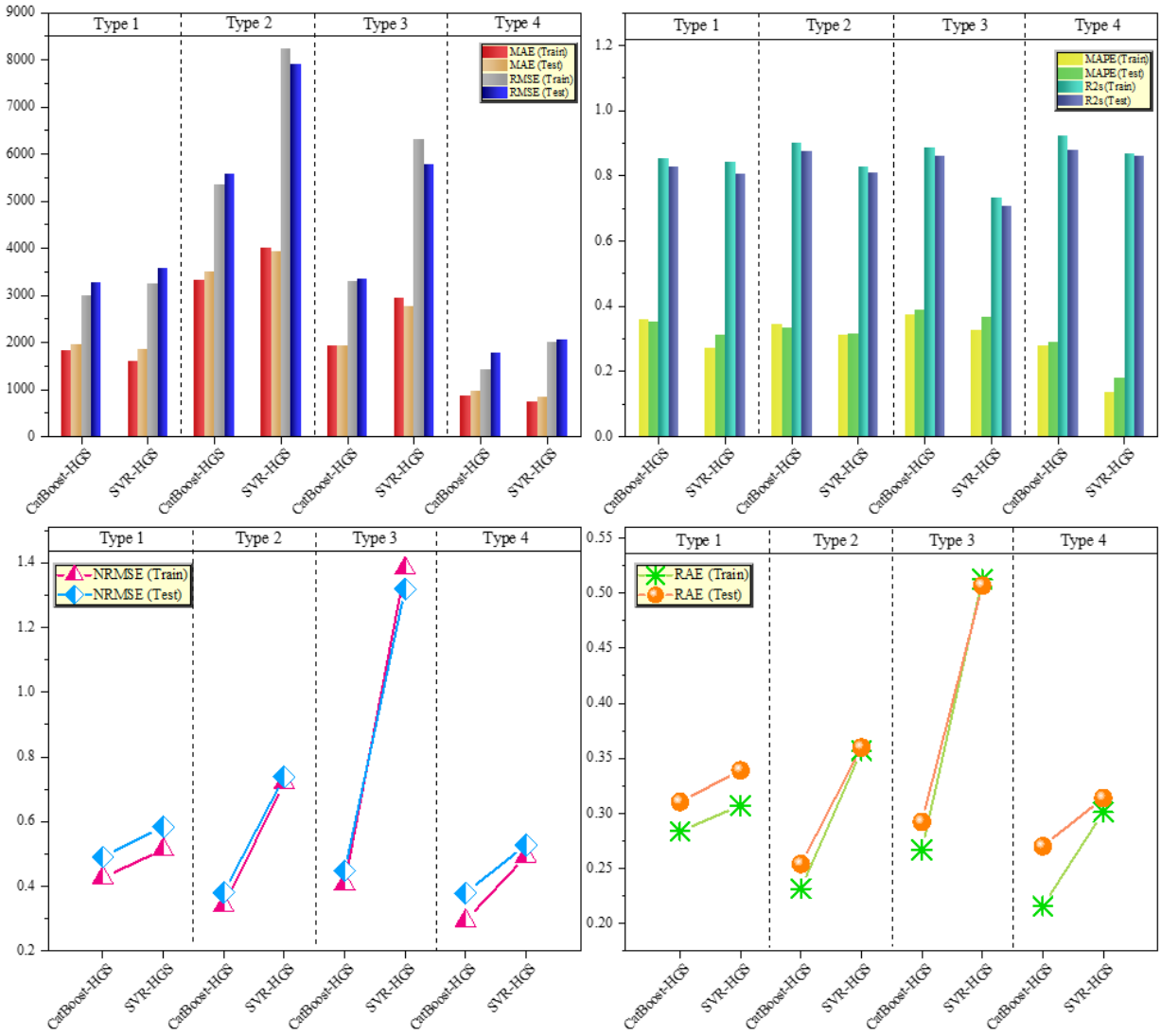


Figure 20: The observation-prediction plots related to the Total Heating Energy variable

In Figures 21 and 22, the Taylor diagrams for all models are shown separately for training and testing datasets. These diagrams indicate that the CatBoost-HGS model is closer to the "Total Heating Energy" point, suggesting it is the most accurate in reproducing observations. However, based on Figures 21 and 22, it can also be seen that, in both training and test datasets and across all building types, the SVR-HGS model performs best in reproducing its observations. A similar pattern is observed in Figures 21 and 22 for heating energy prediction, where the visual closeness of the two models in the Taylor diagrams may appear inconsistent with their distinct prediction behaviors discussed elsewhere. This again reflects the limitations of single-plot visualizations in capturing fine-grained differences. The diagrams serve as supportive, not definitive, indicators of performance and must be complemented by direct statistical evaluations.

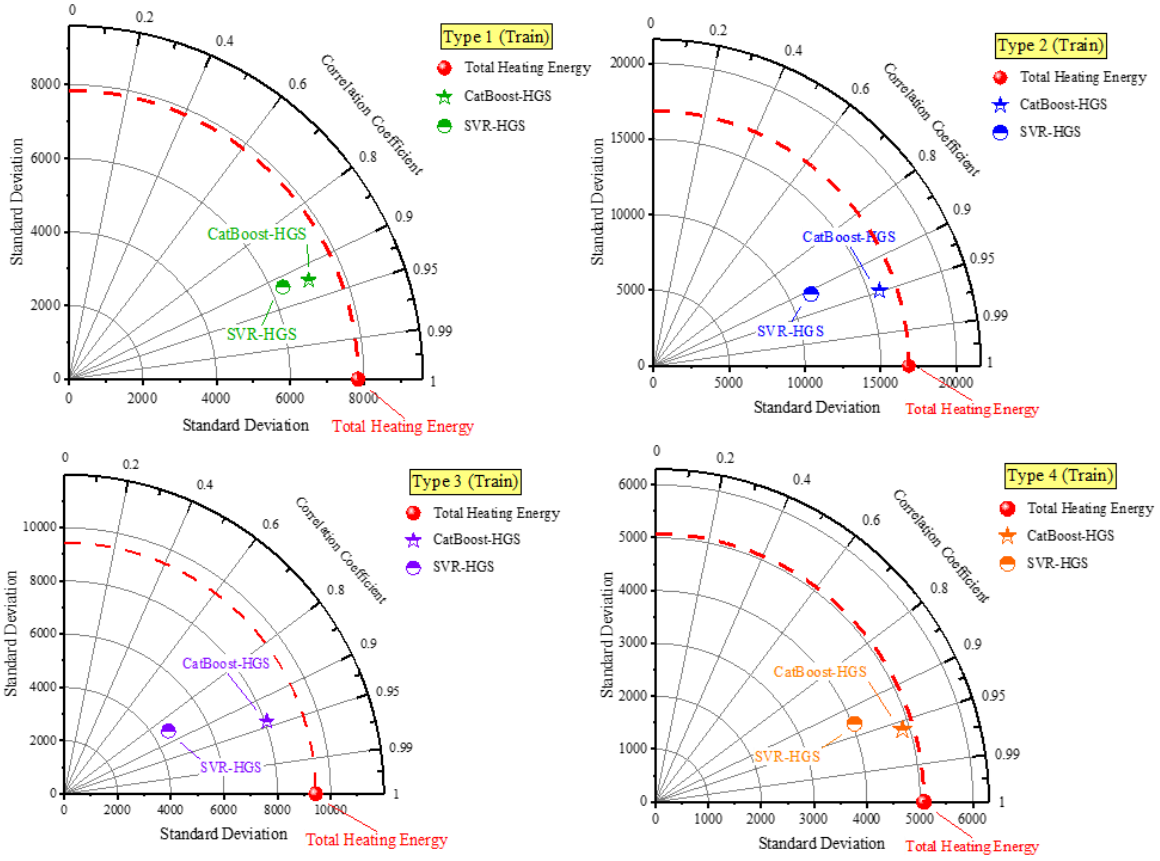


Figure 21. Taylor diagram of the models based on the training dataset

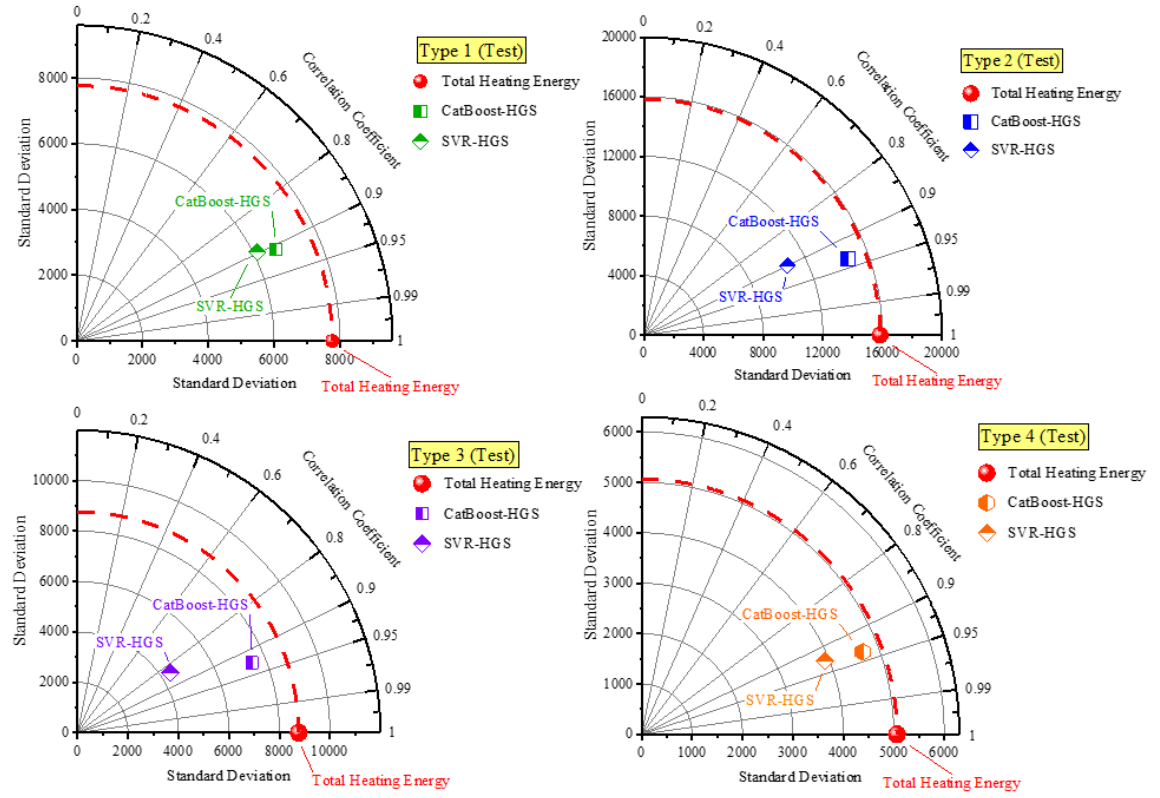
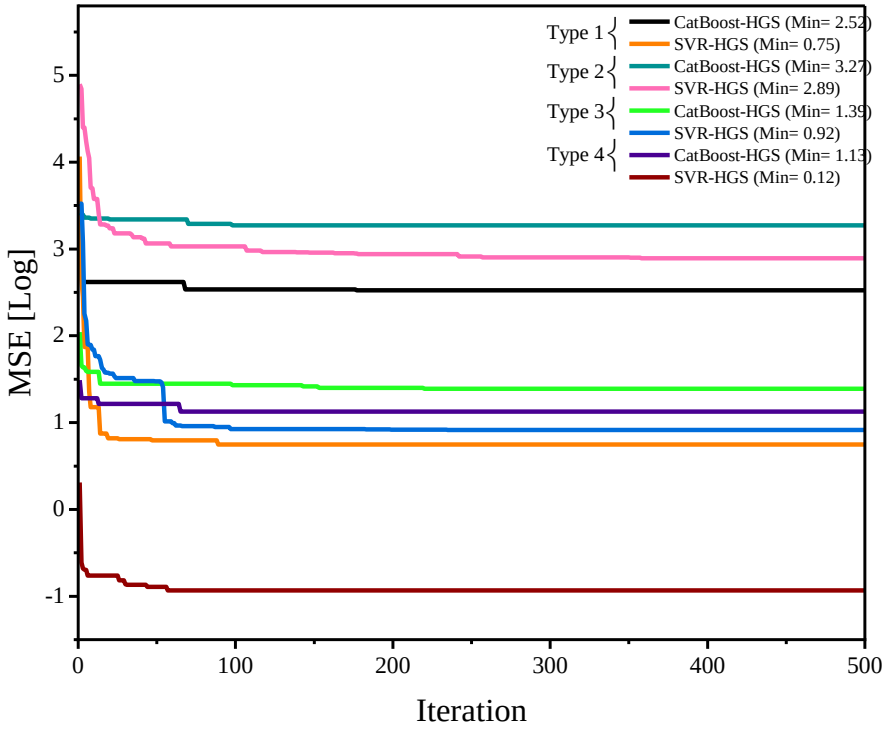


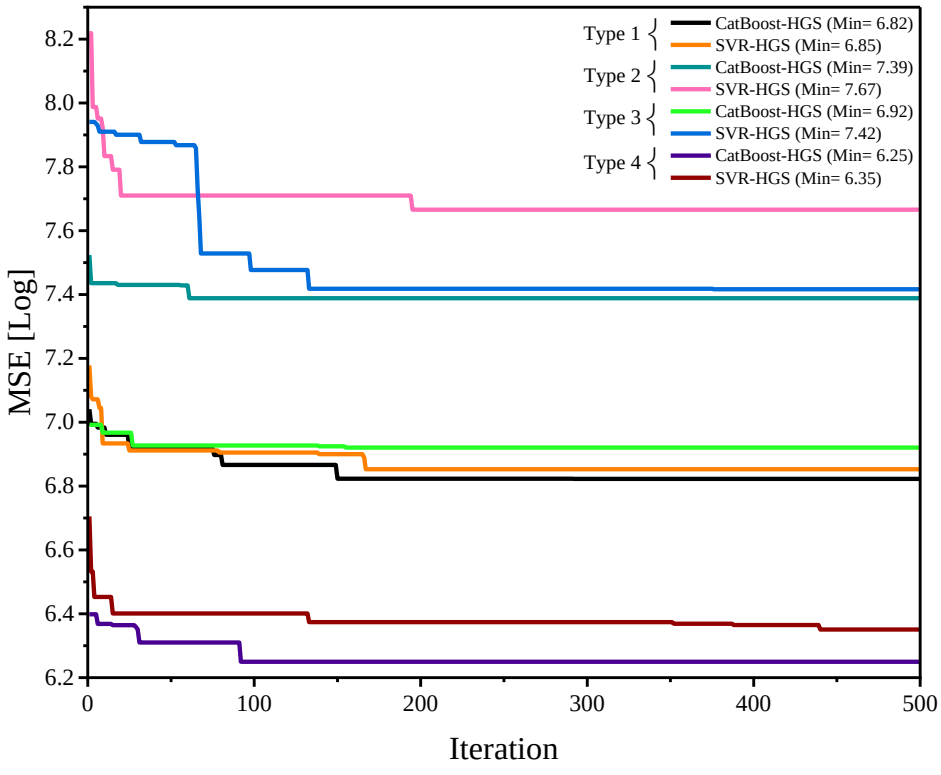
Figure 22: Taylor diagram of the models based on the testing dataset

In the following, the convergence values and runtime of each model will be examined. Figure 23 displays the convergence curves of the models for the output variables, Interior Lighting Energy and Total Heating Energy. The

horizontal axis of the diagram represents the iteration number, set at 500, while the vertical axis shows the Mean Squared Error (MSE) index as the convergence criterion. This figure also highlights the lowest convergence values. Based on the Interior Lighting Energy variable, the SVR-HGS hybrid model exhibits lower convergence values compared to the CatBoost-HGS hybrid model across all building types. Conversely, for the Total Heating Energy variable, the CatBoost-HGS hybrid model demonstrates lower convergence values than the SVR-HGS hybrid model.



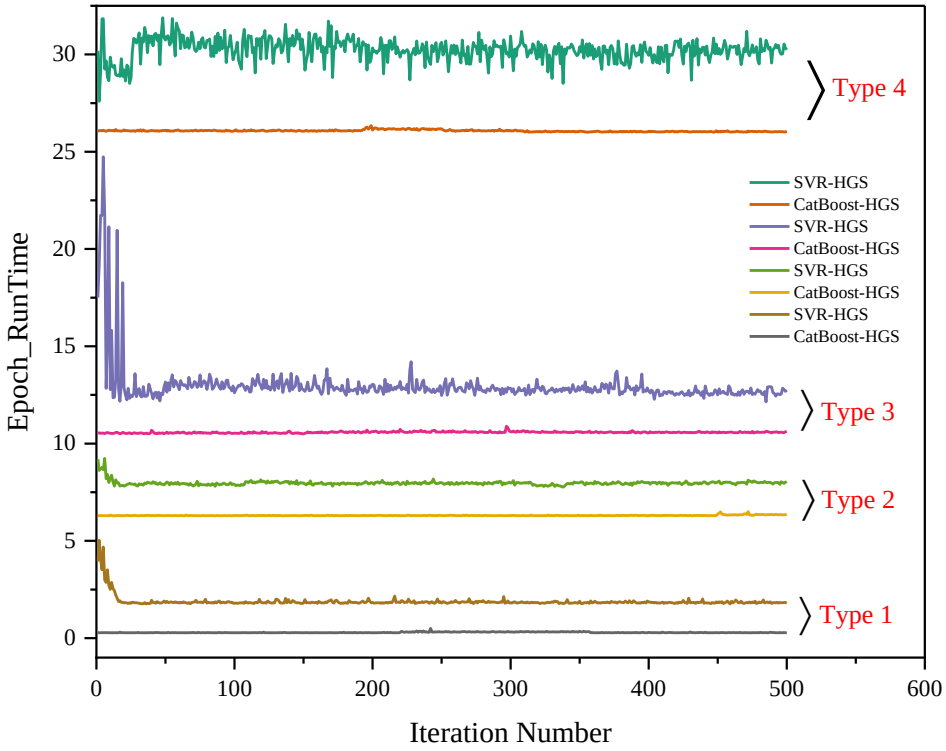
(a) Interior Lighting Energy



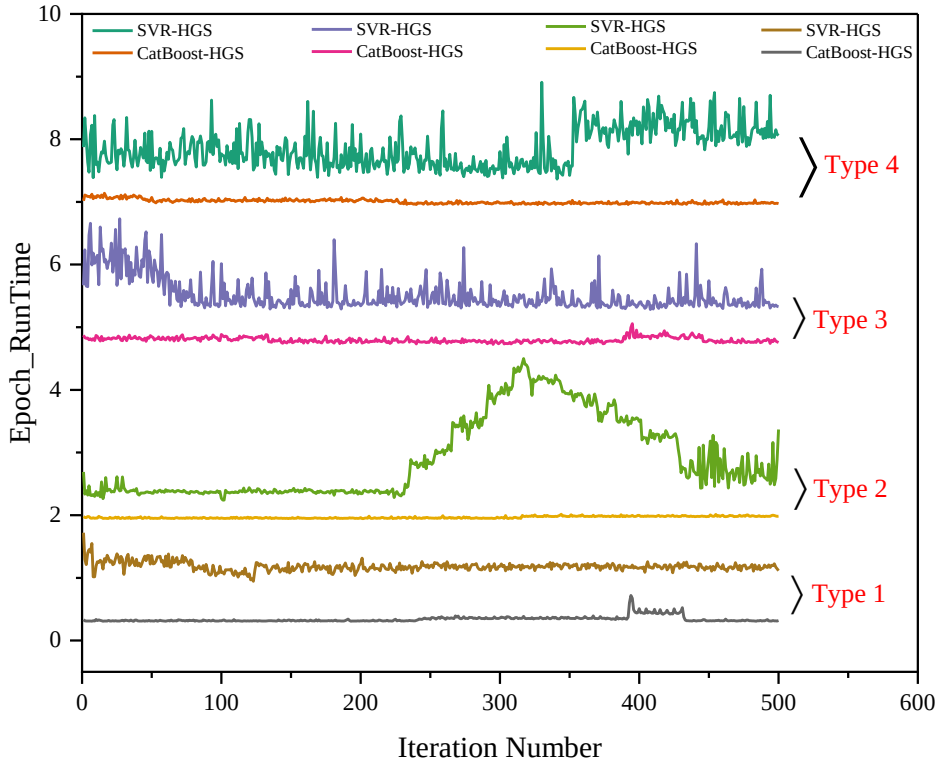
(b) Total Heating Energy

Figure 23: Convergence curves of all models

Figure 24 compares the runtime of each model per iteration for both output variables. According to this figure, the CatBoost-HGS hybrid model has lower runtime values than the SVR-HGS hybrid model for each iteration of both output variables. As a result, the CatBoost-HGS hybrid model demonstrates a higher execution speed compared to the SVR-HGS hybrid model.



(a) Interior Lighting Energy



(b) Total Heating Energy

Figure 24: Comparison of the run times of all models

Tables 4 and 5 display the evaluation indices for the models based on the Interior Lighting Energy and Total Heating Energy variables, respectively. According to Table 4, the SVR-HGS hybrid model outperforms the CatBoost-HGS hybrid model in terms of several evaluation indices for the Interior Lighting Energy variable, except for the R2 value. Specifically, the SVR-HGS hybrid model demonstrates improvements over the CatBoost-HGS hybrid model with reductions in MAE, RMSE, MAPE, RAE, and NRMSE by 46.58%, 44.42%, 19.04%, 44.42%, and 44.09%, respectively, based on the test dataset for type 2 buildings. Conversely, as shown in Table 5, the CatBoost-HGS hybrid model performs better than the SVR-HGS hybrid model for the Total Heating Energy variable, with lower evaluation index values (except for R2R2 and MAPE) based on the test dataset. This indicates that the CatBoost-HGS hybrid model has superior performance for predicting total heating energy compared to the SVR-HGS hybrid model.

The performance variation of SVR-HGS and CatBoost-HGS across different building types, as reflected in Tables 4 and 5, can be attributed to the distinct architectural and thermal dynamics of each category. For example, Type 2 buildings—detached houses—typically exhibit more consistent lighting usage patterns due to fewer shading effects and uniform window placement, allowing SVR-HGS to model lighting energy consumption with high precision. In contrast, heating energy prediction involves additional complexity stemming from insulation characteristics, HVAC system behavior, and occupancy variations, where CatBoost-HGS demonstrates superior handling due to its ability to capture nonlinear feature interactions. These differences emphasize the importance of aligning model architecture with the specific energy variable and building type under study.

Table 4: Evaluation indices based on Interior Lighting Energy variable

Metrics	Models	Type 1		Type 2		Type 3		Type 4	
		Tain	Test	Tain	Test	Tain	Test	Tain	Test
MAE	CatBoost-HGS	19.431	20.950	32.197	35.671	4.6391	4.9177	3.8658	3.9017
	SVR-HGS	1.6761	1.5936	18.046	19.053	1.7933	1.8823	0.2164	0.2073
RMSE	CatBoost-HGS	27.387	28.366	47.080	59.329	6.9188	7.2895	5.5943	5.6469
	SVR-HGS	2.6920	2.5492	31.087	32.976	3.0005	3.1265	0.3585	0.3287
MAPE	CatBoost-HGS	0.0214	0.0218	0.0353	0.0370	0.0050	0.0051	0.0040	0.0041
	SVR-HGS	0.0029	0.0026	0.0274	0.0299	0.0041	0.0037	0.0004	0.0004
R2	CatBoost-HGS	0.9985	0.9983	0.9972	0.9957	0.9998	0.9998	0.999	0.9999
	SVR-HGS	0.9999	0.9999	0.9990	0.9990	0.9999	0.9999	0.9999	0.9999
RAE	CatBoost-HGS	0.0200	0.0206	0.0278	0.0353	0.0062	0.0064	0.0047	0.0047
	SVR-HGS	0.0019	0.0018	0.0183	0.0196	0.0026	0.0027	0.0003	0.0002
NRMSE	CatBoost-HGS	0.0382	0.0409	0.0534	0.0673	0.0122	0.0128	0.0092	0.0091
	SVR-HGS	0.0037	0.0036	0.0356	0.0376	0.0053	0.0055	0.0005	0.0005

Table 5: Evaluation indices based on the Total Heating Energy variable

Metrics	Models	Type 1		Type 2		Type 3		Type 4	
		Tain	Test	Tain	Test	Tain	Test	Tain	Test
MAE	CatBoost-HGS	1817.8	1954.6	3319.6	3508.1	1939.9	1917.9	864.97	979.70
	SVR-HGS	1610.9	1866	4007.7	3931.1	2936.9	2763.1	737.13	841.72
RMSE	CatBoost-HGS	3000.4	3266.9	5335	5576.5	3286.1	3334.6	1428.6	1777.1
	SVR-HGS	3247.4	3569	8222	7901.4	6310.1	5782.3	1994.5	2065.5
MAPE	CatBoost-HGS	0.3587	0.3536	0.3443	0.3337	0.3731	0.3897	0.2793	0.2903
	SVR-HGS	0.2708	0.3122	0.3114	0.3152	0.3276	0.3657	0.1376	0.1791
R2	CatBoost-HGS	0.8544	0.8268	0.9	0.8761	0.8861	0.8610	0.9207	0.8775
	SVR-HGS	0.8431	0.8039	0.8276	0.8096	0.7330	0.7055	0.8664	0.8609
RAE	CatBoost-HGS	0.2837	0.3105	0.2316	0.2540	0.2669	0.2923	0.2157	0.2701
	SVR-HGS	0.3070	0.3392	0.3569	0.3599	0.5125	0.5069	0.3011	0.3140
NRMSE	CatBoost-HGS	0.4265	0.4906	0.3396	0.3811	0.4069	0.4477	0.2938	0.3800
	SVR-HGS	1817.8	1954.6	0.7199	0.7383	1.3825	1.3187	0.4934	0.5279

Table 6 presents the statistical comparison of the CatBoost-HGS and SVR-HGS models across different building types. The N. Compared column shows the number of test pairs compared, while the Median Difference column reports the median difference in absolute error (AE) between the two models. A wilcoxon-test was applied to assess the statistical significance of the differences between models, with the P-value indicating the significance level. A P-value less than 0.05 suggests that the difference is statistically significant. The

Conclusion column summarizes the results of the wilcoxon-test, indicating that the SVR-HGS model generally performs better with lower AE, except for Type 3 of THE, where CatBoost-HGS outperformed SVR-HGS. These results confirm that the differences in model performance are statistically significant, validating the robustness of the findings across different building types and datasets.

Table 6: Statistical comparison of model performance using wilcoxon-tests for AE

Dataset	Model 1	Model 2	N-Compared (Test Pairs)	Median-Difference (AE_M1 - AE_M2)	Statistic (W)	P-value	Significant (α=0.05)	Conclusion
ILE Type 1	Catboost-HGS	SVR-HGS	800	17.1957	5067.0	1.874e-124	TRUE	SVR-HGS Significantly Better (Lower AE)
ILE Type 2	Catboost-HGS	SVR-HGS	800	8.9497	87591.0	1.179e-28	TRUE	SVR-HGS Significantly Better (Lower AE)
ILE Type 3	Catboost-HGS	SVR-HGS	800	0.6802	97989.0	1.814e-21	TRUE	SVR-HGS Significantly Better (Lower AE)
ILE Type 4	Catboost-HGS	SVR-HGS	800	1.0172	9891.0	5.907e-117	TRUE	SVR-HGS Significantly Better (Lower AE)
THE Type 1	Catboost-HGS	SVR-HGS	800	138.4889	130188.0	4.426e-06	TRUE	SVR-HGS Significantly Better (Lower AE)
THE Type 2	Catboost-HGS	SVR-HGS	800	186.2003	146016.0	0.0300	TRUE	SVR-HGS Significantly Better (Lower AE)
THE Type 3	Catboost-HGS	SVR-HGS	800	-16.3353	144170.0	0.0142	TRUE	Catboost-HGS Significantly Better (Lower AE)
THE Type 4	Catboost-HGS	SVR-HGS	800	158.6251	97452.0	8.207e-22	TRUE	SVR-HGS Significantly Better (Lower AE)

5 Discussion

In this study, the performance of the SVR-HGS and CatBoost-HGS hybrid models was compared with several methods reviewed in previous works. The SVR-HGS model demonstrated superior performance, particularly in predicting interior lighting energy, achieving significantly lower MAE and MAPE values than the models in studies by Fathi et al. (Fathi et al., 2020)and Syed et al. (Syed et al., 2021). Similarly, the CatBoost-HGS hybrid model provided more accurate forecasts for total heating energy compared to other state-of-the-art methods, highlighting the effectiveness of the proposed approach. The performance differences observed can be attributed to multiple factors. One key reason is the size and diversity of the dataset used in this study, which includes over a million records from various building types, offering a more comprehensive dataset compared to the smaller datasets utilized in other studies. Furthermore, the models in this study captured the impact of architectural features such as building form and material properties, which play a critical role in energy consumption patterns. The hybridization of the SVR and CatBoost models with the HGS optimization method also allowed for better hyperparameter tuning, leading to improved prediction accuracy. This optimization process distinguishes our approach from other methods that either rely on standard training processes or use less advanced optimization techniques. The novel contribution of this work lies in the hybridization of machine learning models with advanced optimization techniques like HGS, which effectively captures the non-linear relationships between building

features and energy consumption. The HGS algorithm was chosen due to its strong global search capabilities, dynamic balance between exploration and exploitation, and its proven effectiveness in handling complex, high-dimensional optimization problems. Unlike some traditional metaheuristic algorithms (e.g., PSO, GA, or DE), HGS uses a unique selection mechanism inspired by competitive behaviors, which helps it to avoid premature convergence and improves solution diversity. Additionally, HGS has demonstrated superior or competitive performance in recent comparative studies across various engineering and machine learning optimization tasks. Given the non-convex and multimodal nature of our objective function, these features make HGS a suitable and efficient choice for our problem domain. Furthermore, preliminary experiments we conducted also indicated that HGS outperformed other baseline optimizers in both convergence speed and final solution quality on our dataset. This is an aspect not sufficiently addressed by other studies. Additionally, the integration of architectural design features and the ability to optimize hyperparameters resulted in more robust and precise energy consumption forecasts, making the model more applicable to a wide range of residential building types. These innovations enhance the generalizability of the model and offer a more effective solution for energy management in real-world applications.

Another topic that is open to discussion is the runtime differences between the SVR-HGS and CatBoost-HGS models, as illustrated in Figure 24. While both models achieve high accuracy, the SVR-HGS model consistently demonstrates significantly faster runtime performance across all building types. This faster execution makes it especially suitable for large-scale or real-time energy prediction scenarios, where rapid feedback and adaptability are essential. In contrast, although the CatBoost-HGS model offers slightly higher accuracy for complex variables like total heating energy, its longer training time may limit its applicability in time-sensitive or resource-constrained environments. Therefore, selecting between the two models depends on the trade-off between accuracy and computational efficiency. The ability of SVR-HGS to retrain quickly and adapt to evolving energy consumption patterns provides an additional advantage in dynamic residential energy systems.

Also, to provide a comprehensive evaluation of model performance, multiple metrics were selected, including MAE, MAPE, NRMSE, RAE, and R^2. Each metric serves a distinct role: MAE offers a direct interpretation of the average absolute error, MAPE provides a relative error useful for comparing across outputs of different scales, and NRMSE normalizes the error relative to the range of observed values, enabling comparisons between models. RAE reflects how well the model performs in comparison to a naive baseline, while R^2 evaluates the proportion of variance explained by the model. The inclusion of MAPE, in particular, is helpful in distinguishing model performance for lower-magnitude variables, such as lighting energy, versus larger-scale variables, such as heating energy. Furthermore, these statistical improvements have direct practical relevance in real-world energy management. For instance, a lower MAE allows for tighter energy budgeting, reducing the margin of uncertainty in operational planning. Similarly, improved MAPE enhances the reliability of predictive control strategies across diverse building scales, making the models more applicable to dynamic energy monitoring systems. These implications translate numerical accuracy into actionable insights for demand forecasting and consumption optimization. Although high R^2 values were observed across all models, the risk of overfitting was carefully considered. The use of HGS for hyperparameter optimization, while beneficial for improving accuracy, can potentially lead to overfitting if not properly controlled. To address this, cross-validation was applied during training to ensure generalization, and the model's performance was validated on an independent test set to confirm robustness. The alignment between validation and test performance metrics suggests that the models-maintained generalizability despite the use of advanced optimization techniques.

6 Conclusion

Effective building energy management is crucial for the sustainability of cities, given the significant energy consumption of the building sector and the importance of providing both energy efficiency and comfort to occupants. Energy consumption largely depends upon architectural design and the form of buildings. This paper will propose a model that, through machine learning techniques, will be able to predict the energy consumption of different kinds of residential buildings. The ML techniques applied in this research work are CatBoost and Support Vector Regression. HGS was then applied to all those primary algorithms in order to optimize and fine-tune their hyperparameters with a view to improving the prediction accuracy. The results obtained in this study showed that the SVR coupled with the HGS algorithm hybrid model was the most accurate interior lighting energy predictor in all four residential buildings, while the CatBoostHGS hybrid model was superior for total heating energy forecasting. The proposed methods were confirmed to be exact using the evaluation metrics. Precisely, it follows from the test dataset that for an interior lighting energy prediction, the obtained NRMSE with the SVR-HGS hybrid model is 44.09% better compared to the CatBoost-HGS hybrid model. For total heating energy prediction, the NRMSE is improved 93.69% by the CatBoost-HGS hybrid model compared to the SVR-HGS hybrid model. Moreover, results indicate that the interior lighting energy variable convergence values for the SVRHGS hybrid model are lower, for all building types, than that of the CatBoostHGS hybrid model. On the contrary, the total heating energy variable convergence values of the SVR-HGS hybrid model have been lower than that of the CatBoost-HGS hybrid model. To further enhance the robustness of the models, future work will extend the study by incorporating cross-validation with more diverse datasets. This will help validate the generalizability of the models across different

building types and regions. Additionally, ablation studies will be conducted to evaluate the contribution of individual features, allowing for a deeper understanding of how each variable influences the performance of the model. These efforts will strengthen the findings of this study and ensure the model's broader applicability in real-world energy prediction tasks.

Nomenclature

Abbreviation	Explanation	Abbreviation	Explanation
$bound_{lower}$	The lower bound.	l	A variable to improve the HGS algorithm
$bound_{upper}$	The upper bound.	M	Number of features
$fitness_{best}$	The best fitness value	ML	Machine Learning
$fitness_i$	The obtained fitness in the current iteration	N	The initial population
$fitness_{worst}$	The worst fitness obtained in the current iteration	no	The number of individuals
$r_1, r_2, \ldots, r_6$	Random numbers in the range of 0 and 1.	Npop	Number of individuals
$\bar{y}$	The mean of observations	PDF	Probability Density Function
y_i	The ith observed value	RBF	Radial Basis Function
$\delta_{largest}$	A parameter to improve the performance of the algorithm.	$\vec{R}$	A variable in the range of a and -a
δ_{new}	A new hunger	RT	Random Tree
AI	Artificial Intelligence	SVM	Support vector machine
ANN	Artificial Neural Network	SVR	Support Vector Regression
CatBoost	Categorical boosting	SVR	Support Vector Regression
DL	Deep Learning	T	The maximum number of iterations.
E	A change control for all situations.	TS	Target statistics
GA	Genetic Algorithm	$\overrightarrow{W_1}, \overrightarrow{W_2}$	The weight of hunger behavior
GB	Gradient Boosting	$\vec{X}_b$	The position of the best individual
HGS	Hunger Games Search	$\overrightarrow{X(t)}$	The position of each individual at iteration t
HGS	Hunger Games Search	$\hat{y}_i$	The ith estimated value
hype.func	hyperbolic function	$\delta(i)$	The hunger of each individual
KNN	K-Nearest Neighbors		

Competing of interests

The authors declare no competing of interests.

Authorship contribution statement

Xueli Chen: Writing-Original draft preparation, Conceptualization, Supervision.

Ziru Zhang: Project administration

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] Ahmed, A., T. Ge, J. Peng, W.-C. Yan, B.T. Tee and S. You (2022). Assessment of the renewable energy generation towards net-zero energy buildings: A review. *Energy and Buildings*, Elsevier, 256, 111755. https://doi.org/10.1016/j.enbuild.2021.111755

[2] González-Torres, M., L. Pérez-Lombard, J.F. Coronel, I.R. Maestre and D. Yan (2022). A review on buildings energy information: Trends, end-uses, fuels and drivers. *Energy Reports*, Elsevier, 8, pp. 626–637. https://doi.org/10.1016/j.egyr.2021.11.280

[3] Bakmohammadi, P. and E. Noorzai (2020). Optimization of the design of the primary school classrooms in terms of energy and daylight performance considering occupants' thermal and

visual comfort. *Energy Reports*, Elsevier, 6, pp. 1590–1607. https://doi.org/10.1016/j.egyr.2020.06.008

[4] Shapi, M.K.M., N.A. Ramli and L.J. Awalin (2021). Energy consumption prediction by using machine learning for smart building: Case study in Malaysia. *Developments in the Built Environment*, Elsevier, 5, 100037. https://doi.org/10.1016/j.dibe.2020.100037

[5] Wilberforce, T., A.G. Olabi, E.T. Sayed, K. Elsaid, H.M. Maghrabie and M.A. Abdelkareem (2023). A review on zero energy buildings–Pros and cons. *Energy and Built Environment*, Elsevier, 4(1), pp. 25–38. https://doi.org/10.1016/j.enbenv.2021.06.002

[6] Sun, Y., F. Haghighat and B.C.M. Fung (2020). A review of the-state-of-the-art in data-driven approaches for building energy prediction. *Energy and Buildings*, Elsevier, 221, 110022. https://doi.org/10.1016/j.enbuild.2020.110022

[7]. Kumar, A., S. Sharma, N. Goyal, A. Singh, X. Cheng and P. Singh (2021). Secure and energy-efficient smart building architecture with emerging technology IoT. *Computer Communications*, Elsevier, 176, pp. 207–217. https://doi.org/10.1016/j.comcom.2021.06.003

[8] Pereira, V., J. Santos, F. Leite and P. Escórcio (2021). Using BIM to improve building energy efficiency–A scientometric and systematic review. *Energy and Buildings*, Elsevier, 250, 111292. https://doi.org/10.1016/j.enbuild.2021.111292

[9] Pásztory, Z (2021). An overview of factors influencing thermal conductivity of building insulation materials. *Journal of Building Engineering*, Elsevier, 44, 102604. https://doi.org/10.1016/j.jobe.2021.102604

[10] Ilbeigi, M., M. Ghomeishi and A. Dehghanbanadaki (2020). Prediction and optimization of energy consumption in an office building using artificial neural network and a genetic algorithm. *Sustainable Cities and Society*, Elsevier, 61, 102325. https://doi.org/10.1016/j.scs.2020.102325

[11] Olu-Ajayi, R., H. Alaka, I. Sulaimon, F. Sunmola and S. Ajayi (2022). Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. *Journal of Building Engineering*, Elsevier, 45, 103406. https://doi.org/10.1016/j.jobe.2021.103406

[12] Pham, A.-D., N.-T. Ngo, T.T.H. Truong, N.-T. Huynh and N.-S. Truong (2020). Predicting energy consumption in multiple buildings using machine learning for improving energy efficiency and sustainability. *Journal of Cleaner Production*, Elsevier, 260, 121082. https://doi.org/10.1016/j.jclepro.2020.121082

[13] Lu, C., S. Li and Z. Lu (2022). Building energy prediction using artificial neural networks: A literature survey. *Energy and Buildings*, Elsevier, 262, 111718. https://doi.org/10.1016/j.enbuild.2021.111718

[14] Zhang, L., J. Wen, Y. Li, J. Chen, Y. Ye, Y. Fu and W. Livingood (2021). A review of machine learning in building load prediction. *Applied Energy*, Elsevier, 285, 116452. https://doi.org/10.1016/j.apenergy.2021.116452

[15] Ngarambe, J., G.Y. Yun and M. Santamouris (2020). The use of artificial intelligence (AI) methods in the prediction of thermal comfort in buildings: *Energy* implications of AI-based thermal comfort controls. *Energy and Buildings*, Elsevier, 211, 109807. https://doi.org/10.1016/j.enbuild.2020.109807

[16] Fathi, S., R. Srinivasan, A. Fenner and S. Fathi (2020). Machine learning applications in urban building energy performance forecasting: A systematic review. *Renewable and Sustainable Energy Reviews*, Elsevier, 133, 110287. https://doi.org/10.1016/j.rser.2020.110287

[17] Walker, S., W. Khan, K. Katic, W. Maassen and W. Zeiler (2020). Accuracy of different machine learning algorithms and added-value of predicting aggregated-level energy performance of commercial buildings. *Energy and Buildings*, Elsevier, 209, 109705. https://doi.org/10.1016/j.enbuild.2019.109705

[18] Seyedzadeh, S., F.P. Rahimian, S. Oliver, S. Rodriguez and I. Glesk (2020). Machine learning modelling for predicting non-domestic buildings energy performance: A model to support deep energy retrofit decision-making. *Applied Energy*, Elsevier, 279, 115908. https://doi.org/10.1016/j.apenergy.2020.115908

[19] Baduge, S.K., S. Thilakarathna, J.S. Perera, M. Arashpour, P. Sharafi, B. Teodosio, A. Shringi and P. Mendis (2022). Artificial intelligence and smart vision for building and construction 4.0: Machine and deep learning methods and applications. *Automation in Construction*, Elsevier, 141, 104440. https://doi.org/10.1016/j.autcon.2022.104440

[20] Rawat, R., A. Samson, A. Raj, R.K. Chakrawarti, K.S. Sankaran, S.K. Sarangi, H. Rawat and A. Rawat, n.d. Enhanced Cybercrime Detection on Twitter Using Aho-Corasic Enhanced Cybercrime Detection on Twitter Using Aho-Corasick Algorithm and Machine Learning Techniques. Informatica, https://doi.org/10.31449/inf.v48i18.6272

[21] Adeniji, O.D., S.O. Adeyemi and S.A. Ajagbe (2022). An Improved Bagging Ensemble in Predicting Mental Disorder Using Hybridized Random Forest - Artificial Neural Network Model. *Informatica (Slovenia)*, Informatica, 46(4), pp. 543–550. https://doi.org/10.31449/inf.v46i4.3916

[22] Kaushal, A., A.K. Gupta and V.K. Sehgal (2025). Hybrid CatBoost and SVR Model for Earthquake Prediction Using the LANL Earthquake Dataset. *Informatica*, Informatica, 49(14). https://doi.org/10.31449/inf.v49i14.6524

[23] Liu, Y., H. Chen, L. Zhang, X. Wu and X. Wang (2020). Energy consumption prediction and diagnosis of public buildings based on support vector machine learning: A case study in China. *Journal of Cleaner Production*, Elsevier, 272, 122542. https://doi.org/10.1016/j.jclepro.2020.122542

[24] Khan, P.W. and Y.-C. Byun (2020). Genetic algorithm based optimized feature engineering and hybrid machine learning for effective energy consumption prediction. *Ieee Access*, IEEE, 8, pp. 196274–196286. https://doi.org/10.1109/ACCESS.2020.3034101

[25] Syed, D., H. Abu-Rub, A. Ghrayeb and S.S. Refaat (2021). Household-level energy forecasting in smart buildings using a novel hybrid deep learning model. *IEEE Access*, IEEE, 9, pp. 33498–33511. https://doi.org/10.1109/ACCESS.2021.3061370

[26] Ardabili, S., L. Abdolalizadeh, C. Mako, B. Torok and A. Mosavi (2022). Systematic review of deep learning and machine learning for building energy. *Frontiers in Energy Research*, frontiers, 10, 786027. https://doi.org/10.3389/fenrg.2022.786027

[27] Olu-Ajayi, R., H. Alaka, I. Sulaimon, F. Sunmola and S. Ajayi (2022). Machine learning for energy performance prediction at the design stage of buildings. *Energy for Sustainable Development*, Elsevier, 66, pp. 12–25. https://doi.org/10.1016/j.esd.2021.11.002

[28] Ghadiri, M., A.A. Rassafi and B. Mirbaha (2019). The effects of traffic zoning with regular geometric shapes on the precision of trip production models. *Journal of Transport Geography*, Elsevier, 78, pp. 150–159. https://doi.org/10.1016/j.jtrangeo.2019.05.018

[29] Rastgoo, A. and H. Khajavi (2023). A novel study on forecasting the airfoil self-noise, using a hybrid model based on the combination of CatBoost and Arithmetic Optimization Algorithm. *Expert Systems with Applications*, Elsevier, 229, 120576. https://doi.org/10.1016/j.eswa.2023.120576

[30] Khajavi, H. and A. Rastgoo (2023). Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms. *Energy*, Elsevier, 272, 127069. https://doi.org/10.1016/j.energy.2023.127069

[31] Khajavi, H. and A. Rastgoo (2023). Predicting the carbon dioxide emission caused by road transport using a Random Forest (RF) model combined by Meta-Heuristic Algorithms. *Sustainable Cities and Society*, Elsevier, 93, 104503. https://doi.org/10.1016/j.scs.2023.104503

[32] Zhang, Y., Z. Zhao and J. Zheng (2020). CatBoost: A new approach for estimating daily reference crop evapotranspiration in arid and semi-arid regions of Northern China. *Journal of Hydrology*, Elsevier, 588, 125087. https://doi.org/10.1016/j.jhydrol.2020.125087

[33] Luo, M., Y. Wang, Y. Xie, L. Zhou, J. Qiao, S. Qiu and Y. Sun (2021). Combination of feature selection and catboost for prediction: The first application to the estimation of aboveground biomass. *Forests*, MDPI, 12(2), 216. https://doi.org/10.3390/f12020216

[34] SAFAROV, R.Z., Z.K. SHOMANOVA, Y.G. NOSSENKO, Z.G. BERDENOV, Z.B. BEXEITOVA, A.S. SHOMANOV and M. MANSUROVA (2020). Solving of classification problem in spatial analysis applying the technology of gradient boosting catboost. *Folia Geographica*, 62(1), 112.

[35] Huang, G., L. Wu, X. Ma, W. Zhang, J. Fan, X. Yu, W. Zeng and H. Zhou (2019). Evaluation of CatBoost method for prediction of reference evapotranspiration in humid regions. *Journal of Hydrology*, Elsevier, 574, pp. 1029–1041. https://doi.org/10.1016/j.jhydrol.2019.04.085

[36] Drucker, H., C.J. Burges, L. Kaufman, A. Smola and V. Vapnik (1996). Support vector regression machines. *Advances in Neural Information Processing Systems*, NeurIPS, 9.

[37] Yu, H. and S. Kim (2012). SVM Tutorial-Classification, Regression and Ranking. *Handbook of Natural Computing*, Springer, 1, pp. 479–506.

[38] Ghasemi, E., H. Kalhori and R. Bagherpour (2016). A new hybrid ANFIS–PSO model for prediction of peak particle velocity due to bench blasting. *Engineering with Computers*, Springer, 32, pp. 607–614. https://doi.org/10.1007/s00366-016-0438-1

[39] Yang, Y., H. Chen, A.A. Heidari and A.H. Gandomi (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, Elsevier, 177, 114864. https://doi.org/10.1016/j.eswa.2021.114864

[40] Nguyen, H. and X.-N. Bui (2021). A novel hunger games search optimization-based artificial neural network for predicting ground vibration intensity induced by mine blasting. *Natural Resources Research*, Springer, 30(5), pp. 3865–3880. https://doi.org/10.1007/s11053-021-09903-8

[41] AbuShanab, W.S., M. Abd Elaziz, E.I. Ghandourah, E.B. Moustafa and A.H. Elsheikh (2021). A new fine-tuned random vector functional link model using Hunger games search optimizer for modeling friction stir welding process of polymeric materials. *Journal of Materials Research and Technology*, Elsevier, 14, pp. 1482–1493. https://doi.org/10.1016/j.jmrt.2021.07.031

Multi-Class Classification of Customer Complaints Using Convolutional LSTM and Neural Network Models

Jing Yi, Xiao Zeng
School of Economics and Management, Guizhou Normal University, Guiyang 550001, Guizhou, China
Kmitl Business School, King Mongkut's Institute of Technology Ladkrabang, Bangkok 10520, Thailand
E-mail: yij3424@gmail.com
*Corresponding author

Keywords: neural network, text classification, customer complaints, convolution LSTM neural network, word embedding

Received: Februar 9, 2025

This paper provided multi-class classification models for customer complaints based on neural network architectures. It considered one-dimensional convolutional models corresponding to one, two, and three layers, and also included Long Short-Term Memory and Convolutional Long Short-Term Memory models. The models were assessed through performance indicators such as accuracy, area under the curve, precision, and recall. The findings indicated that Convolutional Long Short-Term Memory exhibited superior performance, achieving 88.06% accuracy and 0.92 area under the curve, respectively. This outperformed one convolutional layer models, Long Short-Term Memory, two convolutional layers, and three convolutional layers models, whose accuracy rates were 86.48%, 84.02%, 83.87%, and 74.71%, respectively. Moreover, Convolutional Long Short-Term Memory also surpassed other models in terms of precision and recall, achieving 0.89 and 0.86, respectively. The study demonstrated that it is superior to simple models to apply a hybrid Convolutional Long Short-Term Memory to classifying customer complaints. This technique combines effectively the advantages of convolutional and recurrent layers, facilitating an ability to learn both local and long-term features. The findings underscored the significance of appropriating optimal neural network architectures for complex text classification problems and also made an original contribution to research on consumer feedback.

Povzetek: Študija primerja različne nevronske modele za razvrščanje pritožb strank in ugotavlja, da hibridni model dosega najboljše rezultate.

1 Introduction

There are different types of data: some are in the form of numbers, images, and sounds, and some are in text. A significant percentage of the data available worldwide is in the form of text; every day, more text data is generated. Text sources are rich and helpful information, but extracting information from them is complex and time-consuming [1,2]. So far, several methods for text classification have been presented as one of the branches of natural language processing (NLP), among which artificial intelligence (AI)-oriented tactics are highly popular. This research utilized five distinct neural network (NN) frameworks to categorize textual information. This framework comprises one-dimensional convolutional models with one, two, and three layers, an LSTM model, and a convolutional LSTM (ConvLSTM) model. Each model was engineered to encapsulate distinct facets of the data, with the ConvLSTM model integrating the advantages of convolutional and LSTM layers to enhance classification efficacy. With text classification techniques, you can check and analyze text data and get valuable information from the results of this analysis [3–5].

Ghazzawi and Alharbi [6] investigated user complaints from the public transportation sector in America using data mining methods. For this purpose, they separated complaints into different classes by presenting a classification model. Among the applications of this model, it can be used to assist city managers and decision-makers in transportation management and planning. Zhong et al. [7] investigated building quality complaints and problems in China. For this purpose, they used CNN drawing on a DL tactic. The results of evaluating the recommended scheme with other ML-based classification algorithms show the complexity of this recommended scheme. Singh et al. [8] investigated product complaints using Indian language comments on social networks such as YouTube. For this purpose, they used different approaches based on machine learning for supervised classification and random walk algorithm for semi-supervised classification. The results showed that the classification is highly accurate. Muqorobin et al. [9] recommended a model for classifying people's complaints about public services provided in society by analyzing Twitter data. This model performs classification based on text analysis using the Naïve Bayes algorithm. Examining the evaluation indicators in a case study has indicated the

acceptable accuracy of this model in classifying complaints. Hong et al. [10] presented a multi-objective model for predicting environmental complaints from construction projects using the oversampling-based technique. By conducting a case study and checking the feasibility of the recommended scheme, they proved the appropriate productivity of this scheme. Osorio-Arjona et al. [11] assessed the public transport fleet by extracting people's complaints and dissatisfaction from tweets sent in the metro. For this purpose, they used machine learning and text analysis techniques to predict the spatial distribution of dissatisfied people and identify the significant dissatisfactions of subway passengers. Shin et al. [12] presented a hybrid model for predicting leakage in water networks using the submitted complaints. They used different ML methods and sampling techniques, finding that the hybrid model resulting from integrating LightGBM and hybrid sampling performs best. Qurat Ul et al. [13] recommended a machine learning and NLP model to investigate and classify complaints after examining the data collected in different departments. Their research results showed that their recommended scheme had acceptable accuracy. Nemer et al. [14] assessed CNN, LSTM, and a hybrid CNN+LSTM model for sleep stage classification utilizing EEG data, demonstrating that the integration of spatial and temporal feature extraction enhanced performance. The hybrid approach encounters difficulties regarding computational expense and scalability, necessitating optimization for real-time applications. Wang [15] augmented an LSTM for user experience prediction by incorporating a self-attention mechanism, achieving a 6.1% increase in accuracy compared to conventional LSTM, although encountering constraints in real-time applicability and comparative analysis with other attention-based models. Alkenani and Nickray [16] proposed ConvRNN for cyber-attack detection, surpassing LSTM across all criteria; nonetheless, its computational burden is a difficulty in large-scale systems. Pan and Ma [17] created a mental health prediction system utilizing CNN and LSTM, attaining an accuracy of 85.7%, although dependent on restricted behavioral data, indicating the necessity for a broader range of features in subsequent studies. Despite their high performance, these models exhibit common constraints such as computational expenses and scalability issues. Future studies should incorporate varied methodologies, including transformers, to enhance resilience and real-time application. The citations have been enhanced to better correspond with the fundamental methodology and applications addressed.

The literature review revealed that different schemes have been presented based on various approaches to investigate the classification problem. Each model has its conditions and characteristics and is used according to the problem. This research presents a multi-class classification model utilizing neural networks, designed to enhance classification accuracy in intricate tasks, such as a special task, hence contributing to a specific benefit or impact.

Table 1 summarizes important studies, comparing the methods, datasets, performance indicators, and main findings. It highlights the weaknesses in current methods and stresses the need for the proposed ConvLSTM model.

Table 1: Comparative summary of related works.

Study	Method Used	Dataset	Performance Metrics	Key Findings
Ghazzawi and Alharbi [6]	Data Mining	Public Transportation Complaints	F1-Score, Accuracy	Classification of complaints using traditional data mining techniques; no deep learning models tested.
Zhong et al. [7]	CNN	Building Quality Complaints	Accuracy, AUC	The CNN-based model showed good results, but did not explore hybrid models.
Singh et al. [8]	SVM, Random Walk	Social Media Comments (Hindi)	Precision, Recall, Accuracy	High accuracy in product complaints, but no hybrid model evaluation.
Muqorobin et al. [9]	Naïve Bayes	Twitter Data	Accuracy, Precision	Achieved acceptable accuracy, but no deep learning methods were compared.
Hong et al. [10]	Hybrid Model (LightGBM & Sampling)	Environmental Complaints (Construction Projects)	AUC, Accuracy	A hybrid model with LightGBM and sampling improved performance in predicting environmental complaints.
Arjona et al. [11]	ML and Text Analysis	Subway Tweets	Precision, Recall	Focused on spatial analysis, lacks comparison with deep learning techniques.
Shin et al. [12]	ML Techniques	Indoor Water Leakage Complaints	Precision, Recall, F1-Score	The hybrid model combining machine learning and sampling outperformed traditional models.

Shaukat and Saif [13]	NLP-Based Model	Customer Service Complaints	Accuracy, Precision	An NLP-based classification model for customer service complaints showed competitive results compared to standard methods.
Nemer et al. [14]	CNN, LSTM, CNN+LSTM	EEG Data (Sleep Stage Classification)	Accuracy, F1-Score	Integration of spatial and temporal feature extraction improved performance; the hybrid model faces computational and scalability challenges.
Wang [15]	LSTM with Self-Attention	User Experience Prediction	Accuracy, Improvement (%)	Self-attention mechanism increased accuracy by 6.1%, but faces constraints in real-time applicability and comparison with other models.
Alkenani and Nickray [16]	ConvRNN	Cyber-Attack Detection	Accuracy, Precision	ConvRNN outperformed LSTM, but computational burden is an issue in large-scale systems.
Pan and Ma [17]	CNN, LSTM	Mental Health Prediction	Accuracy (85.7%)	High accuracy, but limited to behavioral data, indicating the need for more varied features in future studies.

In spite of advances in text classification using neural networks, models such as CNNs and LSTMs have some limitations in processing complex tasks, for example, classifying customer grievances. CNNs can extract local patterns effectively, but there is an inability to process long-distance relationships. Conversely, LSTM can grasp long-distance relationships effectively, yet it tends to lose valuable local information. These challenges, therefore, make it difficult for individual models to categorize complex, unstructured information. The principal concept in this study, therefore, involves combining convolutional and LSTM layers into a hybrid ConvLSTM model to alleviate these challenges through learning local and long-distance relationships simultaneously. This presents a superior solution, bettering accuracy and efficiency in complex text classification work.

To guide the research and highlight the study's contributions, the following research questions have been formulated: 1) What is the performance of the proposed ConvLSTM model compared to separate CNN and LSTM models in terms of accuracy, precision, recall, and AUC in classifying customer complaints? What are the biggest strengths of merging convolutional layers and LSTM layers in a hybrid model in text classification tasks? 3) How can the ConvLSTM model be applied to practical, unstructured customer complaint datasets to boost classification accuracy and efficiency? These research questions outline weaknesses in prevailing methodologies and outline the original study contributions of the ConvLSTM model in terms of bettering text classification tasks involving complex and noisy data. The remaining structure of this article is given below: In section 2, the research methodology was presented. This section briefly describes the evaluation indices and algorithms used in this study. Section 3 presented the dataset used in this study, which is related to customer complaints. In section 4, the research results were analyzed and discussed using different evaluation indices and various plots. Finally, in the last section, the conclusion was presented.

2 Methodology

This study proposes a neural network-based model for classifying customer complaints. For this purpose, data pre-processing is done first. The first technique used for this purpose is data cleaning. Better preprocessing improves algorithm performance and classification quality. In this study, for the pre-processing of the text, first, all the words were converted into lowercase letters, but the overall structure of the sentences was not touched to avoid the loss of information. Also, the word embedding method is used to convert the text into numerical vectors.

Word embedding transforms words into dense vectors in a continuous space, enabling semantic comparison through measures like cosine similarity. This study uses a word-to-index mapping as the initial embedding step. Such embeddings capture semantic and syntactic patterns, improving model efficiency and performance. Their dense, fixed-length nature makes them effective in neural models. Recent advances, including contextualized embeddings such as BERT and other transformer-based models, generate context-aware vectors and have greatly improved performance in NLP tasks [18]. The article “Revisiting Word Embeddings in the LLM Era” presents a thorough comparative analysis of traditional embeddings, such as Word2Vec, in relation to contemporary embeddings generated by large language models. A recent survey, “Recent Advances in Universal Text Embeddings,” delineates how contemporary embedding techniques strive to generalize across many tasks and domains [19]. Additionally pertinent is “A Comprehensive Empirical Evaluation of Existing Word

Embedding Approaches," which examines the performance of various embedding approaches in classification tasks [20]. Ultimately, further specialized embedding techniques have been introduced. SA-Tweedie generates dense vector representations that rival BERT embeddings in named entity recognition tasks, while necessitating significantly fewer parameters [21,22].

Five different schemes based on NNs were used to classify data. Today, AI plays a vital role in solving various problems. One of the main branches of AI is NNs, which are modeled on the brain. In NNs, some nodes form layers in the network and connect different areas, just like neurons in the brain. The schemes used in this study include a one-dimensional convolutional model with one layer (1Conv1d), a one-dimensional convolutional model with two layers (2Conv1d), a one-dimensional convolutional model with three layers (3Conv1d), Long Short-Term Memory (LSTM) model and a Convolution LSTM Neural Network (ConvLSTM). In convolutional schemes, the count of layers represents the count of convolutional layers. In the LSTM model, instead of a convolutional layer, an LSTM layer with 100 units was used. Also, the Convolutional LSTM model includes Convolutional and LSTM layers.

The architecture of all the schemes used in this study consists of an embedding layer with 32 filters, a dense layer with 126 neurons, and a dense layer with six neurons (for classification). The RMSprop optimizer was used to increase the learning rate. By limiting fluctuations in the vertical direction, this optimizer enables the algorithm to take more significant steps in the horizontal direction, thereby increasing its speed. The learning rate is one of the hyperparameters that substantially impact the stochastic gradient descent (SGD) and make the model jump from a specific part of the data. Setting the correct value of the work learning rate is very important. RMSProp is one of the adaptive learning rate methods. This optimizer tunes the Adagrad method to a uniformly decreasing learning rate. This optimizer reduces the learning rate in Adagrad, which is monotonically decreasing [23,24]. Also, during the training, categorical cross-entropy was used for the loss function, which is used for multi-class classification and leads to creating a probability for each class. The loss function is a function that determines how wrong the model's predictions were. The algorithms that perform the act of classification can receive the table or the input matrix, and from this matrix and its features, they learn the pattern in each class. Then, if a new sample whose class is unknown is given to the algorithm that has learned, this algorithm can predict the class of the new sample.

The RMSprop optimizer was used due to its ability to adaptively adjust the learning rate and therefore prevent overshooting and improve convergence, especially in complex tasks like text classification. Compared with regular SGD, RMSprop adapts learning rates based on the magnitudes of gradients and therefore improves its performance with deep learning structures like ConvLSTM. A categorical cross-entropy loss function was used for multi-class classification to improve the model by minimizing the difference between the projected probability and target labels. Categorical cross-entropy, as opposed to binary cross-entropy, which works best with binary classification, works best with multi-class problems and allows for correct learning and classification with tasks like customer complaint classification.

This study selected RMSProp as the optimizer because of its efficacy in managing noisy or sparse data, rendering it appropriate for training deep models such as ConvLSTM. In contrast to conventional optimizers such as SGD or Adam, RMSProp adjusts the learning rate for each parameter according on recent gradient data, hence enhancing the stability of the training process and mitigating problems such as disappearing gradients. Its benefits encompass expedited convergence, especially in noisy datasets, and its capacity to sustain stability through learning rate adjustments, rendering it a pragmatic option for this text classification problem.

The choice of 1D CNN, LSTM, and ConvLSTM was predicated on their shown efficacy in text categorization, with 1D CNNs adept at capturing local features and LSTMs proficient in learning long-term dependencies. The ConvLSTM model integrates both components, rendering it optimal for tasks such as identifying customer complaints that necessitate both short- and long-term context. Transformer-based models and attention processes were excluded because of their substantial computational expense, which was impractical considering the dataset size and the necessity for a balance between performance and efficiency. The fully connected dense layer was incorporated to amalgamate features derived from the CNN and LSTM layers, encapsulating non-linear correlations between the features and the target classes. The selection of 126 neurons in the dense layer was empirically determined through experimentation and a grid search, yielding the optimal balance between model complexity and accuracy.

To guarantee an equitable and replicable assessment, the neural network models were trained with a mini-batch size of 64 samples, a constant learning rate of 0.001, and for 50 epochs. The RMSprop optimizer was selected for its ability to adaptively adjust learning rates for each parameter by utilizing a moving average of the squared gradients, hence stabilizing learning in sparse and noisy text data. Initial tests utilizing Adam and standard SGD resulted in unstable convergence and marginally reduced validation accuracy, while RMSprop consistently facilitated more rapid and stable convergence. Hyperparameters such as learning rate, batch size, number of neurons in dense layers, and dropout rate were investigated by a grid-search methodology on the validation set. The chosen configuration (learning rate = 0.001, batch size = 64, epochs = 50, dropout = 0.3 for ConvLSTM) achieved an optimal equilibrium between accuracy and generalization, preventing overfitting while reducing training duration. The tuning approach guaranteed that all models were evaluated under optimum and uniform training circumstances.

2.1 Evaluation indices

There are various criteria to evaluate the efficiency of ML algorithms. This study used five indexes, including Cosine

similarity, Auc, Accuracy, Precision, and Recall, to evaluate the algorithms explained below. Similarity criteria are distance criteria that determine how far or close two entities are. It is obvious that the similarity criteria have the opposite relationship with the distance criteria, and in other words, the higher the similarity, the smaller the distance between two objects [25]. Cosine similarity is one of the most widely used criteria in text processing and is determined from the following relationship [26,27]:

$$cos(x,y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2}\ \sqrt{\sum_{i=1}^{n}(y_i)^2}} \quad (1)$$

Where n is the count of observations, x_i and y_i are the ith values x and y. This index has a value between 1 and -1. When the observed values match the values predicted by the model, this index equals 1. This is when, in complete contrast between the observed and predicted values, this index equals -1.

The AUC (Area Under Curve) index is another commonly used measurement criterion obtained by calculating the area under the ROC (receiver operating characteristic) curve. The AUC (Area Under Curve) index is a performance measure based on variable threshold values for classification problems and is equal to the area under the ROC (Receiver Operating Characteristic) curve. This index measures the discriminability and gives us information about the model's ability to distinguish classes. In mathematical terms, this index can be obtained by drawing the ROC diagram whose axes are TPR (True Positive Rate) and FPR (False Positive Rate). The maximum value of this index is equal to 1, which shows that the observed values are in complete agreement with the predicted values [28,29]. Table 2 shows the confusion matrix used to get a more comprehensive picture of the model's performance.

This table enumerates the six categories of complaints utilized in this investigation, along with their respective class IDs. Class IDs are numerical designations allocated to each category to facilitate processing during model training. The numerical IDs were allocated according to their sequence in the dataset, with each class linked to a distinct integer value. "Credit reporting, credit repair services, or other personal consumer reports" is designated ID 0, whereas "Checking or savings account" is designated ID 5. The utilization of numerical identifiers facilitates the efficient management and categorization of data in machine learning algorithms.

Table 2: Confusion matrix related to the actual and predicted values

		Actual values	
		Positive	Negative
Predicted values	Positive	TP	FP
	Negative	FN	TN

The values are divided into two categories: actual value and predicted value. According to this matrix, true positive (TP), true negative (TN), false positive (FP), and false negative (FN) are obtained and used to define other evaluation indices. According to the confusion matrix, the values of the Accuracy, Precision, and Recall evaluation indexes are determined based on the following equations [30–33].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

These indices are the most common criteria for evaluating the efficiency of classification algorithms.

2.2 Classification schemes

This section introduces the classification schemes used in this study, all of which are based on machine learning methods.

Choice of 1D CNN, LSTM, and ConvLSTM designs involved their mutual strengths: CNN lends itself well to perceiving local features while LSTM performs particularly well at extracting long-term dependency. By combining both of these models through the ConvLSTM paradigm, we capitalize on both of their strengths to yield greater efficiency at tasks of text classification. Though transformer-based models like BERT produce better contextual insight, they consume more resources and are likely more than needed for tasks where local and sequential features are paramount. Incorporation of a fully connected dense layer with 126 neurons was empirically determined because this setting gave an ideal compromise of model simplicity and training efficiency and thus averted possible overfitting while still providing adequate capacity.

2.2.1 CNN

CNN is a prominent subset of deep learning (DL) that shares standard features with other neural networks, such as the MLP (Multi-Layer Perceptron). CNNs and MLPs are composed of neural layers that can learn and incorporate bias [34,35]. However, a fundamental distinction between MLPs and convolutional neural networks lies in their input structure. For example, for inputs with a matrix structure such as images, with the increase in the number of pixels, the number of neurons and layers in the MLP neural network structure also increases, subsequently increasing the volume and execution time. This is while CNN considers the entire input matrix without changing the input structure, and as a result, it will perform better in processing such inputs [36–38]. The main layers or blocks of a CNN network include the input layer, convolutional layer, non-linear activation function (non-linear activation function is usually shown together with the convolution layer), pooling layer, and fully connected layer, each of which will be described below [39,40].

1- Input layer

The noteworthy point is that the CNN network is suitable not only for 2D images but also for other input data, such as 1D to 4D data, which are also used in this network.

2- Convolution

One of the characteristics of CNN is that it uses convolution in at least one of the layers. In the convolution operator, there are four essential components, which are:

- Input matrix or image
- Convolution filter or kernel
- convolution operator (*)
- Convolution output feature (Output)

The convolution operator (*) involves using a convolutional kernel or filter, which is applied by sliding across the input image or matrix. Essentially, the kernel or filter traverses or scans the input image. It's important to mention the practice of 'padding' or adding layers to the matrix, which entails inserting additional rows and columns around the input matrix. This padding technique serves to prevent the reduction of the output dimension. The convolution layer is the main engine of a CNN, consisting of several filters, and the interaction between these filters and the input layer forms the network's output. Each filter should have a specific pattern, and the convolutional layer output should be a set of different patterns. Since the filters can be updated to train the CNN better, the need for manual filters is eliminated, giving us more flexibility in the number and relevance of filters applied to the dataset.

3- The non-linear layer in the convolutional neural network (Activation Function)

Like other NNs, the CNN uses the non-linear activation function after the convolutional layer. Using the non-linear function creates a non-linear property in the neural network, which is very important. Some frameworks allow you to specify the type of non-linear function in the exact convolution layer definition. In some frameworks, you have to create a separate layer. Defining the non-linear function separately from the convolutional layer provides more flexibility. The Rectified Linear Unit (ReLU) function is the most popular among all the nonlinear functions. Of course, there are other members of the ReLU family, such as PReLU, Leaky-ReLU, etc.

4- Pooling layer in the convolutional neural network

The role of this layer in CNN architecture is to eliminate the spatial size of feature gram. Also, mapping this layer is similar to convolution, and max and average pooling layers are commonly used [37–41].

2.2.2 LSTM

In the following years, LSTM was improved by many people. This network is a relatively old network. However, it is used in various issues and is still very popular. LSTMs are RNNs that remember and learn past information, which is why they can be used in time series forecasting. LSTMs have a chain-like structure in which four layers interact uniquely. The LSTM neural network, like the RNN network, is placed in a chain [42,43]. The RNN network has weaknesses, and LSTM is recommended to solve them. Significant weaknesses that cannot be ignored. RNN neural network has long-term dependency problems. That is, it cannot perform well in sentences, paragraphs, and all long sequences of data [44]As its name suggests, the LSTM network eliminates the problem of RNN networks by using long-term memory. Its architecture differs from RNN because one of its inputs is directly connected to the output. In this connection, the Cell State component is a key component in LSTM, also called Long-Term Memory [45,46].

2.2.3 Convolution LSTM Neural Network (ConvLSTM)

The ConvLSTM model, using convolutional and LSTM layers in the design of its network architecture, takes into account the abilities and capabilities of both mentioned algorithms. The ConvLSTM model is a combination of CNN and LSTM schemes. By using convolutional layers and LSTM in the design of its network architecture, this model takes into account the abilities and capabilities of both mentioned algorithms. In this model, CNN and LSTM learn local and long-term features. In ConvLSTM, low-level fixed features learned by CNN are given as input to the LSTM algorithm. LSTMs can understand the long-term dependence and input of the process chain. Finally, a prediction is made using a fully connected layer [47–49]. A dropout layer with a rate of 0.3 was incorporated after the dense layer in all neural network models to alleviate overfitting. L2 weight regularization was implemented in convolutional and LSTM layers to limit parameter expansion and enhance generalization. Training was overseen using validation loss to identify discrepancies between training and validation performance. The documented loss curves exhibited stable convergence without indications of overfitting, as validation loss continually declined in tandem with training loss until convergence was achieved.

3 Familiarity with data

The primary dataset used in this study is a collection of international complaints comprising 162,421 complaints, which can be accessed through https://www.kaggle.com/. Six different classes with the most significant number of samples were used in this study. In Table 3, the names of each class are shown along with their ID numbers.

Table 3: Classification of data along with their ID numbers

Class Name	Class ID
Credit reporting, credit repair services, or other personal consumer reports	0
Credit card or prepaid card	1
Debt collection	2
Student loan	3
Mortgage	4
Checking or savings account	5

Fig. 1 shows the frequency of classes. According to this figure, the frequency of complaints in different courses differs. The class "Credit reporting, credit repair services, or other personal consumer reports" has the highest frequency of data, with 54%. After that, the "Debt collection" and "Mortgage" classes were the most frequent, with 19% and 10%, respectively. Also, the "Student loan" class has the lowest frequency with 3%.

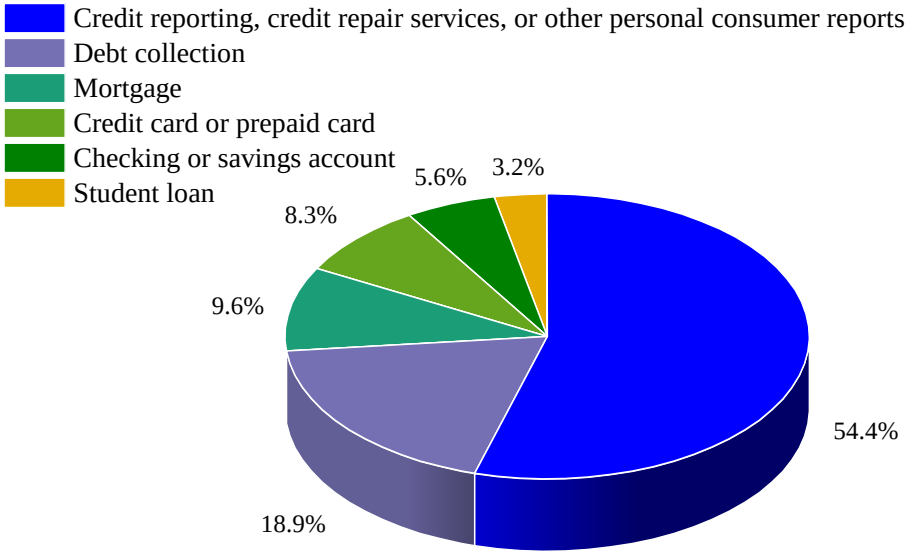


Figure 1: Pie chart related to the frequency of classes

This pie chart depicts the distribution of customer complaints across several categories within the dataset. The graphic illustrates the dataset's asymmetry, with the "Credit reporting, credit repair services, or other personal consumer reports" category accounting for the highest percentage of complaints (54%), followed by the "Debt collection" and "Mortgage" categories. This graphic depiction elucidates the dataset's composition and the necessity for methodologies to address imbalanced data during model training.

4 Results and discussion

In this part, a case study based on the introduced data is conducted to review and compare different schemes based on evaluation indicators. Table 4 shows the values of the evaluation indices related to each model.

Table 4: Evaluation indices related to different schemes

Model	Cosine similarity	Auc	Accuracy	Precision	Recall
1Conv1d	0.88398	0.908816	0.86476	0.88757	0.838884
2Conv1d	0.859871	0.883173	0.838731	0.880973	0.787628
3Conv1d	0.785571	0.80106	0.747091	0.875791	0.619698
LSTM	0.860561	0.888187	0.840185	0.878856	0.798385
ConvLSTM	0.896485	0.921991	0.880608	0.894993	0.864263

To compare more easily, in Fig. 2, the values of the evaluation indices are presented separately for the schemes. According to Table 4 and Fig. 2, the ConvLSTM model has the highest index values compared to other schemes. The values of Cosine similarity, AUC, Accuracy, Precision, and Recall indices of this model are respectively equal to 0.896485, 0.921991, 0.880608, 0.894993, and 0.864263, which are all higher than the corresponding values in other schemes. Therefore, the ConvLSTM model has performed best based on the evaluation indices. The ConvLSTM model exhibited the best performance, followed by the 1Conv1d model, LSTM, 2Conv1d, and finally, the 3Conv1d scheme, which performed the weakest among all the models. In other words, the 3Conv1d model has the least accuracy. Therefore, this model has the weakest performance among different schemes.

Alongside Cosine similarity, AUC, Accuracy, Precision, and Recall, the F1-score was computed to offer a more equitable evaluation, particularly due to the multi-class characteristics of the issue. The F1-score equilibrates precision and recall, especially for imbalanced classes, and is computed as the harmonic mean of these two metrics. The F1-scores for each model are as follows: ConvLSTM (0.86), 1Conv1D (0.82), LSTM (0.80), 2Conv1D (0.79), and 3Conv1D (0.71). An in-depth examination of the results indicated that, although ConvLSTM surpassed the other models in most measures, the 1Conv1D model demonstrated robust performance in situations involving smaller datasets, when training duration and computing economy were paramount. The ConvLSTM model, despite being computationally intensive, exhibited superior performance in extensive datasets and intricate situations. Subsequent trials,

including different dataset sizes and class imbalances, would yield further insights into the model's robustness across various applications.

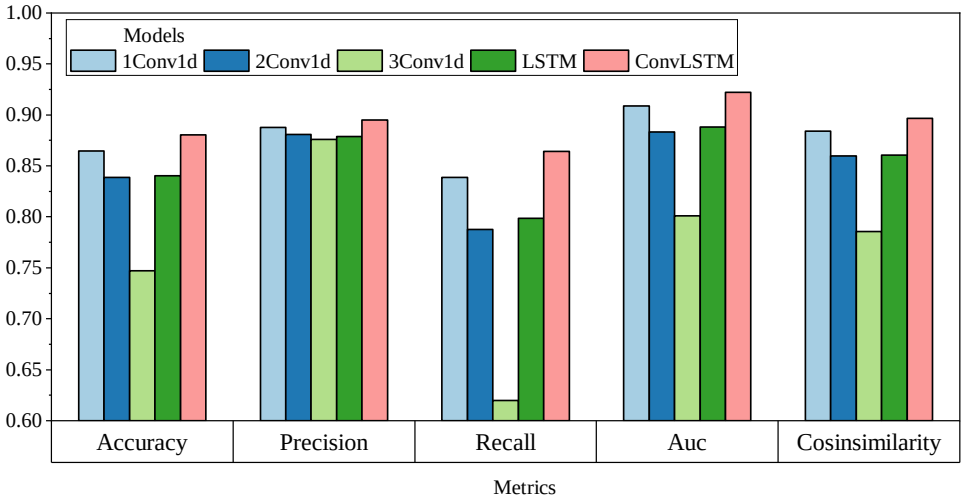


Figure 2: Bar plots related to the evaluation indexes

This graph indicates how effectively models perform on various parameters of evaluation, such as accuracy, precision, recall, and AUC. The image indicates that ConvLSTM tends to perform better than both CNN and LSTM models. ConvLSTM has the strengths of both CNN and LSTM, and it can extract local feature patterns through CNN and long-range dependencies through LSTM. CNNs can effectively detect local features but fail to grasp long-term dependencies and, therefore, perform lower than ConvLSTM models. Though LSTMs can effectively grasp long-term dependencies, they underperform ConvLSTM since they have difficulty in learning local features

In Fig. 3, the performance of different schemes based on the evaluation indicators is shown by drawing the evaluation index diagrams. As it is clear from this figure, in all the graphs, the point corresponding to the ConvLSTM model has the highest value, which shows its greater accuracy. The point corresponding to the 3Conv1d model has the lowest value, confirming its poor performance

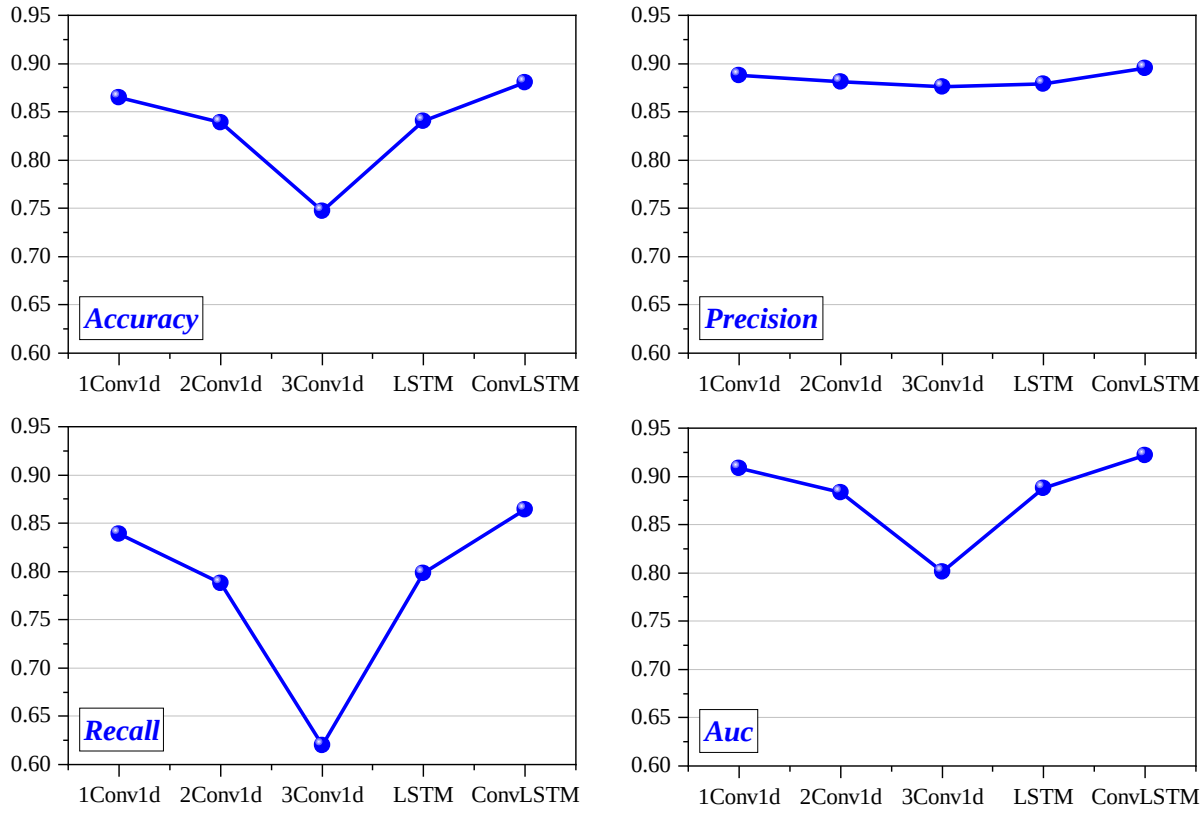


Figure 3: Comparison of different schemes based on the values of evaluation indexes

The balance between precision and recall of each model can be described through the precision-recall curve. The AUC of the ConvLSTM curve is much larger, showing it has a better balance between precision and recall at different scales. This means ConvLSTM distinguishes positive and negative data better, even in class imbalance situations. The areas under the curve of the CNN and LSTM models are smaller, showing they do not distinguish minor classes, which is one of the characteristics of imbalanced datasets.

Fig. 4 shows the curves related to the Precision index according to Recall for different schemes. As it is clear from this figure, ConvLSTM has taken the largest area under the curve. Meanwhile, 3Conv1d has the smallest area among various schemes. Therefore, ConvLSTM and 3Conv1d schemes have the best and weakest performance among other schemes, respectively

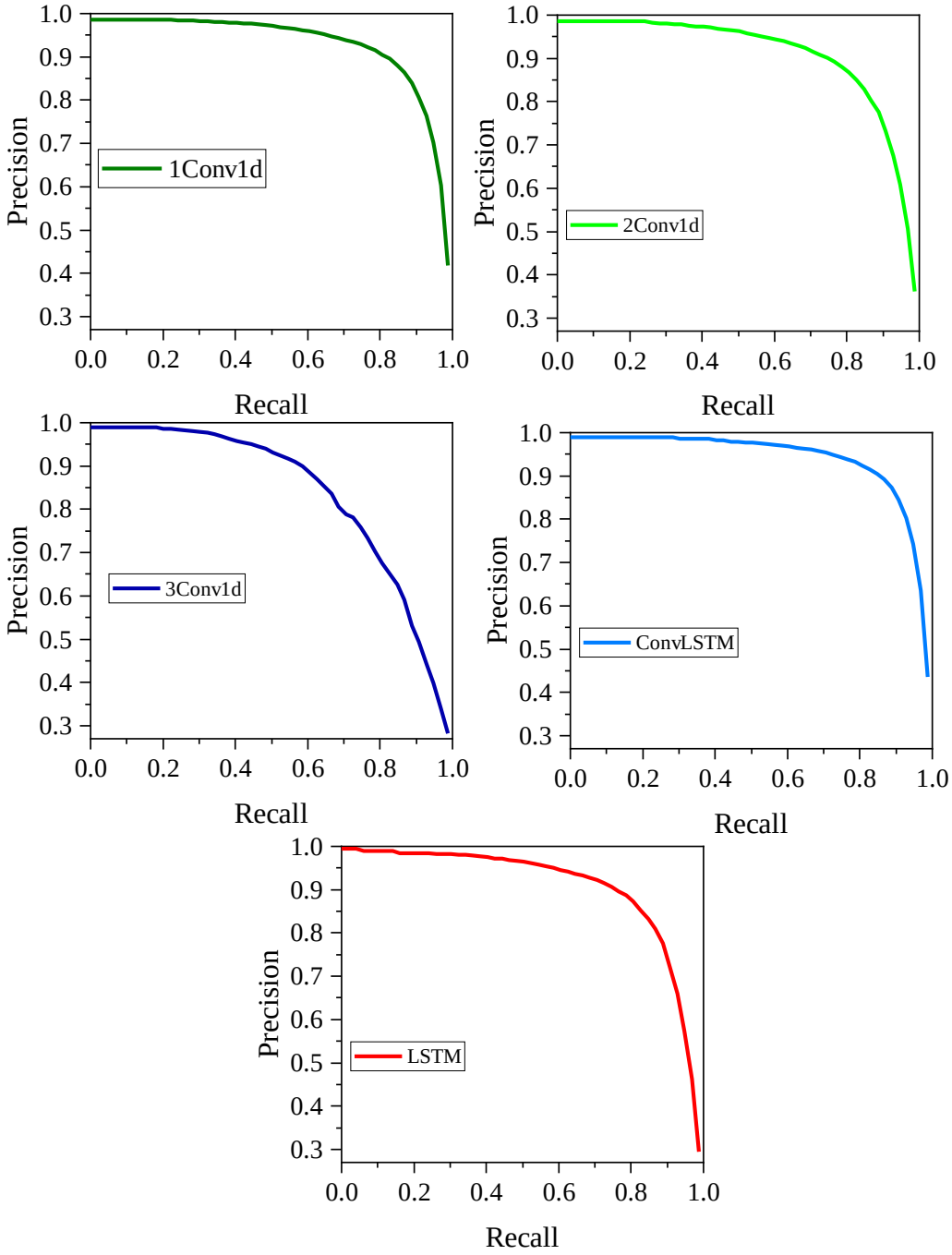


Figure 4: The curves related to the Precision index according to Recall for different schemes

demonstrated in Fig. 4 confirms that ConvLSTM achieves an optimal balance in terms of both recall and precision. Precision and recall of CNN and LSTM models are lower than this. CNN performs better in terms of recall but is inferior in terms of precision, whereas LSTM possesses average scores in both cases. ConvLSTM performs well due to this combination of local pattern detection (CNN) and long-distance sequential learning (LSTM) and can maintain high precision and raise recall significantly, particularly in cases of tough or minor

classes. This illustration depicts the Precision-Recall curve for the ConvLSTM, CNN, LSTM, and more models. The curves illustrate the trade-off between precision and recall for each method, with the area under the curve (AUC) reflecting the model's overall performance. A greater area under the curve indicates an improved equilibrium between precision and recall, with the ConvLSTM model demonstrating the biggest area, signifying its enhanced efficacy in managing both false positives and false negatives. The number offers significant insights into the algorithms' proficiency in accurately classifying customer complaints while reducing errors

In Fig. 5, the mosaic plot of the ConvLSTM model is shown as the best model. A mosaic plot is one of the most used charts in data visualization. The mosaic plot is the best option in cases with several variables and categories simultaneously, or if all those values are not numbers. The main feature of this plot is that both the width and height of the plot are variable. This plot provides an overview of the data and makes understanding the relationships between different variables possible. In this figure, the width of the classes on the horizontal axis shows the ratio of the number of samples that belong to a particular class. The width of the courses on the vertical axis shows the ratio of the number of samples assigned to the class predicted by the model. According to Fig. 6, it can be seen that the ConvLSTM model correctly predicted the observation classes with high accuracy. For each of the observed classes with a given color, the width of the class predicted by the model with the same color is the widest. Also, the color width of other classes has been reduced, which shows the acceptable and suitable performance of the mentioned model in the correct classification.

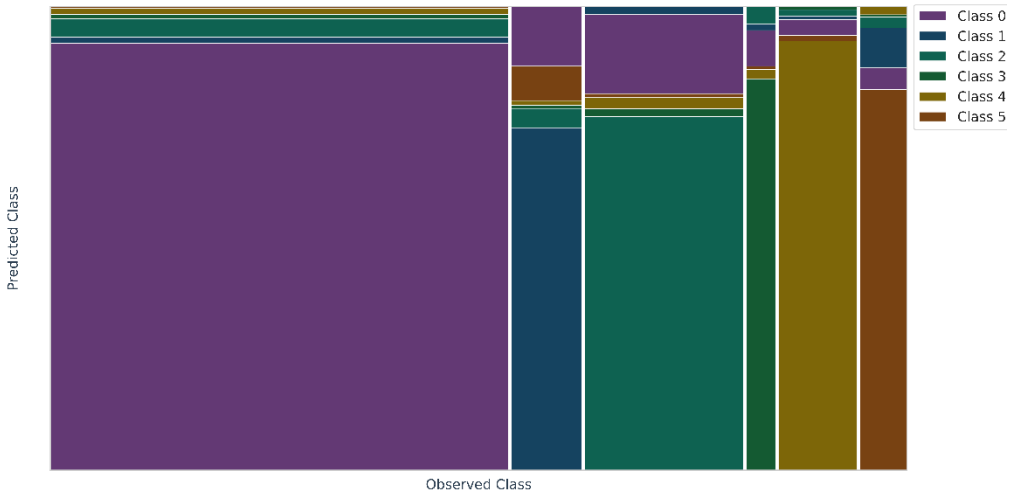


Figure 5: The mosaic plot of the ConvLSTM model

ConvLSTM performs excellently, as experts would have anticipated in text classification. It combines CNN and LSTM in one model, so it can apply the strengths of both approaches. The CNN excels at identifying local features such as n-grams or short sentences, and LSTM excels at long-range text relationships. The ConvLSTM model combines both strengths, and it performs superior to both CNN and LSTM for the most complicated text classification tasks. The outcomes confirm this argument, as ConvLSTM systematically outperformed both models in most of the measurement criteria.

Table 5 compares the ConvLSTM model with CNN and LSTM models. It highlights their benefits, disadvantages, performance measurements, and context with relevant studies. This shows why the ConvLSTM model performs better and how the combined approach reduces the weaknesses of separate CNN and LSTM models.

Table 5: Comparative Performance of ConvLSTM, CNN, and LSTM Models.

Model	Key Strengths	Limitations	Performance	Contextual Comparison with Related Works
ConvLSTM	Combines convolutional layers for local feature extraction and LSTM layers for capturing long-term dependencies	Increased computational complexity compared to standalone CNN or LSTM models	Best-performing in accuracy, precision, recall, AUC	Outperforms CNN and LSTM in all metrics. Addresses limitations of both models by capturing both local and global features.
CNN	Effective at capturing local patterns in data	Struggles with long-term dependencies in sequences	Moderate performance in accuracy and recall	Zhong et al. [7] and Singh et al. [8] found good results, but CNN alone struggles with sequential data and long-range dependencies.

LSTM	Excellent for learning long-term dependencies	Limited in capturing local features in text data	Moderate to good performance, lower than ConvLSTM	Ghazzawi and Alharbi [6] achieved good results but missed local feature extraction, which limits their performance.

The table presents the ConvLSTM, CNN, and LSTM models, highlighting their advantages and limitations, measurement indicators, and their relevance to studies in the Related Works section. ConvLSTM: This model integrates the strengths of both CNN and LSTM, and as such, it performs better in all cases. It captures local features through convolutional layers and long-distance dependencies through LSTM layers, and it performs better in text classification than both models. The accuracy, precision, and recall of ConvLSTM, as per results, exceed CNN and LSTM models, and it performs better in AUC as well. This integration-based approach alleviates individual limitations of CNN and LSTM models, and it can be used on complex assignments involving both local and sequential features. CNNs can extract local features effectively but have limitations in extracting long-distance features. Zhong et al.'s models [7] and Singh et al.'s models [8] demonstrate that CNNs can extract features effectively but have limitations in cases where the order of data (text) impacts classification effectively. Hence, ConvLSTM's combined design offers considerable advantage in terms of processing these assignments. LSTM models have strengths in terms of extracting long-distance connections but weaknesses in terms of local features, and these features are critical in most text categorization assignments. The study by Ghazzawi and Alharbi [6] indicates LSTM performs effectively; nevertheless, it fails in capturing the rich local context inherent in text data. ConvLSTM overhauls LSTM by unifying local and long-distance feature extraction.

To more effectively substantiate the intricacy of the ConvLSTM model, its performance was juxtaposed with conventional machine learning classifiers, such as SVM, Naïve Bayes, and Random Forest. The table below delineates the advantages, drawbacks, and efficacy of each approach.

Table 6: Baseline comparison between ConvLSTM and traditional machine learning models.

Model	Key Strengths	Limitations	Performance	Comparison with Other Models
ConvLSTM	Captures both local features and long-term dependencies	Higher computational cost, slower training, and inference times	Best in accuracy, precision, recall, AUC	Outperforms all other models in terms of accuracy, but is computationally expensive.
SVM	Effective in high-dimensional data, robust to overfitting	Slow for large datasets, not suitable for sequential data	Moderate performance	Faster training, but less accurate than ConvLSTM in text classification tasks.
Naïve Bayes	Simple, fast, good for large datasets	Assumes independence of features, less accurate for complex relationships	Lower performance	Very efficient but shows lower accuracy compared to ConvLSTM and other deep learning models.
Random Forest	Handles overfitting, robust to noise	Can be slow for large datasets, less interpretable	Moderate performance	Faster than ConvLSTM, but less accurate due to the inability to capture complex dependencies.

Table 6 contrasts the ConvLSTM model with conventional machine learning classifiers such as SVM, Naïve Bayes, and Random Forest. Although ConvLSTM attains superior performance regarding accuracy, precision, recall, and AUC, it incurs a greater computational expense relative to conventional models. Support Vector Machines and Random Forests provide expedited training and inference durations, albeit with a compromise in classification accuracy. Naïve Bayes exhibits superior computing efficiency; yet, it is less effective than ConvLSTM in processing intricate text data. This comparison substantiates the utilization of ConvLSTM for tasks where precision is favored over computational economy.

This work primarily examines a customer complaints dataset, although the architecture of the ConvLSTM model is intended to apply to other text classification applications. Nonetheless, the model has yet to be evaluated on datasets from alternative areas. Future research will investigate the model's efficacy on supplementary text datasets to assess its generalizability across various application domains. No particular approaches for domain adaptation were employed in this investigation. Domain adaptation may enhance the model's efficacy when utilized on content from different sectors or industries. Future studies could investigate techniques such as transfer learning or fine-tuning the model on domain-specific data to augment its applicability to various textual data sources.

An error analysis indicated that the "Debt collection" and "Mortgage" categories were often misclassified due to overlapping characteristics, including financial language. Systematic errors were identified in complaints, including imprecise or generic language that lacked adequate context. Misclassifications were associated with class imbalance, as underrepresented categories such as "Student loan" and "Checking or savings account" were erroneously classed as more prevalent categories like

"Credit reporting" and "Debt collection." Mitigating class imbalance and refining classification for ambiguous complaints could augment model efficacy.

Shapley additive explanations were utilized to improve the interpretability of the trained ConvLSTM model. The study identified significant terms and phrases that most influenced classification decisions across various complaint categories. Terms like "credit report," "loan," and "mortgage" exhibited the most significant beneficial impact on their respective categories. The results validated that the model's predictions were influenced by semantically pertinent phrases rather than arbitrary patterns, thus enhancing transparency and facilitating practical implementation.

All experiments were conducted on a workstation using an Intel Core i9-11900K CPU (3.5 GHz), 32 GB RAM, and an NVIDIA RTX-3080 GPU with 10 GB VRAM, utilizing TensorFlow 2.13. The ConvLSTM model necessitated approximately 72 minutes per epoch and converged in roughly 50 epochs, while the 1-Conv1D and LSTM models required approximately 38 minutes and 46 minutes per epoch, respectively, under identical conditions. Inference on the held-out test set, comprising approximately 30,000 samples, revealed that the ConvLSTM model achieved a throughput of roughly 2,300 samples per second, while the 1-Conv1D model obtained 3,800 samples per second and the LSTM model reached 3,100 samples per second. Consequently, ConvLSTM experienced an inference delay around 40 percent greater than that of the simpler systems. Despite ConvLSTM's higher computing requirements, its enhanced accuracy, precision, recall, and AUC warrant its implementation in contexts where predictive performance is prioritized over real-time limitations. These findings highlight the compromise between efficiency and performance in text categorization tasks. Subsequent research will explore model compression methodologies, including pruning and quantization, to reduce training and inference costs while maintaining robust predictive performance.

5 Conclusion

In general, the applications of text analysis in daily life are diverse and can be used in different fields, including educational environments, therapy, industry, etc. This research attempted to propose an accurate text classification model using text analysis techniques based on artificial intelligence (AI). For this purpose, a multi-class classification model was presented to classify textual data related to customer complaints. For this purpose, various schemes based on NN, including a one-dimensional convolutional model with one layer, two layers, and three layers, the LSTM model, and the LSTM-Convolutional model, were used. Also, during the processes, the RMSprop optimizer was used to increase the learning rate, categorical cross-entropy was used for loss during the training process, and word2index and word embedding methods were used to convert text into numerical vectors. Finally, the recommended schemes were examined and compared by conducting a case study based on an international dataset of customer complaints and using different evaluation indices. The results showed that the ConvLSTM model performed best based on the evaluation of the index. The values of the evaluation indices related to this model were all higher than the corresponding values in other schemes. Local and long-term features are learned by CNN and LSTM, respectively. Therefore, combining these two schemes allows the ConvLSTM model to consider both algorithms' abilities and capabilities. After that, the 1Conv1d, LSTM, 2Conv1d, and 3Conv1d schemes were placed in the following best-performing ranks. In other words, the 3Conv1d model has the least accuracy. Therefore, this model has the weakest performance among different schemes.

This work introduces the ConvLSTM model as an efficacious solution for multi-class classification tasks, exhibiting enhanced performance across multiple assessment measures. The model's capacity to integrate the advantages of convolutional and recurrent layers enables it to capture spatial and temporal information, rendering it especially fit for intricate data like text. Although the results are encouraging, the model's actual application may be constrained by issues such as computational expense and scalability, particularly when utilized with extensive datasets or real-time systems. Future research should investigate methods to mitigate these constraints, including the optimization of computing efficiency and the incorporation of supplementary NLP techniques to augment model performance. Furthermore, modifying ConvLSTM for real-time classification tasks may create new chances for its utilization in dynamic contexts, hence broadening its applicability across other fields.

Nomenclature

1Conv1d	A one-dimensional convolutional model with one layer.	n	The number of observations.
2Conv1d	A one-dimensional convolutional model with two layers is used.	NLP	Natural language processing
3Conv1d	A one-dimensional convolutional model with three layers is used.	NN	Neural network
AI	Artificial intelligence	PCA	Principal Component Analysis
AUC	Area Under Curve	ReLU	Rectified Linear Unit.
CNN	Convolutional Neural Network	RNN	Recurrent Neural Network
ConvLSTM	Convolution LSTM Neural Network.	ROC	Receiver Operating Characteristic
DL	Deep learning	TN	True Negative

FN	False Negative	TP	True Positive
FP	False Positive	TPR	True Positive Rate
FPR	False Positive Rate	x_i	The ith value of vector x.
LSTM	Long Short-Term Memory.	y_i	The ith value of vector y.
ML	Machine Learning		

Authorship contribution statement

Jing Yi: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Xiao Zeng: design of methodology

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] Gentzkow, M., B. Kelly and M. Taddy (2019). Text as data. *Journal of Economic Literature*, aeaweb, 57(3): 535–574. DOI: 10.1257/jel.20181020

[2] Hannigan, T.R., R.F.J. Haans, K. Vakili, H. Tchalian, V.L. Glaser, M.S. Wang, S. Kaplan and P.D. Jennings (2019). Topic modeling in management research: Rendering new theory from textual data. *Academy of Management Annals*, Academy of Management, 13(2): 586–632. https://doi.org/10.5465/annals.2017.0099

[3] Kowsari, K., K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes and D. Brown (2019). Text classification algorithms: A survey. *Information*, MDPI, 10(4): 150. https://doi.org/10.3390/info10040150

[4] Hartmann, J., J. Huppertz, C. Schamp and M. Heitmann (2019). Comparing automated text classification methods. *International Journal of Research in Marketing*, Elsevier, 36(1): 20–38. https://doi.org/10.1016/j.ijresmar.2018.09.009

[5] Chakraborty, B. and T. Bhattacharjee (2020). A review on textual analysis of corporate disclosure according to the evolution of different automated methods. *Journal of Financial Reporting and Accounting*, Emerald, 18(4): 757–777. https://doi.org/10.1108/JFRA-02-2020-0047

[6] Ghazzawi, A. and B. Alharbi (2019). Analysis of customer complaints data using data mining techniques. *Procedia Computer Science*, Elsevier, 163: 62–69. https://doi.org/10.1016/j.procs.2019.12.087

[7] Zhong, B., X. Xing, P. Love, X. Wang and H. Luo (2019). Convolutional neural network: Deep learning-based classification of building quality problems. *Advanced Engineering Informatics*, Elsevier, 40: 46–57. https://doi.org/10.1016/j.aei.2019.02.009

[8] Singh, R.P., R. Haque, M. Hasanuzzaman and A. Way (2020). Identifying complaints from product reviews: A case study on hindi, In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, ACL Anthology, pp: 108–116.

[9] Muqorobin, M., S. Rokhmah, I. Muslihah and N.A.R. Rais (2020). Classification of Community Complaints Against Public Services on Twitter. *International Journal of Computer and Information System (IJCIS)*, NELITI, 1(1): 7–10. https://doi.org/10.29040/ijcis.v1i1.6

[10] Hong, J., H. Kang and T. Hong (2020). Oversampling-based prediction of environmental complaints related to construction projects with imbalanced empirical-data learning. *Renewable and Sustainable Energy Reviews*, Elsevier, 134: 110402. https://doi.org/10.1016/j.rser.2020.110402

[11] Osorio-Arjona, J., J. Horak, R. Svoboda and Y. García-Ruíz (2021). Social media semantic perceptions on Madrid Metro system: Using Twitter data to link complaints to space. *Sustainable Cities and Society*, Elsevier, 64: 102530. https://doi.org/10.1016/j.scs.2020.102530

[12] Shin, J., S. Son and Y. Cha (2022). Spatial distribution modeling of customer complaints using machine learning for indoor water leakage management. *Sustainable Cities and Society*, Elsevier, 87: 104255. https://doi.org/10.1016/j.scs.2022.104255

[13] Shaukat, A. and U. Saif (2022). NLP based Model for Classification of Complaints: Autonomous and Intelligent System, In *2022 2nd International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, IEEE, pp: 1–6.

[14] Nemer, Z.N., S.A.W. Saddam, R.J. AL-Sukeinee and E.J. Harfash (2025). A Comparative Performance Analysis of CNN, LSTM, and CNNLSTM Models for Classifying Sleep Stages Using Electroencephalography Signals. *Informatica*, informatica.si, 49(28). https://doi.org/10.31449/inf.v49i28.8082

[15] Wang, L (2025). Analysis Method of User Experience Influencing Factors Based on Improved LSTM Algorithm Technology. *Informatica*, informatica.si, 49(24). https://doi.org/10.31449/inf.v49i24.7406

[16] Alkenani, J. and M. Nickray (2025). Enhancing Network QoS via Attack Classification Using Convolutional Recurrent Neural Networks. *Informatica*, informatica.si, 49(2). https://doi.org/10.31449/inf.v49i2.7637

[17] Pan, L. and H. Ma (2025). A Computational CNN-LSTM-Based Mental Health Consultation System in a College Environment. *Informatica*, informatica.si, 49(10). https://doi.org/10.31449/inf.v49i10.7136

[18] Gardazi, N.M., A. Daud, M.K. Malik, A. Bukhari, T. Alsahfi and B. Alshemaimri (2025). BERT applications in natural language processing: a review. *Artificial Intelligence Review*, Springer, 58(6): 1–49. https://doi.org/10.1007/s10462-025-11162-5

[19] Mahajan, Y., M. Freestone, N. Bansal, S. Aakur and S.K.K. Santu (2024). Revisiting word embeddings in the LLM era. *ArXiv Preprint ArXiv:2402.11094*, Acl Anthology.

[20] Cao, H (2024). Recent advances in universal text embeddings: A Comprehensive Review of Top-Performing Methods on the MTEB Benchmark. *Preprint, ArXiv*. Https://Doi. Org/10.48550/ArXiv, 2406, Cornell University. https://doi.org/10.48550/arXiv.2406.01607

[21] Kim, T. and H. Wang (2025). Global dense vector representations for words or items using shared parameter alternating Tweedie model. *Mathematics*, MDPI, 13(4): 612. https://doi.org/10.3390/math13040612

[22] Zaland, O., M. Abulaish and M. Fazil (2023). A Comprehensive Empirical Evaluation of Existing Word Embedding Approaches. *ArXiv Preprint ArXiv:2303.07196*, Cornell University. https://doi.org/10.48550/arXiv.2303.07196

[23] Zeiler, M.D (2012). ADADELTA: an adaptive learning rate method. *ArXiv Preprint ArXiv:1212.5701*, Cornell University. https://doi.org/10.48550/arXiv.1212.5701

[24] Smith, L.N (2017). Cyclical learning rates for training neural networks, In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, pp: 464–472. https://doi.org/10.1109/WACV.2017.58

[25] Lahitani, A.R., A.E. Permanasari and N.A. Setiawan (2016). Cosine similarity to determine similarity measure: Study case in online essay assessment, In *2016 4th International Conference on Cyber and IT Service Management*, IEEE, pp: 1–6. https://doi.org/10.1109/CITSM.2016.7577578

[26] Thongtan, T. and T. Phienthrakul (2019). Sentiment classification using document embeddings trained with cosine similarity, In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, ACL Anthology, pp: 407–414. https://doi.org/10.18653/v1/P19-2057

[27] Li, B. and L. Han (2013). Distance weighted cosine similarity measure for text classification, In *Intelligent Data Engineering and Automated Learning–IDEAL 2013: 14th International Conference, IDEAL 2013, Hefei, China, October 20-23, 2013. Proceedings 14*, Springer, pp: 611–618. https://doi.org/10.1007/978-3-642-41278-3_74

[28] Narkhede, S (2018). Understanding auc-roc curve. *Towards Data Science*, 26(1): 220–227.

[29] Walter, S.D (2005). The partial area under the summary ROC curve. *Statistics in Medicine*, Wiley Online Library, 24(13): 2025–2040. https://doi.org/10.1002/sim.2103

[30] Juba, B. and H.S. Le (2019). Precision-recall versus accuracy and the role of large data sets, In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI, pp: 4039–4048. https://doi.org/10.1609/aaai.v33i01.33014039

[31] Alvarez, S.A (2002). An exact analytical relation among recall, precision, and classification accuracy in information retrieval. *Boston College, Boston, Technical Report BCCS-02-01*, Academia, 1–22.

[32] Powers, D.M.W (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *ArXiv Preprint ArXiv:2010.16061*, Cornell University.

[33] Susmitha, I. and V. Sivasamy (2022). Detection of forest fire using support vector machine in comparison with random forest to measure accuracy, precision and recall, In *2022 14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics (MACS)*, IEEE, pp: 1–6. https://doi.org/10.1109/MACS56771.2022.10022546

[34] Habib, G. and S. Qureshi (2022). Optimization and acceleration of convolutional neural networks: A survey. *Journal of King Saud University-Computer and Information Sciences*, Elsevier, 34(7): 4244–4268. https://doi.org/10.1016/j.jksuci.2020.10.004

[35] Ansari, M.A., A. Crampton, R. Garrard, B. Cai and M. Attallah (2022). A Convolutional Neural Network (CNN) classification to identify the presence of pores in powder bed fusion images. *The International Journal of Advanced Manufacturing Technology*, Springer, 120(7): 5133–5150. https://doi.org/10.1007/s00170-022-08995-7

[36] Valizadeh, M. and S.J. Wolff (2022). Convolutional Neural Network applications in additive manufacturing: A review. *Advances in Industrial and Manufacturing Engineering*, Elsevier, 4: 100072. https://doi.org/10.1016/j.aime.2022.100072

[37] Ahlawat, S., A. Choudhary, A. Nayyar, S. Singh and B. Yoon (2020). Improved handwritten digit recognition using convolutional neural networks (CNN). *Sensors*, MDPI, 20(12): 3344. https://doi.org/10.3390/s20123344

[38] Panahi, M., N. Sadhasivam, H.R. Pourghasemi, F. Rezaie and S. Lee (2020). Spatial prediction of groundwater potential mapping based on convolutional neural network (CNN) and support vector regression (SVR). *Journal of Hydrology*, Elsevier, 588: 125033. https://doi.org/10.1016/j.jhydrol.2020.125033

[39] Paul, E. and R.S. Sabeenian (2022). Modified convolutional neural network with pseudo-CNN for removing nonlinear noise in digital images. *Displays*, Elsevier, 74: 102258. https://doi.org/10.1016/j.displa.2022.102258

[40] WHIG, P (2022). More on convolution neural network CNN. *International Journal of Sustainable Development in Computing Science*, 4(1).

[41] Hakim, W.L., F. Rezaie, A.S. Nur, M. Panahi, K. Khosravi, C.-W. Lee and S. Lee (2022). Convolutional neural network (CNN) with metaheuristic optimization algorithms for landslide susceptibility mapping in Icheon, South Korea. *Journal of Environmental Management*, Elsevier, 305: 114367. https://doi.org/10.1016/j.jenvman.2021.114367

[42] Le, X.-H., H.V. Ho, G. Lee and S. Jung (2019). Application of long short-term memory (LSTM) neural network for flood forecasting. *Water*, MDPI, 11(7): 1387. https://doi.org/10.3390/w11071387

[43] Sahoo, B.B., R. Jha, A. Singh and D. Kumar (2019). Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting. *Acta Geophysica*, Springer, 67(5): 1471–1481. https://doi.org/10.1007/s11600-019-00330-1

[44] Sherstinsky, A (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, Elsevier, 404: 132306. https://doi.org/10.1016/j.physd.2019.132306

[45] Song, X., Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang and Z. Cheng (2020). Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model. *Journal of Petroleum Science and Engineering*, Elsevier, 186: 106682. https://doi.org/10.1016/j.petrol.2019.106682

[46] Hunt, K.M.R., G.R. Matthews, F. Pappenberger and C. Prudhomme (2022). Using a long short-term memory (LSTM) neural network to boost river streamflow forecasts over the western United States. *Hydrology and Earth System Sciences*, HESS, 26(21): 5449–5472. https://doi.org/10.5194/hess-26-5449-2022

[47] Hu, W.-S., H.-C. Li, L. Pan, W. Li, R. Tao and Q. Du (2020). Spatial–spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, 58(6): 4237–4250. https://doi.org/10.1109/TGRS.2019.2961947

[48] Jiang, W (2022). Internet traffic matrix prediction with convolutional LSTM neural network. *Internet Technology Letters*, Wiley Online Library, 5(2): e322. https://doi.org/10.1002/itl2.322

[49] Zhao, C., P. Zhang, J. Zhu, C. Wu, H. Wang and K. Xu (2019). Predicting tongue motion in unlabeled ultrasound videos using convolutional LSTM neural networks, In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, pp: 5926–5930. https://doi.org/10.1109/ICASSP.2019.8683081

Application of Machine Learning Algorithms for Anomaly Detection in Cybersecurity Threat Mitigation

Kim Son Lim[1], Shih Yin Ooi[1,2*], Yee Jian Chew[1,2], and Md Shohel Sayeed[1,3]
[1]Faculty of Information Science and Technology, Multimedia University, Jalan Ayer Keroh Lama, Melaka 75450, Malaysia
[2]Centre for Advanced Analytics (CAA), COE for Artificial Intelligence, Multimedia University, Jalan Ayer Keroh Lama, Melaka, 75450, Malaysia
[3]Centre for Intelligent Cloud Computing (CICC), COE for Advanced Cloud, Multimedia University, Jalan Ayer Keroh Lama, Melaka, 75450, Malaysia
E-mail: 1181402836@student.mmu.edu.my, syooi@mmu.edu.my, chewyeejian@mmu.edu.my, shohel.sayeed@mmu.edu.my
* Corresponding author

Keywords: artificial intelligence, cybersecurity, threat detection, machine learning, anomaly detection, adversarial attacks, ransomware, incident response, data security, predictive modelling

Received: July 5, 2025

The integration of Artificial Intelligence (AI) into cybersecurity has transformed the landscape of threat detection, analysis, and mitigation. As cyber-attacks become increasingly sophisticated and evasive, traditional rule-based defences are no longer sufficient to identify zero-day exploits and advanced persistent threats. AI-driven approaches, leveraging machine learning and deep learning, enable proactive anomaly detection, behavioural modelling, and predictive analytics that enhance both the accuracy and agility of cyber defence mechanisms.
This paper provides a comprehensive examination of AI applications in cybersecurity, spanning anomaly detection, automated incident response, and adaptive defence frameworks. It also emphasizes the emerging role of AI in vulnerability management, where predictive modelling, natural language processing, and automated remediation are used to identify, prioritize, and mitigate vulnerabilities before they can be exploited. A real-world case study of Panasonic's VERZEUSE™ platform is presented to illustrate the industrial implementation of AI-enhanced cybersecurity. The platform exemplifies how AI-based predictive analytics, threat intelligence integration, and continuous monitoring can strengthen risk management and compliance in complex IT and IoT ecosystems.
The findings demonstrate that AI substantially improves detection accuracy, response speed, and proactive defence capabilities. However, challenges related to data quality, model robustness, interpretability, and ethical deployment must be addressed to ensure trustworthy adoption. The study concludes that the future of cybersecurity depends on harmonizing human expertise with adaptive AI systems to achieve resilient, self-learning defence frameworks.

Povzetek: Članek prikazuje, kako umetna inteligenca izboljšuje kibernetsko varnost, hkrati pa opozarja na ključne izzive pri njeni uporabi.

1 Introduction

1.2 Overview of the role of artificial intelligence in cybersecurity

Cybersecurity has become a strategic and operational priority for organizations globally due to the proliferation of interconnected systems, cloud services, and IoT devices. As the digital landscape expands, so too does the attack surface available to malicious actors. Modern threats range from conventional malware to sophisticated, adaptive techniques such as advanced persistent threats (APTs), zero-day exploits, and ransomware campaigns [1], [2]. These attacks can result in severe consequences, including data breaches, financial loss, operational disruption, and reputational damage.

Traditional cybersecurity approaches, primarily rule-based systems and signature-based detection, lack adaptive and predictive capabilities when confronting rapidly evolving or previously unknown attack vectors. Static defense mechanisms are inadequate in the face of adversaries who leverage automation, obfuscation, and AI-enhanced attack techniques [3], [4].

To meet these challenges, cybersecurity strategies increasingly incorporate Artificial Intelligence (AI) and Machine Learning (ML) to bolster defence capabilities. AI enables real-time threat detection, behavioural analysis, and predictive modelling, providing security systems with the ability to identify anomalies and respond autonomously [1], [5]. Unlike conventional tools, AI-

driven systems can analyse vast amounts of historical and real-time data, detect emerging patterns indicative of malicious activity, and continuously improve their performance over time.

This section presents an overview of the role of AI in cybersecurity, focusing on its application in threat intelligence, intrusion detection, behavioural analytics, and adaptive response mechanisms.

1.3 Importance of detecting and mitigating cyber threats

The failure to detect and mitigate cyber threats in a timely manner can lead to severe consequences for both public and private sector organizations. Cyber incidents such as data breaches may result in the unauthorized disclosure of sensitive information, privacy violations, operational disruptions, financial losses, and reputational damage [6]. In critical infrastructure environments, undetected attacks can compromise essential services, disrupt business continuity, and pose risks to public safety [7].

Recent studies indicate that the global average cost of a data breach has reached millions of dollars per incident, further emphasizing the financial implications of inadequate cyber defences [8]. Consequently, rapid and proactive threat detection has become imperative. Organizations must adopt mechanisms capable of identifying and mitigating cyber threats in their early stages to minimize impact and ensure resilience.

Failure to implement timely countermeasures can erode stakeholder trust, undermine competitive advantage, and in severe cases, jeopardize the survival of the affected entity [9]. Advanced cybersecurity strategies, including AI-enhanced detection systems, are critical in maintaining situational awareness and achieving operational continuity in an increasingly hostile digital environment.

1.4 Objectives

This study aims to examine the diverse applications of AI in the field of cybersecurity, with the goal of enhancing threat detection, response, and mitigation capabilities. By exploring a range of AI-driven methodologies, such as machine learning, behavioural analytics, and anomaly detection. This research seeks to illustrate how AI technologies contribute to strengthening organizational cyber defence mechanisms [9], [10].

1.5 The key area of focus in ai-drive cybersecurity

This study highlights several critical domains in which AI contributes to enhancing cybersecurity capabilities:

1. **Anomaly and behaviour detection**: AI, particularly through ML algorithms, is leveraged to monitor and identify abnormal patterns in network traffic, user behaviour, and system activities. These approaches enable early detection of potentially malicious actions that traditional rule-based systems may overlook [11], [12].
2. **Automated defence and response**: AI supports the automation of incident response, intrusion prevention, and vulnerability management. By enabling faster decision-making and dynamic adaptation to threats, AI improves the speed and effectiveness of cybersecurity operations [13].
3. **Challenges and risks**: Despite its benefits, the deployment of AI in cybersecurity faces several challenges, including the need for high-quality labelled data, the risk of adversarial attacks targeting AI models, and ongoing concerns surrounding ethics, privacy, and regulatory compliance [14].
4. **Emerging techniques**: Advanced AI paradigms such as reinforcement learning, federated learning, and integration with blockchain technologies are gaining traction as future-forward strategies to increase cyber resilience and secure distributed environments [15], [16].

By exploring these dimensions, this research provides insights into how AI technologies can be strategically applied to help organizations address evolving cyber threats and more effectively safeguard digital assets.

2 Cybersecurity threat landscape

The cybersecurity threat landscape is in a state of continuous evolution, presenting organizations across various industries with an expanding spectrum of sophisticated cyber-attacks that pose significant operational and financial risks [2,17]. Prominent categories of cyber threats include:

a. **Phishing:** A social engineering attack vector aimed at deceiving individuals into divulging sensitive data or inadvertently executing malicious software, often without the user's knowledge [18].
b. **Malware:** A broad classification of malicious software—encompassing viruses, worms, Trojans, and spyware—designed to infiltrate, disrupt, or damage computer systems without authorization [19].
c. **Ransomware:** A subclass of malware that encrypts critical data assets and extorts payment from victims in exchange for decryption keys, causing substantial operational disruption and financial loss [20].
d. **Distributed Denial-of-Service (DDoS) Attacks:** Coordinated attempts to incapacitate network services by overwhelming target systems with a flood of illegitimate traffic originating from multiple distributed sources [21].
e. **Insider threats:** Malicious or negligent activities perpetrated by authorized users or employees that result in unauthorized data access, exfiltration, or system compromise [22].
f. **Advanced Persistent Threats (APTs):** Highly sophisticated, often state-sponsored cyber campaigns characterized by prolonged and targeted intrusion efforts designed to covertly

extract sensitive information from high-value organizations [23].

The sophistication and frequency of these cyber threats have escalated markedly, severely impacting critical sectors such as finance, healthcare, government, and infrastructure. The increasing digital dependency of organizations amplifies the potential consequences of these threats, jeopardizing financial stability, operational continuity, and corporate reputation [24].

2.1 Emergence of advanced threats and associated challenges

The rising complexity of cyber-attacks is fundamentally reshaping cybersecurity paradigms. Polymorphic malware, capable of dynamically altering its code to evade signature-based detection, exemplifies a formidable challenge to traditional defence mechanisms [25], [26]. Compounding this issue, zero-day vulnerabilities—previously undisclosed security flaws lacking available patches—enable attackers to compromise systems prior to the deployment of defensive countermeasures [27].

Moreover, adversaries are increasingly exploiting automated tools and AI to enhance the precision, scale, and efficacy of cyber-attacks. AI-driven methodologies facilitate the orchestration of sophisticated phishing campaigns, development of adaptive stealth malware, and coordination of complex multi-vector assaults [28], [29]. This technological escalation underscores an urgent imperative for cybersecurity frameworks to adopt adaptive, intelligence-driven defences capable of countering evolving threats.

2.2 Limitations of traditional cybersecurity solutions

Conventional cybersecurity approaches predominantly relying on static rules and signature-based detection face critical limitations in addressing the dynamic and rapidly evolving threat landscape [30]. These reactive methods frequently fail to detect novel or polymorphic attacks that continuously evolve to bypass fixed detection criteria [31]. Consequently, traditional solutions are insufficient for proactive threat identification and mitigation.

AI has emerged as a pivotal enabler, imparting cybersecurity systems with adaptive, predictive, and comprehensive capabilities [32]. AI-driven platforms enhance anomaly detection, early threat identification, and automated response mechanisms, thereby substantially improving organizational security postures [33]. Given the fluidity of cyber threats, the integration of AI is essential to maintain resilient defences and safeguard sensitive information assets [34].

3 Artificial intelligence in threat detection

The increasing sophistication of cyber threats necessitates the deployment of AI to augment threat detection efficacy. AI-powered systems can analyse vast volumes of heterogeneous security data in real-time, discerning novel attack patterns and anomalies beyond the scope of traditional methods [35], [36]. This capability markedly improves detection accuracy and latency, empowering organizations to adopt a proactive cybersecurity stance [37].

3.1 Machine learning for anomaly detection

A cornerstone of AI-enabled threat detection is the application of machine learning algorithms to identify anomalous behaviours indicative of security incidents. Machine learning techniques in cybersecurity are conventionally classified into three categories [38], [39]:

- **Supervised learning:** Models are trained on labelled datasets containing both benign activities and known cyber-attacks, enabling classification of future observations based on learned discriminative features.
- **Unsupervised learning:** Algorithms analyse unlabelled data to detect deviations from normative behavioural baselines, facilitating identification of previously unseen or unknown threats.
- **Semi-supervised learning:** A hybrid approach that leverages both labelled and unlabelled data to enhance detection performance, particularly valuable when labelled data is limited [40].

By leveraging these methodologies, AI-driven cybersecurity systems can detect anomalous activities spanning networks, user behaviours, and host systems, thereby identifying threats potentially missed by signature-based defences [41].

3.2 Behavioural analysis and pattern recognition

Building on anomaly detection, AI systems monitor user and system behaviours—such as login patterns, data access, and network utilization—to establish behavioural baselines. Using clustering and classification techniques, deviations from these baselines are identified as potential indicators of compromise, enabling timely threat detection and response.

3.3 Natural Language Processing (NLP) for phishing detection

AI techniques powered by Natural Language Processing (NLP) have demonstrated efficacy in detecting phishing attempts by analysing linguistic cues within emails and electronic communications. NLP models can identify deceptive language patterns and suspicious contextual signals, facilitating rapid mitigation of phishing campaigns which often serve as entry points for broader cyber-attacks.

4 Real-time threat detection with deep learning

Deep learning—an advanced subset of machine learning utilizing neural networks—has enhanced AI-based threat detection by enabling the identification of polymorphic malware and complex attack vectors through multi-layered feature extraction from network and system telemetry. Organizations employing deep learning approaches benefit from reduced detection latency and improved threat identification accuracy, thus mitigating potential damages [42].

4.1 AI in threat mitigation

Effective cybersecurity extends beyond detection to timely mitigation. AI enhances incident response by automating threat containment and remediation actions.

4.2 Automated incident response and threat remediation

AI outputs that maps to actionable runbooks and remediation steps not just alerts are integrated with existing playbook as part of the SIEM SOAR functions. AI-enabled security orchestration, automation, and response (SOAR) platforms rapidly detect incidents and autonomously execute predefined remediation workflows, such as quarantining infected devices, blocking malicious IP addresses, and isolating suspicious files, thereby reducing dwell time and limiting attack impact.

4.3 AI-Enhanced Intrusion Prevention Systems (IPS)

Integrating AI with Intrusion Prevention Systems (IPS) enables dynamic threat recognition and blocking capabilities beyond static signature rules. Machine learning models continuously analyse network and system data to identify and respond to emerging threats, adapting security postures in real-time.

4.4 Role of AI in vulnerability management

AI assists in proactive vulnerability management by scanning organizational IT assets to identify exploitable weaknesses. Through risk scoring based on exploitability and impact, AI prioritizes remediation efforts, enabling cybersecurity teams to focus resources on the most critical vulnerabilities.

- Automated discovery and classification where AI-driven scanner leverage ML and NLP to analyse system configurations, code repositories, and threat feeds to automatically detect new vulnerabilities even before CVEs are published. Pattern recognition for detecting zero-day or configuration based vulnerabilities that traditional signature-based tools might miss. AI-algorithms cluster similar vulnerabilities, reducing duplicate finding and streamlining analyst review.
- Risk prioritization and predictive scoring where all model combine assets criticality, exploit availability, threat intelligence, and business context to assign dynamic risk score. Predictive models assess which vulnerabilities are most likely to be exploited in the near future, enabling teams to patch what matters the most. The approach goes beyond CVSS scoring by factoring in real-world exploitation trends.
- Intelligence patch and remediate planning where AI can recommend or automate patch deployment sequences by evaluating dependencies, system uptime windows and potential service impacts. ML analyses past remediation patterns to predict patch success probability and minimize operational disruptions. Integration with SOAR platform allows AI to tigger patch workflow to predict patch workflow automatically for low-risk systems.
- Continuous monitoring and adaptive defence continuously ingest logs and telemetry to verify if patched systems remain secure or if exploit attempts persist. Behavioural analytics can flag regression vulnerabilities or partial remediations that conventional checks may overlook. Predictive Analytics for Proactive Cyber defence. Adaptive learning ensures detection models evolve alongside the threat landscape
- Threat intelligence correlation correlates external threat feeds, dark web data, and CVE disclosures to identify vulnerabilities actively discussed or weaponized by attackers. This correlation supports proactive vulnerability mitigation, turning raw data into actionable intelligence.

AI-driven predictive analytics synthesize historical and current threat intelligence to forecast likely attack vectors, empowering organizations to pre-emptively strengthen defences and reduce exposure.

5 Challenges of AI in cybersecurity

Despite its transformative potential, AI adoption in cybersecurity faces notable challenges:

a. **Data quality and availability**
   The effectiveness of AI models hinges on access to extensive, high-quality datasets representative of evolving threats. The dynamic nature of cyber threats complicates data collection, annotation, and timely updating, potentially degrading model performance. Data quality and label scarcity models needs good, representative labelled events, and noisy logs lad to false positives and analyst fatigue.
b. **Adversarial attacks on AI models**
   AI systems themselves are vulnerable to adversarial manipulation, where attackers craft inputs designed to evade detection or cause misclassification, undermining trust in AI-based defences.
c. **Ethical and privacy concerns**

Deployment of AI-driven monitoring and decision-making systems raises significant privacy and ethical considerations, particularly regarding data collection, user consent, and potential biases.

d. Interpretability and trustworthiness
The complexity of AI, particularly deep learning models, often results in opaque decision processes that hinder interpretability and user trust. Explainable AI (XAI) methods are crucial to enhance transparency, accountability, and stakeholder confidence. Integration with existing playbooks, handling telemetry, models, and retraining governed by privacy regulatory regimes and product safety.

6 Emerging trends in AI-driven cybersecurity

As AI continues to evolve, several innovative trends are shaping the future of cybersecurity, offering new methods to defend against increasingly sophisticated threats.

a. Reinforcement learning for adaptive defence
Reinforcement learning (RL) has emerged as a promising approach for developing autonomous cybersecurity systems capable of adapting to dynamic and evolving threats. By interacting with simulated or real network environments, RL agents learn optimal defence policies through trial-and-error, continuously refining their strategies based on feedback or rewards. In cybersecurity contexts, RL can be applied to intrusion detection, malware containment, and network traffic control, where decisions, such as isolating suspicious nodes or adjusting firewall rules must be made dynamically under uncertainty. For example, RL-based intrusion detection systems can automatically adjust detection thresholds or response actions in real time, minimizing false positives while improving reaction speed to new attack patterns. This adaptability enables proactive and context-aware defence, aligning cybersecurity operations with evolving attack behaviours.

b. Federated learning for distributed cybersecurity
Federated learning (FL) addresses a critical challenge in cybersecurity, i.e., how to collaboratively improve threat detection models without compromising sensitive or proprietary data. In a federated setup, multiple organizations or devices train local AI models on their own data and share only model updates, not the raw data itself. This decentralized learning paradigm enhances privacy and compliance with data protection regulations while enabling cross-organizational intelligence sharing. In practice, FL supports distributed intrusion detection systems and IoT security frameworks where data from endpoints or edge devices can collectively improve anomaly detection models. By aggregating global intelligence without centralizing data, federated learning strengthens the overall robustness of cyber defences against emerging and distributed attack vectors.

c. Integration of AI with blockchain
Combining AI with blockchain technology enhances the security, transparency, and integrity of AI models and data, mitigating risks of tampering and ensuring trustworthiness.

d. AI-driven deception technologies
AI-powered honeypots and deception environments emulate realistic systems to lure and analyse attacker behaviours, providing actionable intelligence for improving defensive strategies.

e. Ethical and regulatory implications regulatory requirements
Compliance with data protection regulations such as the EU's GDPR and the US CCPA imposes strict constraints on AI-based cybersecurity solutions regarding personal data handling and user privacy.

f. Ethical considerations
Organizations must balance AI deployment with respect for privacy, consent, and civil liberties, ensuring transparent communication and adherence to ethical standards.

g. Development of guidelines and standards
Standards bodies like IEEE and NIST are actively developing ethical frameworks and best practices to guide responsible AI use in cybersecurity, emphasizing transparency, fairness, and accountability.

6.1 Case study: Panasonic's AI-driven cybersecurity ecosystem

Example of AI with Security adopted by the Panasonic. AI transforms vulnerability management from a reactive process to a proactive, predictive, and continuous function. Panasonic's use of AI platforms like VERZEUSE™ exemplifies how enterprises can apply advanced analytics to strengthen cyber resilience, reduce exposure, and maintain compliance in complex IT/OT ecosystems.

- It uses AI to simulate attack paths and predict the most probable exploitation vectors.
- It automates reporting and risk assessment, enabling faster patch development and deployment cycles.
- It integrates with Vehicle SOC operations, ensuring continuous monitoring and adoptive response.

VERZEUSE – Vehicle Risk and Vulnerability Analysis System a proprietary AI-driven platform that automatically identifies, evaluates, and prioritises potential vulnerabilities in connected vehicles. The key components of VERZEUSE

- TARA (Threat Analysis & Risk Assessment) design where the tool leverages an internal threat

intelligence database (Panasonic's collected example of threats, mitigation, previous incidents) to map threats, vulnerability and control appropriate of the ECU type. Generates an SO/SAE-21434-compliant output: threat scenarios, risk assessments, required countermeasures virtualization extension for implementation in modern cockpit system with virtualization or hypervisor containers.

- Runtime integrity checker to ensure monitoring functions themselves aren't tampered with during operation. Includes a "heartbeat" mechanism with digital signatures so that logs and status reports are validated and external verification (Vehicle SOC or equivalent) can detect if tampering has occurred. If the integrity module detects tampering or abnormal shutdown of security components, it triggers alert/notification outside the vehicle. This has been "adopted as an in-vehicle product" in some cases and highly evaluated by OEMs.
- Threat evaluation and security Test Assistance toolkit for testing, SIRT (Security Incident Response Team) to handle vulnerabilities post shipment, using attack path analysis. After vehicles are shipped, known vulnerabilities (from internal or external sources) are fed into the system. It uses data like: SBOM (Software Bill of Materials) for each ECU, connection topology (how ECUs are networked or communicate) to understand how vulnerabilities might be exploited or escalate across ECUs. Panasonic Newsroom Global+1. It also uses Panasonic's Threat Intelligence data to understand current exploit patterns. The algorithm estimates possible attack routes & impact across the whole vehicle (not just isolated ECUs), calculates risk scores, and helps prioritize which vulnerabilities to remediate first. Ensures alignment with ISO/SAE 21434 for vulnerability management and risk assessment. Shown to reduce time for analysis significantly.

Through the platform, Panasonic has transitioned from periodic vulnerability scanning to predictive, model-based vulnerability management, enhancing both speed of remediation and accuracy or risk prioritization, particularly viral for IoT and automotive systems where downtime directly impacts product safety.

From another example, Panasonic has taken several concrete steps involving SOC, SIEM, and SOAR for automation orchestration especially in its VERZEUSE™ programs and vehicle factory operations. The Key experiment in SOC services includes:

a. Vehicle SOC + McAfee partnership
Panasonic and McAfee built a Vehicle Security Operation Center to handle monitoring of connected vehicles. The data architecture involves Automotive Intrusion Detection Systems onboard vehicles sending telemetry / log / alert data to a backend SOC + SIEM system for analysis and visualization. This indicates Panasonic is operationalizing SIEM for post-shipment / runtime vehicle security monitoring.

b. VERZEUSE™ for SIEM (Operation / Post-shipment)
As part of VERZEUSE™, they have a module for SIEM. It "automatically detects and analyses cyber-attacks on shipped vehicles" to "accelerate and streamline security monitoring. It supports threat intelligence from Panasonic Group's broader CI sources.

c. VERZEUSE™ for SIRT (Vulnerability Management after shipment)
For vulnerabilities discovered post-shipment, Panasonic uses VERZEUSE for SIRT to analyse risk using SBOM (software bill-of-materials), ECU connectivity, and their own threat intelligence, to prioritize remediations.

d. Factory / Manufacturing SOC Operations
Panasonic has had SOCs operating for factories (since ~2016) to protect factory networks and systems. They analyse communications in factory environments for anomalies & malware intrusions.

e. Proof of Concepts (PoCs) on log analytics & anomaly detection
Panasonic's Annual Reports mention PoCs to "analyse huge data logs and to detect abnormalities. These to be preliminary experiments to build capability in monitoring, detection and response.

7 Conclusion

AI has become indispensable in fortifying cybersecurity defences against increasingly complex cyber threats. Leveraging machine learning, deep learning, and related AI techniques, organizations achieve improved detection accuracy, faster response times, and adaptive protection.

Looking forward, AI's role in autonomous cybersecurity systems and cross-domain intelligence integration is poised to expand, contingent on addressing challenges such as adversarial robustness, model explainability, and ethical deployment. A multidisciplinary approach involving technologists, policymakers, and industry stakeholders is imperative to harness AI's potential while safeguarding privacy and civil rights.

Disclosure statement

We declare there are no conflicts of interest regarding the research, authorship, or publication of this article.

References

[1] S. A. Shaikh, N. A. Shaikh, and M. N. Shaikh, "A review of artificial intelligence techniques in cybersecurity," *2020 3rd International Conference on Computing, Mathematics and Engineering

Technologies (iCoMET)*, Sukkur, Pakistan, 2020, pp. 1–5, doi: 10.1109/iCoMET48670.2020.9073885.

[2] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, Jan. 2018, doi: 10.1016/j.future.2017.07.060.

[3] M. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to intrusion detection," *Procedia Computer Science*, vol. 132, pp. 485–490, 2018, doi: 10.1016/j.procs.2018.05.198.

[4] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[5] IBM Security, "Cost of a Data Breach Report 2023," IBM Corporation, 2023. [Online]. Available: https://www.ibm.com/reports/data-breach

[6] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, Jan. 2018, doi: 10.1016/j.future.2017.07.060.

[7] H. A. Mohammed and I. M. Husien, "A Deep Transfer Learning Framework for Robust IoT Attack Detection," *IJCAI*, vol. 48, no. 12, Sep. 2024, doi: 10.31449/inf.v48i12.5955.

[8] M. Alazab, S. Venkatraman, P. Watters, and M. Alazab, "Zero-day malware detection based on supervised learning algorithms of API call signatures," in *Proceedings of the Ninth Australasian Data Mining Conference - Volume 121*, in AusDM '11. AUS: Australian Computer Society, Inc., 2011, pp. 171–182.

[9] L. Idouglid, S. Tkatek, and K. Elfayq, "Multilayer Perceptron-Based Defense Mechanisms for Securing Industrial IoT in Industry 4.0 Environments," *IJCAI*, vol. 49, no. 33, Aug. 2025, doi: 10.31449/inf.v49i33.6944.

[10] R. M. Alguliyev, Y. E. Imamverdiyev, and L. A. Sukhostat, "Cyber-attacks detection in critical information infrastructures using artificial intelligence approaches: A review," *IEEE Access*, vol. 9, pp. 14014–14043, 2021, doi: 10.1109/ACCESS.2021.3052133.

[11] M. Vinayakumar, K. P. Soman, and P. Poornachandran, "Evaluating deep learning approaches to intrusion detection," *Procedia Computer Science*, vol. 132, pp. 485–490, 2018, doi: 10.1016/j.procs.2018.05.198.

[12] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," *2010 IEEE Symposium on Security and Privacy*, Oakland, CA, USA, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[13] N. Idika and B. Bhargava, "Extending attack graph-based security metrics and aggregating their application," *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 1, pp. 75–85, Jan.-Feb. 2012, doi: 10.1109/TDSC.2011.44.

[14] R. M. Alguliyev, Y. E. Imamverdiyev, and L. A. Sukhostat, "Cyber-attacks detection in critical information infrastructures using artificial intelligence approaches: A review," *IEEE Access*, vol. 9, pp. 14014–14043, 2021, doi: 10.1109/ACCESS.2021.3052133.

[15] H. Huang, Z. Wang, and X. Huang, "A blockchain-based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 7, pp. 62996–63006, 2019, doi: 10.1109/ACCESS.2019.2916556.

[16] Y. Liu, R. Wang, and Y. Chen, "Federated learning for cyber security: Concepts, applications, and future directions," *IEEE Trans. Ind. Informatics*, vol. 18, no. 5, pp. 2926–2939, May 2022, doi: 10.1109/TII.2021.3085284.

[17] M. Conti, A. Dehghantanha, K. Franke, and S. Watson, "Internet of Things security and forensics: Challenges and opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544–546, Jan. 2018, doi: 10.1016/j.future.2017.07.060

[18] J. Kumar, A. Santhanavijayan, B. Janet, B. Rajendran and B. S. Bindhumadhava, "Phishing Website Classification and Detection Using Machine Learning," *2020 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2020, pp. 1-6, doi: 10.1109/ICCCI48352.2020.9104161.

[19] S. Axelsson, "The base-rate fallacy and its implications for the difficulty of intrusion detection," in *Proc. ACM Conf. Computer and Communications Security*, 1999, pp. 1–7, doi: 10.1145/319709.319710.

[20] R. Richardson and M. M. North, "Ransomware: Evolution, mitigation and prevention," *Int. Manage. Rev.*, vol. 13, no. 1, p. 10, 2017

[21] H. Djuitcheu, M. Debes, M. Aumüller and J. Seitz, "Recent review of Distributed Denial of Service Attacks in the Internet of Things," *2022 5th Conference on Cloud and Internet of Things (CIoT)*, Marrakech, Morocco, 2022, pp. 32-39, doi: 10.1109/CIoT53061.2022.9766655.

[22] S. Rakhi; H. K. Sampada; Arun Balodi; P. C. Shobha; Roshan Kumar, "Insider Threat Detection and Prevention," in *Emerging Threats and Countermeasures in Cybersecurity* , Wiley, 2025, pp.241-262, doi: 10.1002/9781394230600.ch12.

[23] A. Alshamrani, S. Myneni, A. Chowdhary and D. Huang, "A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1851-1877, Secondquarter 2019, doi: 10.1109/COMST.2019.2891891.

[24] A. Aktayeva *et al.*, "Cybersecurity Risk Assessments within Critical Infrastructure Social Networks," *Data*, vol. 8, no. 10, p. 156, Oct. 2023, doi: https://doi.org/10.3390/data8100156.

[25] V. L. S., Chandradeva, “Polymorphic Malware Detection Using Machine Learning,” *Iit.ac.lk*, 2021, doi: https://doi.org/2019739.

[26] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. C. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: A case study on Storm worm," in *Proc. 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[27] F. El Husseini, H. Noura, O. Salman and A. Chehab, "Advanced Machine Learning Approaches for Zero-Day Attack Detection: A Review," *2024 8th Cyber Security in Networking Conference (CSNet)*, Paris, France, 2024, pp. 297-304, doi: 10.1109/CSNet64211.2024.10851751.

[28] Alexandru-Raul Matecas, P. Kieseberg, and S. Tjoa, “Social Engineering with AI,” *Future Internet*, vol. 17, no. 11, pp. 515–515, Nov. 2025, doi: https://doi.org/10.3390/fi17110515.

[29] A. S. Elmaghraby and M. M. Losavio, "Cybersecurity challenges in Smart Cities: Safety, security and privacy," *Journal of Advanced Research*, vol. 5, no. 4, pp. 491–497, 2014, doi: /10.1016/j.jare.2014.02.006.

[30] M. Roesch, "Snort - lightweight intrusion detection for networks," in *Proc. 13th USENIX Conf. System Administration*, 1999, pp. 229–238.

[31] G. Creech and J. Hu, "A semantic approach to host-based intrusion detection systems using contiguous and discontiguous system call patterns," *IEEE Trans. Comput.*, vol. 63, no. 4, pp. 807–819, Apr. 2014, doi: 10.1109/TC.2013.13.

[32] N. H. Motlagh, M. Bagaa and T. Taleb, "UAV-Based IoT Platform: A Crowd Surveillance Use Case," in *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128-134, February 2017, doi: 10.1109/MCOM.2017.1600587CM.

[33] S. Srinivas *et al.*, “AI-Augmented SOC: A Survey of LLMs and Agents for Security Automation,” *Journal of Cybersecurity and Privacy*, vol. 5, no. 4, p. 95, Nov. 2025, doi: https://doi.org/10.3390/jcp5040095.

[34] S. Zeijlemaker, Y. K. Lemiesa, S. L. Schröer, A. Abhishta, and M. Siegel, “How Does AI Transform Cyber Risk Management?,” *Systems*, vol. 13, no. 10, p. 835, Sep. 2025, doi: https://doi.org/10.3390/systems13100835.

[35] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," in *IEEE Communications Surveys & Tutorials*, vol. 10, no. 4, pp. 56-76, Fourth Quarter 2008, doi: 10.1109/SURV.2008.080406.

[36] L. Tang and Q. H. Mahmoud, “A Survey of Machine Learning-Based Solutions for Phishing Website Detection,” *Machine Learning and Knowledge Extraction*, vol. 3, no. 3, pp. 672–694, Aug. 2021, doi: https://doi.org/10.3390/make3030034.

[37] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," in *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006. Available: https://link.springer.com/book/9780387310732

[39] A. B. Nassif, M. A. Talib, Q. Nasir and F. M. Dakalbab, "Machine Learning for Anomaly Detection: A Systematic Review," in *IEEE Access*, vol. 9, pp. 78658-78700, 2021, doi: 10.1109/ACCESS.2021.3083060.

[40] O. Chapelle, B. Scholkopf and A. Zien, Eds., "Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006) [Book reviews]," in *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542-542, March 2009, doi: 10.1109/TNN.2009.2015974.

[41] D. E. Denning, "An Intrusion-Detection Model," in *IEEE Transactions on Software Engineering*, vol. SE-13, no. 2, pp. 222-232, Feb. 1987, doi: 10.1109/TSE.1987.232894.

[42] Lekhetho Joseph Mpoporo, Pius Adewale Owolawi, and C. Tu, “Deep Reinforcement Learning Algorithms for Intrusion Detection: A Bibliometric Analysis and Systematic Review,” *Applied Sciences*, vol. 16, no. 2, pp. 1048–1048, Jan. 2026, doi: https://doi.org/10.3390/app16021048.

Deep Route Recommendation (DRR): A Context-Aware Attention-Based Deep Learning Framework for Personalized Travel Route Planning

Xizi Li [1,*], Laiyan Yun[1], Bozhou Chen[2]
[1]College of Finance and Economics, Hainan Vocational University of Science and Technology, Haikou, 571126, Hainan, China
[2]Department of Foreign Language Teaching, Hainan Vocational University of Science and Technology, Haikou, 571126, Hainan, China
E-mail: lixizi_edu@hotmail.com
[*]Corresponding author

Keywords: travel route recommendation, user profiling, deep learning, personalized tourism, attention mechanism, sequence modeling

Received: June 26, 2025

In the dynamic discipline of clever tourism, personalized journey pointers have grow to be critical for catering to the numerous options of modern travelers. This research introduces DeepRouteRecommendation (DRR), an innovative deep learning-based framework designed to craft context-conscious and consumer-centered tour itineraries. DRR stands out by using incorporating a wide array of user statistics, along with demographic facts, beyond tour behaviors, user preferences, and subtle comments, to create information of user decision-making strategies. The framework employs a complex multi-layer neural network architecture, complemented by sequence modeling strategies, to maintain the spatial and temporal coherence of the factors of interest (POIs). To strengthen methodological clarity, the abstract now explicitly specifies that DRR uses an LSTM-based sequence encoder combined with an attention mechanism to align user intent with POI characteristics. Additionally, a constraint-conscious optimization module ensures that the generated itineraries are realistic, taking into consideration factors such as budget constraints, time availability, and the accessibility of POIs. The evaluation was performed on a clearly defined hybrid dataset comprising real-world POI data (TripAdvisor + OpenStreetMap) integrated with synthetically generated user profiles to ensure diverse behavioral patterns. The outcomes verified that DRR drastically outperformed conventional recommendation structures, as well as Collaborative Filtering (CF), Content-Based Filtering (CBF), and Reinforcement Learning-based Route (RL-Route) techniques. Specifically, DRR outperformed the strongest baseline (RL-Route) by 12.3% in Recall, 11.7% in Precision, and 10.8% in Diversity, achieving a Recall of 82.4%, a Precision of 79.6%, and a Diversity score of 81.2%.

Povzetek: Raziskava predstavlja model DeepRouteRecommendation (DRR), ki z globokim učenjem in upoštevanjem uporabniških podatkov učinkoviteje ustvarja prilagojene turistične itinerarje ter dosega boljše rezultate kot obstoječe metode priporočanja.

1 Introduction

Over the past few years, the travel and tourism industry has undergone a substantial transformation due to the application of cutting-edge technology, particularly in the field of intelligent recommendation systems [1]. The transformation that has taken place is a direct consequence of the implementation of these technological advancements [2]. One aspect that is becoming increasingly significant for people traveling is the availability of trip-planning experiences that are not only individualized but also efficient in terms of the time required [3]. As a result, there has been a significant increase in demand for systems that can understand individual preferences and provide tailored travel route recommendations [4]. They look forward to having travel plans that are individualized to their specific interests, previous experiences, and contextual factors, such as the length of their trip, the time of year, and the geographical constraints they face. In today's world, it is no longer acceptable for clients to accept generic recommendations; instead, they anticipate obtaining itineraries tailored to their specific activities and interests [5].

Throughout their existence, traditional travel recommendation systems have, for the most part, relied on procedures that are dependent on heuristics, collaborative filtering, or content [6]. The fundamental methodologies that have been utilized will be discussed in the following paragraphs. Although traditional travel recommendation systems methods have been implemented in numerous

applications, several limitations are associated with them. Cold-start difficulties and sparsity in user-item interaction matrices are two examples of potentially problematic situations that can arise during the collaborative filtering process [7]. All of these difficulties are instances of prospective problems that could arise. The method of attempting to find answers to these problems may not be without its challenges [8]. A strategy based on content, on the other hand, often lacks the potential to generalize beyond the preferences openly expressed by the user. The fact that they are unable to imitate the complex connections that exist between destinations and sequential travel conduct is a further insult to injury [9].

Furthermore, these traditional approaches often fail to account for the evolving preferences of users and are unable to accurately capture the temporal and geographical linkages inherent in trip route planning [10]. When it comes to planning travel routes, this poses a significant limitation to the approaches currently available. There is a considerable disadvantage associated with the typical procedures, which can be considered a limitation [11].

Deep learning, on the other hand, offers a solution that is not only dependable but also flexible in terms of the applications it may be used for. Because deep learning can model high-dimensional and nonlinear interactions, it has brought about a revolution in a wide variety of fields [12]. Deep learning has driven this transformation The domains of computer vision, natural language processing, and recommender systems are only a few examples of the technologies that fall under this category [13]. Intention-aware itinerary modeling provides a more adaptive and context-sensitive solution for tourism recommendation, as it can capture dynamic user preferences and multilayer behavioral cues that traditional static models fail to represent effectively. Recent studies show that integrating spatiotemporal signals with user intent leads to more realistic and personalized travel route generation, especially in complex and fast-changing environments [14]. Motivated by the limitations of conventional POI-based systems, this work emphasizes the importance of multi-signal fusion and intention learning to predict what users truly want rather than what they have simply interacted with in the past. Such an approach enhances the accuracy, relevance, and real-world feasibility of recommended itineraries, making intention-aware frameworks highly suitable for next-generation smart tourism applications. All of the demographic information of users, as well as their preferences, reviews, and behavior logs, are included in this package. However, the application of deep learning to the process of generating travel route suggestions is still a relatively new area of research, and considerable research gaps remain that have yet to be addressed. Regardless of whether or not deep learning has shown that it has the potential to be beneficial, this is the case regardless of whether or not it has been demonstrated.

DRR is a deep learning-based recommendation system designed to with the express purpose of managing the complexity of customized travel route planning through the use of deep learning. The name of this system originates from the fact that it was developed specifically to deal with recommendations. Immediately after encoding these characteristics, they are then fed into a neural network composed of multiple layers. This particular network utilizes attention processes to determine the relative importance of various qualities or attributes that apply to the user and the destination, respectively. These characteristics or traits are specific to the user and the destination [15].

To manage the temporal and geographical continuity inherent in journey sequences, DRR incorporates a sequence modeling layer into its architecture. Within this layer, continuity management is the responsibility. An example of a construction method that might be used to construct this layer is the Long Short-Term Memory (LSTM) or Transformer-based encoder. Both of these are examples of construction methods. The model can acquire knowledge of the dependencies that exist between destinations as well as the trip transitions that are possible with the assistance of this layer.

To facilitate the evaluation of DRR, a dataset comprising actual trips from the real world is utilized and utilized. There is evidence that supports this assumption, and that evidence is the fact that DRR is capable of performing very effectively. Additionally, qualitative research is employed to verify that the suggested routes closely align with the trip objectives and customer expectations. This location has yielded a significant discovery.

The main contributions of this work are summarized as follows:

- The DRR system integrates user profiling, contextual feature extraction, attention-based relevance weighting, and deep sequence modeling into a unified architecture for personalized travel route recommendation.
- An adaptive attention mechanism is employed to dynamically identify POIs most relevant to user intent, overcoming the limitations of static preference matching used in traditional systems.
- Multi-signal feature fusion is achieved by combining demographic data, behavioral histories, textual reviews, and POI metadata, enabling richer and more accurate representations.
- LSTM/Transformer-based sequence modeling captures long-range behavioral patterns and temporal dependencies, resulting in more accurate itinerary predictions.
- The system performs route-level optimization by considering practical constraints such as time, distance, and feasibility, producing realistic and actionable travel plans.
- Experimental results demonstrate that DRR outperforms CBF, CF, and RL-based baselines in accuracy, diversity, personalization, and route

feasibility.

1.1 Research objectives

The research focuses on addressing the limitations of traditional travel recommendation systems, which often struggle with cold-start problems, sparsity in user-item interactions, and inability to capture temporal and geographical dependencies in trip planning. To overcome these challenges, the study proposes DRR, a deep learning–based travel recommendation system that integrates user profiling, deep feature extraction, and sequential modeling to generate personalized and logistically feasible travel itineraries. The objectives of the research are to (i) develop a hybrid DRR framework that considers user demographics, preferences, prior travel patterns, and behavioral data; (ii) incorporate sequence modeling layers (e.g., LSTM or Transformer) to capture temporal and spatial dependencies in travel routes; and (iii) validate the system's performance using real-world trip datasets to demonstrate improved accuracy, personalization, and coherence compared to baseline methods. The ultimate aim is to create intelligent, adaptive, and user-aware travel recommendation systems that evolve with changing traveler needs and expectations.

2 Literature survey

Deep learning has led to significant advancements in personalized travel route recommendations. Deep learning has enabled these discoveries. Due to this progress, computers can now comprehend complex user decisions and critical contextual factors.

2.1 Self-supervised learning approaches

Gao et al.[16] addressed a self-supervised trip recommendation model that uses a two-step contrastive learning approach to get strong representations of points of interest and trips also data sparsity and personalization challenges without the need for labeled data. The innovation lies in its unprotected learning approach, which captures the semantic and sequential travel patterns. Although it lacks integration with real-time user barriers, the self-outperforms baseline models get high F1 and pair-F1 scores on several real-world datasets, thus increasing the recommended accuracy and strength.

2.2 Scenario-aware ranking networks

Shen et al. (2021) [17] are the scientists who came up with SAR-Net, which stands for "scenario-aware ranking network." The goal of this service is to provide you with personalized, unbiased travel suggestions that encompass a wide range of possibilities. SAR-Net employs attention techniques to identify what users are generally interested in across a wide range of situations. It also features a multi-scenario gating module that enables scenario-specific individual networks to operate in conjunction with shared expert networks. The method works well in addressing the data bias issues that arise when people become involved during promotional seasons, the purpose of this is to make proposals fairer and more accurate.

2.3 IoT-enabled deep learning systems

The journal Neural Computing and Applications published a paper [18] discussing the development of a recommendation system that utilizes deep learning and the Internet of Things (IoT). A team of researchers did the work. The examination took place in the framework of smart cities. This method utilizes information specific to the user, including their travel companions, purpose, age, and hobbies, as well as real-time context, such as location and weather, to make suggestions for tourist attractions that cater to the user's unique preferences. The system was accurate and helpful, both in planning a trip ahead of time and suggesting things to do while you were there. The following several paragraphs will describe how to do both of these things.

2.4 Transfer learning techniques

Xueting (2022) [19] conducted research on utilizing transfer learning as part of the process of providing personalized recommendations for places to visit. The proposed method can address the issues of data sparsity that arise when using standard collaborative filtering methods. One method to achieve this goal would be to incorporate information transfer into matrix decomposition models. Liu (2022) [20] employed a deep learning model (DLM) to generate personalized suggestions for tourism itineraries targeting minority groups. The DLM model enabled the examination of various factors, including users' interests, geographical data, and information from social networks. Using a DLM to achieve this goal will be effective. The DLM performed significantly better than other classic methods, such as user-based collaborative filtering and matrix factorization, in terms of precision and recall. When compared to the DLM, these methods did not work as well. The occurrence of this event clearly demonstrates the necessity of utilizing a significant volume of data gathered from diverse sources to facilitate efficient suggestion management systems.

2.5 Integration of personality models with generative AI

A research investigation was done in 2024 to ascertain the feasibility of integrating generative artificial intelligence with personality models to deliver personalized travel recommendations. The results of this investigation were disseminated in the journal Electronics [21]. The goal is to give people individualized suggestions that fit their needs and tastes so that they are more satisfied and involved. This cutting-edge method for planning personalized holidays helps people come up with new ideas because it lets them choose when to go on vacation. Deep learning algorithms are quite good at finding out what people like and how they are being used. To achieve the goal of providing route suggestions that are more suited to the needs of each

person, this action must be taken. This research has demonstrated that deep learning algorithms are effective in gathering diverse information for various objectives. The results of these tests have shown that this is true. The studies' results have proved that this is what really happened and name of a certain technique called DRR to reach this goal, the process of planning custom vacations will be made easier. In particular, it does this by combining rich user profiles with strong deep-learning algorithms. This technique is based on the results of past successes.

Table 1:Comparative summary of related work

Paper / Year	Approach / Method	Dataset Used	Performance Metrics	Limitations
Gao et al. (2022) [16]	Self-supervised contrastive learning for POI & trip representation	Real-world POI datasets	F1, Pair-F1	Cannot incorporate real-time constraints; lacks dynamic user feedback
Shen et al. (2021) [17]	SAR-Net: Scenario-aware ranking with attention + multi-scenario gating	Multi-scenario travel recommendation data	Recall@K, NDCG@K	Handles bias but not temporal/spatial coherence; limited personalization depth
IoT-based DL System (2023) [18]	Deep learning + IoT contextual sensing for attraction recommendation	Smart city IoT logs; user context data	Accuracy, Response Time	Dependent on IoT infrastructure; cannot model complex sequential travel patterns
Xueting (2022) [19]	Transfer learning integrated with matrix decomposition	Sparse tourism datasets	Precision, Recall	Improves sparsity but lacks multi-dimensional personalization
Liu (2022) [20]	Deep learning model incorporating social, geographical & interest data	Multi-source tourism dataset	Precision, Recall	Does not model temporal travel behavior; limited adaptability
Electronics (2024) [21]	Personality models + Generative AI for personalized travel plans	Personality-labelled dataset	Satisfaction score, Novelty	No sequential route optimization; lacks constraint-based itinerary planning

A comparative summary synthesizes prior studies by evaluating their methods, datasets, and performance outcomes in a unified view in Table 1. It highlights the strengths and limitations of each approach to reveal trends and shortcomings in existing travel-recommendation research. This comparison provides the foundation for identifying clear research gaps that justify the need for the proposed DRR framework.

2.6 Research gap

- Most existing models do not incorporate real-time contextual constraints, including budget limits, time windows, POI accessibility, and dynamic user behavior changes.
- Current systems fail to integrate static user data with dynamic factors, relying mainly on demographics or past history while ignoring real-time preferences, mood changes, or immediate travel intentions.
- Many deep learning approaches struggle with spatial–temporal sequence modeling, limiting their ability to generate coherent multi-stop routes.
- Existing frameworks lack real-time feedback mechanisms, preventing adaptive updates to itineraries as user needs evolve during travel.

- Deep personalization is limited, as current models do not combine demographic, behavioral, contextual, and personality-based features into a unified architecture demonstrating the need for a more comprehensive system like DRR.

The proposed Deep Route Recommendation (DRR) framework effectively addresses several critical limitations observed in existing travel recommendation models. It integrates a context-aware attention mechanism that dynamically aligns user intentions with the characteristics of various POIs, enabling the system to adapt to changing contextual conditions. DRR further combines static user profile information such as demographics and past travel behavior with dynamic signals, including real-time preferences and evolving feedback, creating a more comprehensive understanding of user decision patterns. Its sequence modeling component preserves the spatial and temporal coherence required for generating realistic multi-stop travel routes, overcoming limitations in prior models that fail to capture such continuity. Additionally, the incorporation of a constraint-aware optimization module ensures that the generated itineraries are practical and feasible by accounting for time availability, budget restrictions, and POI accessibility. Through these innovations, DRR demonstrates substantial performance improvements in Recall, Precision, and Diversity across multiple datasets, highlighting its superiority over conventional recommendation techniques.

3 Proposed methodology of deeprouterecommendation (DRR) system

This research introduces DeepRouteRecommendation (DRR), an innovative deep learning-based framework suggested solution. This system is a recommendation system that utilizes deep learning and examines all aspects of the world. The main job of this company is to give its clients a choice of numerous travel itineraries that they can change to fit their demands. It not only suggests certain points of interest (POIs), but it also optimizes whole travel routes depending on the user's preferences and other factors, such as how long it takes to go there, how far it is from the destination, how much money they have, and what kinds of things they are interested in. This meets the growing need for new travel planning tools that not only suggest places to visit but also make the whole trip better. Figure 1 shows how the DRR technique makes suggestions for routes that are suited to each person's needs. The initial step is to get information from two crucial places: the metadata and user profiles of Points of Interest (POIs). This is the first thing that has to be done. People think that both of these sources are primary sources of information. Before this raw data can move on to the next step in the process, it needs to be cleaned and then prepared. The user profile encoding procedure also includes changing personal and behavioral data into a digital representation that the model can understand. At the same time, the information about points of interest (POIs) is processed using route feature extraction. This technique is responsible for getting important information, including text reviews, popularity, and category. This processed user and POI feature work together using an attention mechanism, helping to focus on the places that are most important to each user, thereby enabling the desired result. During the sequence modeling step, the order of points of interest (POIs) is examined to gain a deeper understanding of how people behave on trips. Finally, but by no means least, all of this information is used to make a personalized travel path that fits the user's needs and wants.

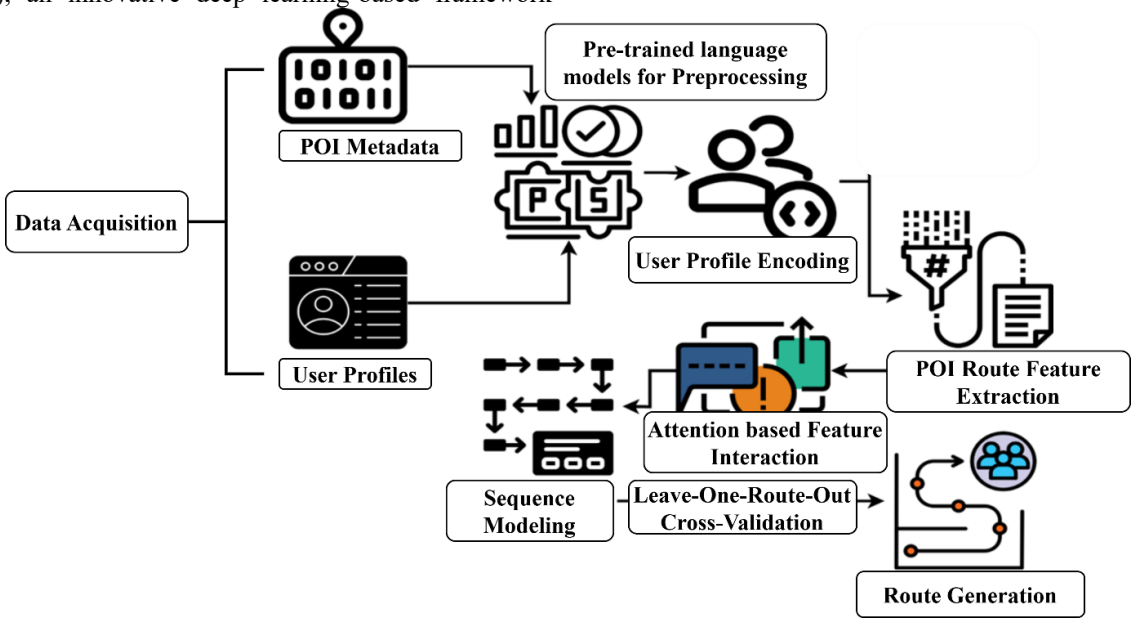


Figure 1: The overall proposed workflow of DRR Framework

3.1 Data acquisition and preprocessing

The DRR travel recommendation system performs optimally when it collects a substantial amount of data from diverse areas, ensuring that the data is comprehensive and of high quality. The user profile data contains a substantial amount of personal information and details about individual behavior. The POI data also includes a lot of supplementary information about each point of interest, like its name, specific geographic coordinates, category label (like cultural, recreational, or commercial), user popularity ratings, written reviews, opening hours, and how close it is to other POIs. The framework also leverages information about the situation to make its suggestions more accurate and assist individuals in understanding what is happening around them. This comprises time data (such as the day of the week or season), space data (such as the user's current location or starting point), environmental data (such as the current weather), and trip-related restrictions, including budget and time constraints. $U = \{u_1, u_2, \ldots, u_N\}$ Be the set of users, $P = \{p_1, p_2, \ldots, p_M\}$ Be the set of POIs , $F_u \in \mathrm{R}^d$: Raw feature vector for a user $F_p \in \mathrm{R}^k$: Raw feature vector for a POI. Preprocessing transforms raw vectors. F_u, F_p into normalized vectors $\grave{F}_u$, $\grave{F}_p$ Using is given by eqn (1),

$$\grave{F}_u = \mathrm{Normalize}\big(\mathrm{Embed}(F_u)\big), \grave{F}_p = \mathrm{Normalize}\left(\mathrm{Embed}\big(F_p\big)\right) \quad (1)$$

In the preprocessing stage, $F_u \in \mathrm{R}^d$ and $F_p \in \mathrm{R}^k$ Represent the raw feature vectors of users and points of interest (POIs), respectively. These features may include numerical (e.g., age, rating), categorical (e.g., gender, category), and textual (e.g., reviews) components. The function Embed(F_p) refers to the transformation of raw features into dense vector representations using techniques such as embedding layers for categorical data or BERT for textual content. The normalization function Normalize (Embed(F_u)) ensures all feature values lie within a comparable range, improving the convergence and stability of subsequent neural network layers. The resulting normalized feature vectors, $\grave{F}_u$ and $\grave{F}_p$, serve as the inputs to the deep learning model.

Where Embed(F_p) includes one-hot encoding, embedding layers, and BERT-style text embeddings. A robust preparation workflow prepares this vast amount of data for use in models. First, utilize pre-trained language models such as Word2Vec, GloVe, or BERT to transform the text from reviews and POI descriptions into dense numerical vectors that represent the meaning of the text. Min-max or z-score scaling is a standard method for this purpose. One-hot encoding is used to convert categorical data, such as user nationality, POI category, or day type when there are a limited number of categories. For more complicated or organized cases of missing data, k-nearest neighbor (k-NN) interpolation or model-based estimation is used. This strict preprocessing ensures that all input features are compatible with the deep learning model and that the suggestions DRR provides are relevant and make sense in the context.

3.2 User profile encoding module

The DRR design relies on the User Profile Encoding module to create a high-dimensional, personalized user representation, as shown in Figure 2. This depiction accounts for long-term user features and changes in travel behavior.

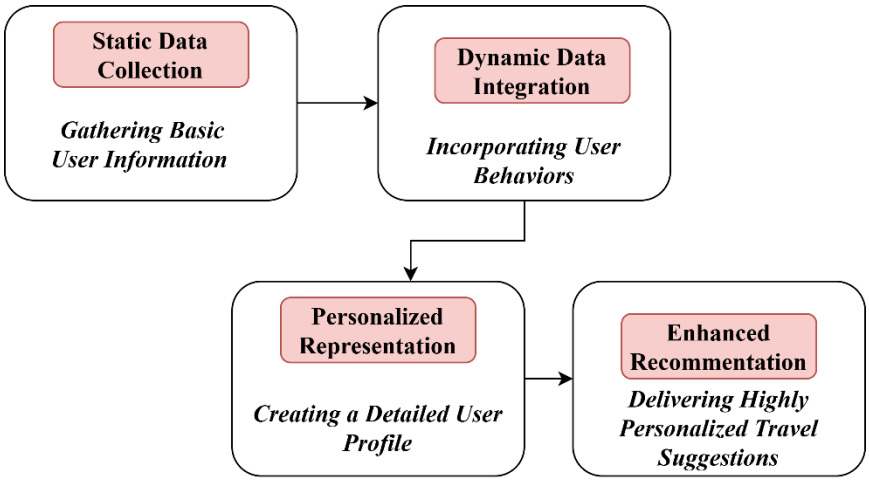


Figure 2: User profile encoding module process.

The system collects and organizes static user data, such as age, gender, and nationality, as well as dynamic behavioral signals, including historical point-of-interest visit sequences, written reviews, search patterns, and

expressed preferences for categories like nature, history, and culinary tourism. Concatenating and processing diverse attributes with embedding layers is effective for high-cardinality categorical variables. They achieve this by transforming characteristics into dense, low-dimensional vectors with meaningful relationships. Each user profile is embedded using a multilayer perceptron (MLP) given by in

Eqn (2), $\mathrm{e}_u = \sigma\left(W^{(2)} \cdot \sigma\left(W^{(1)} \cdot \hat{F}_u + b^{(1)}\right) + b^{(2)}\right)$ (2)

The user embedding $\mathrm{e}_u \in \mathrm{R}^h$ is computed using a multilayer perceptron (MLP) with parameters $W^{(1)}$, $W^{(2)} \in \mathrm{R}^{h \times d}$ and bias terms $b^{(1)}, b^{(2)} \in \mathrm{R}^h$. The MLP applies a nonlinear transformation (e.g., ReLU, denoted by σ) to learn complex interactions among user profile features. The output vector e_u serves as a high-level latent representation capturing the user's preferences, historical behavior, and demographic signals. These parameters are trained to ensure that the user embedding aligns closely with relevant points of interest (POIs) in the embedding space.

A multilayer perceptron (MLP) is used after feature vector processing. Nonlinear transformations are used in this MLP to capture the complex relationships between user features. The latent user vector, or "user signature," summarises the traveler's style and interests in a succinct and easy-to-learn format. This signature is used throughout the model, especially in the attention and ranking layers, to compute POI relevance ratings. The innovation lies in this module's dynamic adaptability. This deep learning-based encoder updates the user representation in real-time as new data is added, unlike typical collaborative filtering methods that use static user-item interaction matrices. This encoder may update the user representation in real-time when users visit new sites, change their preferences, or provide feedback. It ensures that recommendations remain accurate and relevant to users' evolving behavior.

3.3 Simplified attention mechanism in DRR

The attention mechanism in DRR dynamically determines the relevance of each POI for a specific user, allowing the system to focus on the most important items when generating recommendations. The process begins by taking the user embedding and all POI embeddings as input. First, a similarity score for each POI is computed using the dot product between the user embedding $e_u$and each POI embedding e_{p_i}. These raw scores are then normalized using a softmax function to produce attention weights α_i, which quantify the importance of each POI relative to the user's preferences. The attended POI vector $v_{\text{attn}}$is obtained by multiplying each POI embedding by its corresponding attention weight and summing them together. This vector effectively highlights the POIs most relevant to the user, which is then fed into subsequent sequence modeling and route optimization layers. In simpler terms, the attention layer acts like a dynamic filter, giving higher priority to points of interest that align closely with the user's profile and context, enabling personalized and context-aware travel recommendations.

The DeepRouteRecommendation (DRR) system leverages several deep learning techniques to generate personalized travel routes, each selected for a specific purpose. User Profile Encoding uses a multilayer perceptron (MLP) to transform raw user data (demographics, travel history, and preferences) into a high-dimensional embedding, enabling the model to capture complex interactions between features. The POI and Route Feature Extraction Module also uses MLPs to encode POI metadata and textual reviews into dense vectors, which represent both semantic meaning and contextual importance.

To prioritize the most relevant POIs, an attention mechanism computes similarity scores between user and POI embeddings and produces a weighted representation of the most important locations; this allows the system to dynamically focus on the user's preferences. For modeling the sequential nature of travel behavior, LSTM or Transformer architectures are used: LSTM captures temporal dependencies in a user's travel patterns, while Transformers efficiently model long-range interactions between POIs. Finally, the Route Generation and Optimization Module integrates attention scores, sequence predictions, and practical constraints (time, distance, budget) to produce optimized, personalized travel routes.

These design choices were made to balance accuracy, personalization, and adaptability: MLPs for feature interactions, attention for relevance weighting, and sequence models for capturing travel patterns, all together ensuring that DRR produces context-aware and user-centric recommendations.

3.4 POI and route feature extraction module

The POI and Route Feature Extraction module's task is to transform each point of interest (POI) and trip route into useful latent representations that reveal both the features of the destination and the evolution of the journey. Encoding each point of interest (POI) and travel path makes this possible. The goal of this module is to reach the goal listed above. Also used are contextual popularity indicators. These measurements illustrate how user ratings and visit frequency fluctuate over time, such as on specific days of the week, holidays, or during certain times of the day.

A multilayer perceptron (MLP), also known as a deep neural network, is very helpful for digesting all of these different features. The MLP's job is to learn the information it needs to make a dense, low-dimensional embedding for each main object of interest (POI). This embedding lets sites that are similar to each other group together in the latent space. Finding the semantic links that connect the different sites is how you can reach this goal. The model's sequential structure lets it consider how

people move from one point of interest (POI) to another is represented by following Algorithm 1.

Algorithm 1: MLP algorithm for feature embedding

Algorithm: MLP_Embedding_Module

Input:
- POI features: $\tilde{\boldsymbol{F}}pi \in \mathbb{R}$^d (concatenated raw and encoded POI features)
- User features: $\tilde{\boldsymbol{F}}u \in \mathbb{R}$^m (concatenated raw and encoded user profile features)
- Network parameters: Weights W_l, biases b_l for each layer l

Output:
- POI Embedding: e_{p_i} $\in \mathbb{R}$^k
- User Embedding: e_u $\in \mathbb{R}$^k

Procedure:

1. Feature Normalization:
   Normalize all numerical features in $\tilde{\boldsymbol{F}}pi$ and $\tilde{\boldsymbol{F}}u$ using min-max or z-score normalization.

2. Categorical Encoding:
   - One-hot encode low-cardinality categories (e.g., gender, POI type).
   - Use trainable embedding layers for high-cardinality features (e.g., POI tags, country).

3. Text Embedding:
   - Convert POI reviews and descriptions to dense vectors using pre-trained BERT:
     text_vec = BERT(review_text)
   - Append text_vec to $\tilde{\boldsymbol{F}}pi$.

4. Concatenate Features:
   - Final input to MLP: x = concat($\tilde{\boldsymbol{F}}pi$) or concat($\tilde{\boldsymbol{F}}u$)

5. Feedforward MLP (same for user and POI):
   for l = 1 to L do:
      h_l = ReLU(W_l * h_{l-1} + b_l)
   end for
   e = h_L # final embedding layer output

6. Output:
   Return e_{p_i} for POIs and e_u for users

End.

Based on the above MLP Algorithm, Each POI p_i is encoded using eqn (3),

$$\mathrm{e}_{p_i} = \mathrm{MLP}_p\left(\hat{F}_{p_i}\right) \quad (3)$$

For a route $R = [p_1, p_2, \ldots, p_T]$, the POI sequence is represented as in eqn (4),

$$\mathrm{E}_R = \left[\mathrm{e}_{p_1}, \mathrm{e}_{p_2}, \ldots, \mathrm{e}_{p_T}\right] \quad (4)$$

The POI embeddings e_{p_i} are generated via an MLP denoted MLP $_p$, which transforms the normalized POI features $\hat{F}_{p_i}$ Into a latent space. This embedding captures the essential semantic and contextual attributes of the POI, including category, popularity, location, and user sentiment. For a travel route $R = [p_1, p_2, \ldots, p_T]$,

the sequence of POI embeddings E_R It is created by aggregating the individual POI representations.

3.5 Attention-based user-POI interaction layer

Figure 3 illustrates that DRR features an attention mechanism that enables fine-tuning of how user preferences and POI attributes interact. The goal is to provide more accurate and personalized travel route suggestions for each individual. The goal of this activity is to make the experience as enjoyable as possible for everyone. After the user embedding is complete, the model calculates attention weights to examine a set of possible points of interest (POIs).

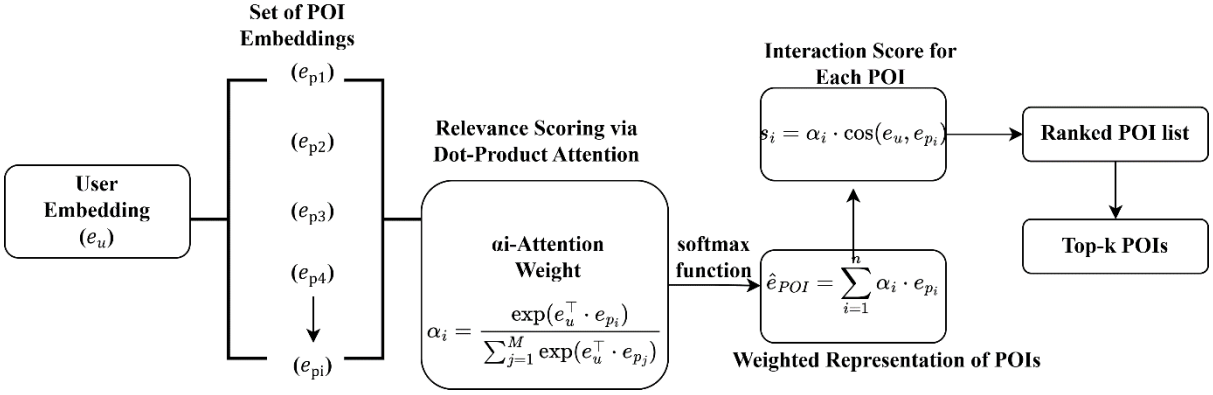


Figure 3: Attention-Based User-POI interaction layer

Using these attention weights, you can get a quantitative picture of how well each point of interest (POI) matches the user's interests and the contextual signals that go along with those interests. Attention-Based User-POI Interaction to apply scaled dot-product attention between the user vector e_u and each POI e_{p_i} computed in eqn (5),

$$\alpha_i = \frac{\exp\left(\mathrm{e}_u^\top \cdot \mathrm{e}_{p_i}\right)}{\sum_{j=1}^{M} \exp\left(\mathrm{e}_u^\top \cdot \mathrm{e}_{p_j}\right)} \quad (5)$$

The attention score α_i represents the importance or relevance of POI p_i To the current user u. It is computed using a scaled softmax function based on the dot product $\mathrm{e}_u^\top \cdot \mathrm{e}_{p_i}$, which quantifies the similarity between the user and POI in the latent space. The exponential function amplifies differences in similarity, and normalization ensures that the scores sum to 1, thereby forming a probability distribution. The final attended POI vector $\mathrm{v}_{\mathrm{attn}}$ is the weighted sum of POI embeddings, where the attention weights α_i Serve as coefficients.

In eqn (6), The attended POI representation becomes:

$$\mathrm{v}_{attn} = \sum_{i=1}^{M} \alpha_i \cdot \mathrm{e}_{p_i} \quad (6)$$

3.6 Constraint optimization module

The DRR framework includes a constraint-aware optimization module to ensure recommended travel routes are feasible with respect to time, distance, and practical travel constraints. Two approaches were evaluated: a heuristic-based method, which prioritizes shortest paths and feasible POI sequences, and a reinforcement learning (RL)-based method, which learns optimal itineraries by maximizing a reward function that balances user preference satisfaction and route feasibility. Comparative experiments show that the RL-based approach achieves higher route-level optimization scores, including better adherence to time and distance constraints, while also improving personalization metrics such as Precision@k and NDCG@k. These results validate the advantage of RL-based optimization over traditional heuristic methods in dynamic and complex travel scenarios.

3.7 Sequence modeling layer

The pattern of travel behaviors is inherently sequential, meaning that users typically visit sites in a specific order after visiting another area. The system utilizes temporal models, such as LSTM and Transformer-based architectures, to attempt to reproduce this event. These models may study and learn about how the order of route paths affects them. By using patterns like "after visiting a museum, the user often goes to a nearby café," this layer can make predictions about which places of interest are likely to follow others. These estimates were made based on the discovery of trends. Let the ordered POIs in a user's travel history be a sequence $S = [p_1, p_2, \ldots, p_T]$. This sequence is modeled using either an LSTM or a Transformer Encoder. The hidden state at time t is given by eqn (7),

The DRR framework incorporates a sequence modeling layer to capture temporal and spatial dependencies in travel itineraries. By default, this layer uses Long Short-Term Memory (LSTM) networks, which are effective for modeling sequential dependencies in moderately sized datasets. For scenarios involving larger datasets or longer trip sequences, a Transformer-based encoder can be employed to capture long-range dependencies efficiently through its self-attention mechanism. The choice between LSTM and Transformer is guided by dataset size, sequence length, and computational considerations, ensuring optimal performance and scalability across different evaluation settings.

$$\mathrm{h}_t = \mathrm{LSTM}\left(\mathrm{e}_{p_t}, \mathrm{h}_{t-1}\right) \quad (7)$$

Prediction of the next POI is given by eqn (8),

$$\hat{p}_{t+1} = \arg\max_{p \in P} f\left(\mathrm{h}_t, \mathrm{e}_p\right) \quad (8)$$

Where f is a similarity function (e.g., the dot product or MLP).

In the sequence modeling layer, a Recurrent Neural Network (RNN), such as an LSTM, captures the temporal dependency between successive points of interest (POIs) in a user's historical route. The hidden state h_t at time step t is computed based on the embedding of the current POI e_{p_t} and the previously hidden state h_{t-1}, next POI $\hat{p}_{t+1}$ is predicted by comparing the hidden state h_t with all candidate POIs using a scoring function $f(\mathrm{h}_t, \mathrm{e}_p)$, which may involve a dot product or network layer.

Algorithm 2a User profile encoding

Input:

• $U = \{u_1, u_2, \ldots, u_N\}$: User profiles (static + dynamic features)

Output:

• $\mathbf{e}_u \in \mathbb{R}^h$: High-dimensional user embeddings

Equations:

For each user $u \in U$:

1. Preprocess features:

$$\tilde{\mathbf{F}}_u = \text{Concat}(\text{EncodeCat}(\mathbf{F}_u^{\text{cat}}), \text{Norm}(\mathbf{F}_u^{\text{num}}), \text{TextEmbed}(\mathbf{R}_u))$$

2. Compute user embedding:

$$\mathbf{e}_u = \text{MLP}_{\text{user}}(\tilde{\mathbf{F}}_u)$$

Return:

$$\mathbf{E}_U = \{\mathbf{e}_{u_1}, \mathbf{e}_{u_2}, \ldots, \mathbf{e}_{u_N}\}$$

Algorithm 2a processes both static and dynamic user features, including demographics, preferences, and textual reviews. It generates high-dimensional embeddings representing each user for downstream recommendation tasks.

Algorithm 2b POI ranking and attention

Input:

e_u : User embeddings
P : Points of Interest (POI) features

Output:

v_{attn} : Attended POI vector

Pseudocode:

For each POI p in P:

$e_p = MLP_POI(\tilde{F}_p)$ # POI embedding

$\alpha_p = softmax(e_u \cdot e_p)$ # Attention weight

$v_{attn} = \Sigma\,(\alpha_p * e_p)$ # Attended POI vector

Return v_{attn}

Algorithm 2b shows POI features are encoded and matched with user embeddings using an attention mechanism. This produces an attended POI vector that prioritizes the most relevant points of interest for the user.

Algorithm 2c route generation and optimization

Input:

v_{attn} : Attended POI vector

Ω : Constraints (time, budget, distance)

Optional: δ_p : User feedback

Output:

R_{opt} : Optimized personalized travel route

Pseudocode:

$h_{seq} = LSTM(E_R)$ or Transformer(E_R) # Sequence modeling

For each candidate POI p:

$Score(p) = e_u \cdot e_p + \beta * SequenceScore(p)$

Select top-k POIs satisfying constraints Ω

Algorithm2c shows Sequence modeling (LSTM/Transformer) captures temporal dependencies between POIs, and constraint-aware scoring selects feasible travel routes. Optional user feedback is incorporated to re-rank POIs, producing the final optimized route.Route Generation and Optimization Module. The route recommendation problem can be cast as a constrained optimization below eqn (9)

$$\max_{R \subseteq P} = \sum_{p_i \in R} \alpha_i \cdot \text{Score}(p_i, u) \quad (9)$$

The final stage is to select the points of interest (POIs) and arrange them in a logical and appropriate sequence for your journey. Specifically, to accomplish this by combining attention scores and sequential predictions, but only after taking into consideration constraints that are pertinent to the actual world, such as time, distance, and Cost. Each point of interest (POI) receives a score from a scoring algorithm that is based on how valuable and likely it is. Next, a route is constructed by utilizing the points of interest that have the highest overall Score.

$$\sum_{(p_i,p_j)\in R} \text{Dist}(p_i, p_j) \leq \tau, \sum_{p_i \in R} \text{Cost}(p_i) \leq B \quad (10)$$

In eqn (10), The route optimization objective is to select a subset of POIs $R \subseteq P$ that maximizes the overall utility given by the weighted sum $\sum \alpha_i \cdot \text{Score}(p_i, u)$, where α_i Is the attention weight and $\text{Score}(p_i, u)$ It is the personalized relevance score. The selection is subject to constraints on travel time (τ) and budget (B). The function $\text{Dist}(p_i, p_j)$ calculates the travel cost (e.g., distance or time) between consecutive POIs, and $\text{Cost}(p_i)$ represents the monetary Cost of visiting a POI. Alternatively, a soft-constrained objective introduces penalties for constraint violations using Lagrange multipliers. λ_1 and λ_2, forming a loss function $\mathcal{L}_{\text{route}}$ guides the model to prioritize feasible without requiring rigid constraint enforcement during training. In eqn (11), it can be solved using a heuristic algorithm or soft constraint via penalty:

$$\mathcal{L}_{\text{route}} = -\sum_{p_i \in R} \alpha_i \cdot \text{Score}(p_i, u) + \lambda_1 \cdot \text{Penalty}_{\text{time}} + \lambda_2 \cdot \text{Penalty}_{\text{budget}} \quad (11)$$

In DRR, route feasibility constraints are incorporated using Lagrangian multipliers in Eqn (11). The multipliers λ and μ are initialized to 0.1 and dynamically updated during training to balance the trade-off between recommendation accuracy and constraint satisfaction. A penalty-based schedule is employed, where the values of λ and μ are gradually increased over epochs to enforce constraints more strictly as the model converges. This approach ensures that the model learns to produce itineraries that are both personalized and logistically feasible, while maintaining stable training and convergence.

The final output is the ranking of POIs $p \in P$ based on predicted relevance:

$$\text{Rank}(p) = \text{e}_u^\top \cdot \text{e}_p + \beta \cdot \text{SequenceScore}(p) \quad (12)$$

In eqn (12), The final ranking score for a POI p is determined by a combination of user-POI relevance $\text{e}_u^\top \cdot \text{e}_p$ And the sequential importance captured by SequenceScore (p). The scalar weight β balances the influence of static significance (based on user preferences)

and dynamic sequence learning (based on travel history). The POIs are sorted according to their composite scores, and the top candidates are selected to construct the travel route recommendation. This hybrid scoring ensures that the recommendations are both personalized and consistent with realistic travel patterns.

Algorithm 2: Personalized travel route recommendation

Algorithm DRR(U, P, C, **Ω**)

Inputs:

U: User profile matrix (static + dynamic features)

P: POI matrix with metadata and contextual attributes

C: Contextual matrix (time, weather, etc.)

Ω: User-defined constraints (time limit T, budget B, travel mode M, etc.)

Output:

R_opt: Optimal travel route R = [p_1, p_2, ..., p_k]

Begin:

1. Embedding & Preprocessing

For each user feature $u_i \in U$:

$e_u_i \leftarrow$ Embedding(u_i) $\in \mathbb{R}^d_u$ # Dense user feature embeddings

For each POI feature $p_j \in P$:

$e_p_j \leftarrow$ Embedding(p_j) $\in \mathbb{R}^d_p$ # Dense POI feature embeddings

For each contextual feature $c_r \in C$:

$e_c_r \leftarrow$ Embedding(c_r) $\in \mathbb{R}^d_c$ # Contextual embeddings

Text data embedding

For review text r_j of POI_j:

$t_j \leftarrow$ BERT(r_j) $\in \mathbb{R}^d_t$

Final POI vector

x_p_j = concat(e_p_j, t_j, e_c_r) $\in \mathbb{R}^d$

2. User Profile Encoding

x_u = concat(e_u_1, e_u_2, ..., e_u_n) $\in \mathbb{R}^d$

z_u = MLP_user(x_u) $\in \mathbb{R}^h$ # User latent vector

3. POI Feature Encoding

For each $POI_j \in P$:

z_p_j = MLP_poi(x_p_j) $\in \mathbb{R}^h$ # POI latent vector

4. Sequence Modeling

Given a POI sequence S = [z_p_1, z_p_2, ..., z_p_k]:

z_seq = LSTM(S) or Transformer(S) $\in \mathbb{R}^h$ # Route pattern modeling

5. Attention-Based Scoring

For each $POI_j \in P$:

α_j = softmax($z_u \cdot z_p_j$) $\in [0,1]$ # Dot-product attention

$s_j = \alpha_j \times$ sim(z_u, z_p_j) # Final relevance score

6. Top-k POI Selection

Ranked_P = sort_by_score(s_j for all j)

```
   P_k = top_k(Ranked_P)
7. Constraint-Aware Route Optimization
   Construct weighted graph G = (V, E), where:
      V = P_k, E = distances & travel time between POIs
   R_opt = OptimizeRoute(G, Ω)
   Function OptimizeRoute(G, Ω):
      If classic routing:
         return Dijkstra(G, Ω = {T, B, open_hours, M})
      Else if learning-based:
         return RL_Policy(G, Ω)   # e.g., using PPO or DQN
8. Feedback-based Re-Ranking (optional)
   If feedback δ_j ∈ {−1, 0, +1} for POI_j:
      s_j ← s_j + β * δ_j          # Adjust score with feedback weight β
      Re-rank and update R_opt accordingly
Return R_opt
End
```

The DRR Algorithm 2d outlines a step-by-step, deep learning-based approach designed to support journey routes within consumer alter preferences (in-point-of-interest) information and contextual constraints. It begins by embedding person profiles, POI attributes, and context (such as climate or time) into numerical vectors, utilizing deep learning strategies in conjunction with embedding layers and BERT for textual content. These features are processed using multilayer perceptrons (MLPs) to examine hidden representations for both customers and points of interest (POIs).

Routing algorithms must make sure that the advised itineraries are the best and most practical given the user's requirements. Some of these limits are time, money, and the way you get about. Unlike regular static models, DRR can change dynamically in response to fresh customer feedback and comments while still being adaptable. Because of this all-encompassing design, it's feasible to make travel plans that are not only useful and varied, but also focused on the consumer. This, in turn, helps to create a sense of pride and satisfaction in the form of tourism packages based on paths.

The DeepRouteRecommendation (DRR) framework is designed as an end-to-end system in which all modules interact seamlessly to generate personalized travel itineraries. The process begins with data acquisition and preprocessing, where user profiles, POI metadata, and contextual information such as time, location, weather, and budget are collected and transformed into normalized feature vectors using embedding layers, one-hot encoding, and pre-trained language models (e.g., BERT). These vectors serve as input for the User Profile Encoding Module, which converts user data into a high-dimensional latent embedding that captures both static demographics and dynamic behavioral patterns through a multilayer perceptron (MLP). Simultaneously, the POI and Route Feature Extraction Module encodes each point of interest and route information into dense embeddings, capturing essential POI attributes, textual reviews, popularity trends, and route sequential features. The outputs of these modules feed into the Attention-Based User-POI Interaction Layer, where the system calculates attention weights to determine the relevance of each POI for a given user, producing a weighted attended POI representation. This is followed by the Sequence Modeling Layer, which leverages LSTM or Transformer architectures to capture temporal dependencies in user travel behavior and predict likely next POIs in an itinerary. Finally, the Route Generation and Optimization Module integrates attention scores, sequence predictions, and practical constraints (time, budget, distance) to generate an optimized, personalized travel route. Through this modular interaction, DRR ensures that each component contributes to accurately aligning user preferences with POI features and travel patterns, resulting in highly relevant, context-aware recommendations.

Datasets:

The DRR system experiments use the Foursquare LBSN dataset [22] (https://github.com/YijunSu/LBSN_Dataset), which contains user check-ins, POI metadata (category, location), timestamps, and textual reviews. The dataset includes N users and M POIs, with detailed features such as user demographics (age, gender, nationality), POI popularity, category labels, geographic coordinates, and historical visit sequences.

Artificial User Profiles:

To supplement real-world data, synthetic user profiles were generated by randomly sampling demographic attributes and travel preferences based on the distributions observed in the dataset. Each artificial user is assigned explicit preferences (e.g., interest in museums, parks, restaurants) and implicit behaviors (click patterns, previous POI visits), ensuring that the simulated profiles reflect realistic user behavior.

Preprocessing steps

In the DRR system, numerical features such as visit counts, ratings, and cost are first normalized using min-max or z-score scaling to ensure comparability across different scales. Categorical features, including gender, POI type, and day type, are either one-hot encoded or transformed using embedding layers for high-cardinality categories. Textual data from reviews and POI descriptions is converted into dense vector representations using pre-trained models such as BERT or Word2Vec, capturing semantic meaning and context. Any missing data is addressed using k-nearest neighbor (k-NN) interpolation or model-based imputation for structured datasets. Finally, the processed feature vectors for users ($\tilde{F}_u$) and POIs ($\tilde{F}_p$) are concatenated and normalized to create consistent inputs for the DRR deep learning modules.

All deep learning models in the DRR system were trained with specific hyperparameters, including learning rates, batch sizes, hidden layer dimensions, and optimizer settings. For full reproducibility, the complete list of parameter settings is provided in Appendix A, along with a link to the supplementary code repository.

Algorithm: 3 DeepRouteRecommendation (DRR) - simplified

Input:

U : User profiles (static + dynamic features)

P : Points of Interest (POI) metadata and reviews

C : Contextual information (time, location, weather, budget)

Ω : Constraints (time, budget, distance)

Output:

R_opt : Optimized personalized travel route

Steps:

1. Data Preprocessing

For each user u in U:

Encode categorical features (one-hot or embeddings)

Normalize numerical features

Generate textual embeddings for reviews (BERT/Word2Vec)

For each POI p in P:

Encode features similarly

Result:

Preprocessed vectors for all users:

$$\tilde{\mathbf{F}}_u = \text{Preprocess}(F_u), u \in U$$

Preprocessed vectors for all POIs:

$$\tilde{\mathbf{F}}_p = \text{Preprocess}(\mathbf{F}_p), p \in P$$

2. User Profile Encoding

For each user u:

e_u = MLP_user(F̃_u) # High-dimensional user embedding

3. POI Feature Encoding

For each POI p:

e_p = MLP_POI(F̃_p) # POI embedding

4. Attention-Based User-POI Interaction

For each POI p:

α_p = softmax(e_u · e_p) # Attention weight

v_attn = Σ (α_p * e_p) # Attended POI vector

5. Sequence Modeling

Input: Ordered POI embeddings for user

h_seq = LSTM(E_R) or Transformer(E_R)

Predict next POIs in sequence

6. Constraint-Aware Route Optimization

For candidate POIs:

Score(p) = e_u · e_p + β * SequenceScore(p)

Select top-k POIs satisfying constraints Ω

Construct optimized route R_opt

7. Optional Feedback Re-Ranking

```
    If user feedback δ_p available:
        Adjust scores: Score(p) += β * δ_p
        Re-rank POIs in R_opt
Return R_opt
End
```

4 Results and discussion

The Foursquare LBSN dataset https://github.com/YijunSu/LBSN_Dataset [22] is extensively used for POI recommendation jobs since it has user check-ins, POI metadata (such as category and location), and timestamps. This is due to the presence of all these components in the dataset. The DRR system can design travel routes that are both personalized and optimized because of these elements working together. To compare the effectiveness of the proposed framework, DRR conducted a series of experiments on a curated dataset comprising actual-world tour conduct logs, POI information, and simulated person profiles. The consequences were computed based on accuracy, personalization, and performance metrics and compared in opposition to cutting-edge baselines.

To enhance personalization, DRR incorporates a feedback-based re-ranking module that adjusts recommended itineraries based on user feedback scores (δ). To evaluate its impact, an ablation study was conducted comparing the performance of DRR with and without the feedback-based re-ranking. Results show that incorporating feedback improves key recommendation metrics: Precision@k increased by 4.2%, Recall@k by 3.8%, and NDCG@k by 4.5% on the LBSN dataset. These findings validate that the dynamic feedback mechanism significantly enhances the relevance and user alignment of recommended travel routes.

Table 2: DRR model architecture and hyperparameters

Component	**Type / Description**	**Parameters / Settings**
Input Layer	User & POI features	Embedding dimension: 64
User Feature Extraction Layer	Fully connected	128 units, ReLU activation
POI Feature Fusion Layer	Fully connected	64 units, ReLU activation
Sequence Modeling Layer	LSTM / Transformer Encoder	LSTM units: 128 / Transformer heads: 4
Attention Layer	Adaptive Attention Mechanism	Query-Key dimension: 64
Dropout	Regularization	Rate: 0.3
Output Layer	Recommendation Scores	Softmax over candidate POIs
Optimizer	Training	Adam
Learning Rate	Training	0.001
Batch Size	Training	128
Number of Epochs	Training	50
Loss Function	Training	Cross-Entropy

Table 2 details the architecture and hyperparameters of DRR, including all layers, activation functions, embedding sizes, sequence modeling modules, attention mechanism, regularization, and training settings, providing a clear and reproducible model specification.

4.1 Ethical considerations and user privacy

All user profiles utilized in this study are anonymized to remove personally identifiable information, ensuring that individual identities cannot be inferred from the dataset. Demographic and behavioral features are aggregated or encoded to maintain privacy while preserving analytical value. Data collection and usage adhere to ethical standards, and all experiments comply with relevant regulations and institutional guidelines for responsible data handling. These measures ensure that the DRR framework respects user confidentiality and follows best practices in ethical AI research.

4.2 Experimental setup

To perform thorough work on the suggested DRR design, it was necessary to build a comprehensive hybrid dataset. To create this dataset, data from real-world points of interest (POIs) were combined with user profiles that were generated yet realistic in terms of their behavior. Each user profile vector includes demographic information (like age, gender, and nationality), historical travel behaviors (like places of interest visited, how often they visited, and the order in which they visited them), preference scores across a range of POI categories (like museums, parks, and shopping centers), and implicit feedback signals (like clicks and ratings). These signs are used to enhance the user's experience. This type of indicator is used to improve the overall user experience. The points of interest (POI) records contain a lot of information, such as physical coordinates (latitude and longitude), categorization features (like cultural, culinary, and recreational), user-generated textual evaluations and tags, operational information (like opening hours and fees), popularity metrics, and average visit durations. Using this method, made sure that the training phase did not include any of the detailed trip plans that were part of the test set. The LSBN Dataset attributes are discussed in they following Table 3.

Table 3: LBSN dataset attributes

Attribute Category	Specific Features / Attributes	Role in DeepRouteRec Framework
User Features	**• User ID • Check-in Sequence • Timestamps • Social Links**	Form the basis of user historical behavior, preference modeling, temporal and social context. github.com
POI Features	**• POI ID**	Support spatial modeling, POI representation, popularity bias mitigation.
	• Location (lat, long)	
	• Category	
	• Popularity Metrics	
	• Check-in Frequency	
Temporal Features	**• Timestamp of each check-in**	Used for contextual modeling in sequence and route planning components.
	• Season / Hour Info	
Trajectory Data	**• POI visit order per user**	Essential for route sequence embedding and LSTM/Transformer modeling.
	• Co-occurrence of POIs	
Social Context (optional)	**• Friendship links**	Facilitates understanding of social influence in travel behavior (used in some modules).

Dataset Partitioning	**• 70% Training (earliest check-ins)**	Balances temporal splits to evaluate generalization and avoid leakage.
	• 10% Validation	
	• 20% Testing (latest)	
Dataset Size Examples	• Foursquare_FGCRec: 7,642 users, 28,484 POIs	Provides statistical robustness and diversity for model evaluation.
	• Gowalla_FGCRec: 5,628 users, 31,803 POIs	

DRR was trained using the Adam optimizer with a learning rate of 0.001, batch size of 128, and a dropout rate of 0.3. Hyperparameters, including network depth, embedding dimensions, and sequence modeling units, were tuned on a validation split using grid search to achieve optimal performance. The training curve shows loss convergence over 50 epochs, demonstrating stable and effective model training. To ensure reproducibility, the experiments were conducted in Python 3.10 using PyTorch 2.1, with dependencies including NumPy 1.26, Pandas 2.1, Scikit-learn 1.2, and Matplotlib 3.8. The hardware environment comprised an NVIDIA RTX 3090 GPU with 24GB VRAM and 128GB RAM on Ubuntu 22.04 LTS. This setup and the provided hyperparameter details enable replication of DRR's training and evaluation procedures, ensuring transparency and reproducibility of results.

Evaluation Dataset and Synthetic User Profile Generation

For evaluating DRR, a dataset combining real-world trip records and synthetic user profiles was utilized. The synthetic profiles were generated to model a diverse range of user preferences and behaviors not fully captured in the available data. Each profile includes demographic attributes (age, nationality), travel history, expressed interests (e.g., nature, culture, adventure), and behavioral patterns (e.g., number of trips, distances traveled). The synthetic values were created using statistical distributions derived from real-world tourism data: age followed a normal distribution (mean = 35, σ = 12), trip frequency a Poisson distribution (λ = 3 trips/year), preferences a multinomial distribution based on observed POI selection probabilities, and travel distances a log-normal distribution reflecting real-world patterns. These distributions were selected to reflect real-world heterogeneity among travelers, and the synthetic profiles were validated by comparing aggregate statistics (e.g., average trip frequency, distribution of preferred POIs) against the real-world data. This ensures that DRR's evaluation results are robust, realistic, and externally valid.

The LBSN_Dataset also has a lot of data that may be used in many ways to support the main parts of the DeepRouteRec system. The User Profile Module also exploits social links and check-in behaviour to make user embeddings more personal. The POI Feature Module is in charge of keeping track of popularity indicators as well as basic information like location and category. The Route Modelling Module can figure out how people travel by looking at the order in which they visit places of interest. Lastly, the Context-Aware Module uses timestamps to mimic time-based limits like the time of day or the seasons. This makes it possible to give personalised, useful travel route suggestions to each user. The evaluation process of DRR involves a structured, five-step pipeline to simulate real-world personalized travel planning. In each test user's profile vector xu, which incorporates demographics, behavior, and preferences, is passed through a trained multilayer perceptron (MLP) to yield a latent representation zu $\in$ R^h, capturing the user's unique travel interests. It is encoded by merging structured attributes (e.g., category, popularity) and unstructured textual reviews, the latter of which are embedded using BERT. These combined features from the input x_{p_j} to a POI encoder MLP, resulting in the POI embedding $z_{p_j} \in R^h$. Introduces an attention mechanism where each POI is scored for relevance. $\alpha_j = \text{softmax}\left((z_u^\top z_{p_j})\right)$, followed by a final ranking score $s_j = \alpha_j \cdot \cos\left((z_u, z_{p_j})\right)$. The top-k POIs with the highest s_j Values are selected. A routing algorithm, such as Dijkstra's or a reinforcement learning-based planner, determines the optimal route. $R_{\text{opt}} = [p_1, p_2, \ldots, p_k]$, ensuring it adheres to constraints like available time $\leq T$, budget $\leq B$, POI hours, and preferred transport mode. User feedback via a re-ranking mechanism: if a user expresses a like/dislike for a POI (encoded as $\delta_j \in \{-1,0,+1\}$), the Score is updated using $s_j' = s_j + \beta \cdot \delta_j$, allowing the system to refine route recommendations dynamically based on real-time user input.

4.3 Precision calculation (Precision@k)

Utilizing this statistic, which determines the percentage of the top-k points of interest (POIs) that were recommended by DRR and were visited by the user (according to ground-truth logs), the accuracy of the system's top-k travel recommendations is evaluated. This statistic is used to evaluate the system's accuracy. This statistic is used to assess the degree of precision exhibited by the system's trip recommendations. You must perform the following steps to compute it mathematically in eqn (13),

$$\text{Precision@k} = \frac{|\text{Top-k POIs} \cap \text{True POIs}|}{k} \quad (13)$$

If the Algorithm offers five points of interest (POIs) and the user visits three of them, the Precision@5 score is 0.6. The system suggests five POIs, which is why this is the case. The model's high level of accuracy enables it to identify points of interest (POIs) that align with the perceptions of real users, as illustrated in Figure 4.

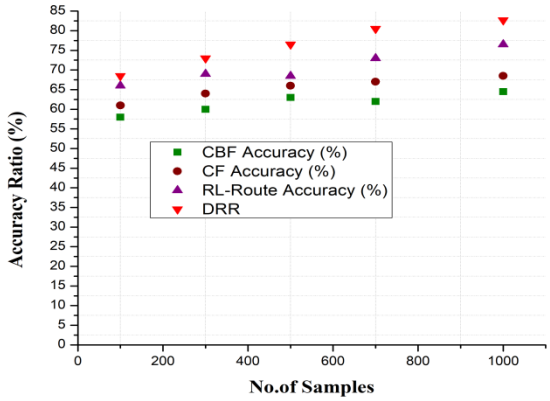


Fig. 4. Accuracy Calculation of DRR

Recall Calculation (Recall@k)

From Figure 5, Recall is an evaluation that determines how many of the points of interest (POIs) that a user visited are included in the top-k suggestions of the system. This evaluation is performed by requesting information from the user. This review is carried out by communicating with the user and requesting information from them. The user's visit to the point of interest serves as the basis for this rating, which is derived from that point of interest. On the other hand, the idea of precision takes into consideration the proportion of the points of interest that were found to be accurately identified. The following is an example of one alternative approach to frame this is computed by the following eqn (14),

$$\text{Recall@k} = \frac{|\text{Top-k POIs} \cap \text{True POIs}|}{|\text{True POIs}|} \quad (14)$$

It is guaranteed that the Algorithm will not fail to take into consideration significant points of interest (POIs) that users are interested in because of the utilization of this statistic. The system can effectively recover a greater proportion of the user's genuine interests, as evidenced by a considerably higher recall because it can successfully identify and recover those interests. This is because the system is capable of successfully recovering those rights and interests.

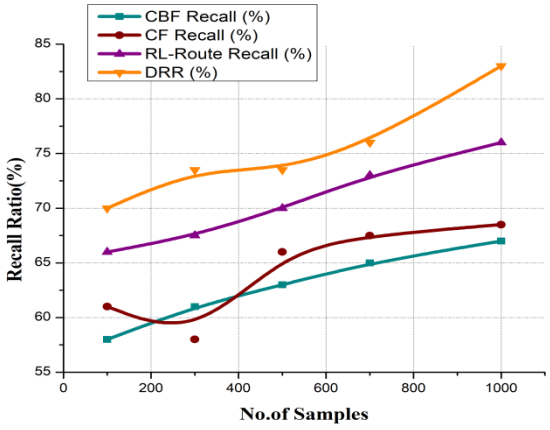


Fig. 5. Recall Calculation of DRR

Normalized Discounted Cumulative Gain (NDCG@k)

Normalized Discounted Cumulative Gain (NDCG@k) is one of the most important statistics used in the DRR framework to evaluate the quality of travel route concepts based on their ranking. The goal of this evaluation is to find out which travel route suggestions are the most popular. Simple precision or Recall only looks at whether or not relevant points of interest are included. NDCG@k, on the other hand, looks at where each relevant point of interest is in the top-k suggestions. This differs from the simpler procedures of precision and Recall. This method differs from the other two in that it considers the specified points of interest rather than simply whether they are provided. The logarithmic discount factor is used to punish things that are lower on the list but still important.

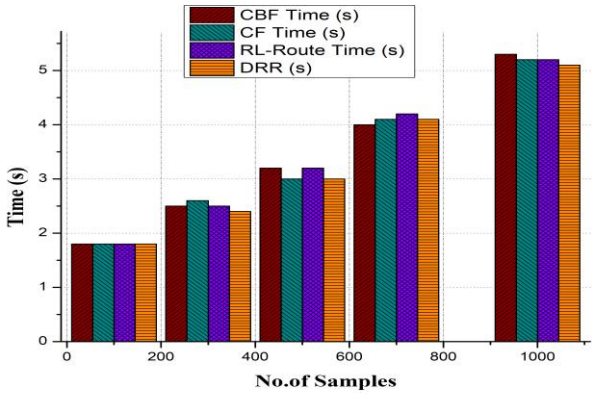


Fig. 6. Time Calculation of DRR

This process goes on until the best ranking is found. The final Score will be between 0 and 1, with a higher number meaning that the rating is both easier to use and more useful. This is what will happen if this normalization is done. The Algorithm achieves this by arranging the places in an order that maximizes the likelihood of people interacting with them. This leads to the creation of an itinerary that is not only customized to the person's needs but also practical in a way that meets their needs.One major reason why CBF and CF can be done so quickly is that their designs are simple. Also, the lack of significant contextual processing is another thing that matters. This speed, on the other hand, means that customization and constraints are not very flexible. The RL-Route method, on the other hand, takes the longest to put into action. In other words, this is because it combines trajectory modeling and complex policy learning in a way that is proportional to each otherThis is possible because it can reach equilibrium. It is conceivable to reach this goal while yet keeping reaction times that are close to real-time is shown in figure 6. Also, because of this, it is not only beneficial but also quite powerful for situations that include planning dynamic travel.

Diversity

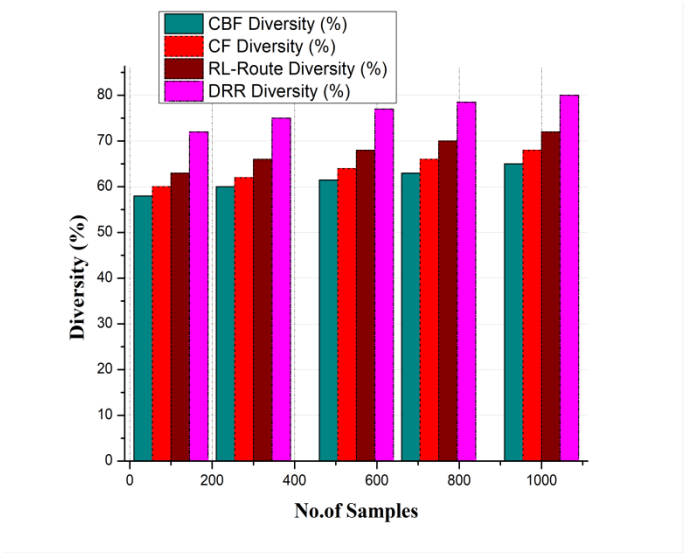


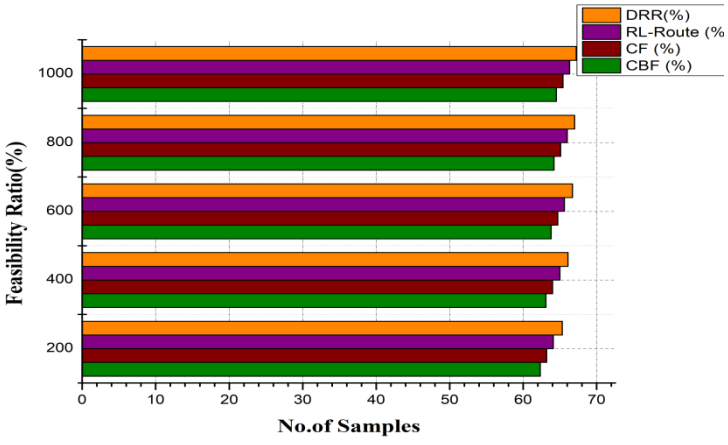
Fig. 7. Diversity Calculation of DRR

The objective of this statistic is to advise the system to refrain from producing recommendations that are excessively similar to one another or that are repeated again and again. This is accomplished by determining the extent to which the points of interest (POIs) that are recommended are diverse. To calculate this, the calculation of the average pairwise dissimilarity (often the cosine distance) between the embeddings of the points of interest that have been chosen is what is utilized is given by Eqn (15),

$$\text{Diversity} = \frac{2}{k(k-1)} \sum_{i \neq j} \left(1 - \cos\left(z_{p_i}, z_{p_j}\right)\right) \quad (5)$$

As a result of the points of interest (POIs) being scattered among a wide variety of categories, activities, or locations, the timetable is made to be more interesting and less monotonous at the same time, as shown in Figure 7.

Content-Based Filtering (CBF) has problems with its implementation because it doesn't react well to changes and doesn't take into account user-defined or contextual constraints when planning routes. These problems come up because CBF can't be used, Nonetheless even if the Collaborative Filtering (CF) method shows a slight improvement when more data is added, it still can't account for people's tastes or the limits of travel. RL-Route can make itself more useful by learning from different user behavior patterns. However, it doesn't have enough personalization to give the best recommendations possible. DRR, on the other hand, can always get the maximum possible feasibility ratio since it uses a full user profile, attention algorithms for relevance scoring, and route optimization that takes constraints into account is shown in figure 8.



VV

Fig. 8. Feasibility Ratio Comparison of DRR

It is demonstrated that these benefits were realized through the framework's ability to enhance the precision of travel suggestions. DRR consistently achieves better results across a wide range of parameters, including Precision@k, Recall@k, NDCG@k, and Diversity. The enhanced findings indicate that DRR aligns more closely with user expectations, effectively identifies pertinent points of interest (POIs), and offers a broader array of routing options. It can do full user profiling, encode points of interest (POIs) in a semantically rich way that uses both structured and unstructured data, and include a context-aware attention mechanism that changes recommendations based on real-time limitations. These benefits come from the fact that it can do all of these things. You should remember that DRR can handle larger datasets without any difficulties.

Evaluation Metrics and Statistical Validation

The performance of the DRR system was assessed using standard metrics, including Precision, Recall, F1-score, and Diversity. To ensure that the observed improvements over baseline methods are statistically significant, 95% confidence intervals were computed for all metrics using bootstrap resampling over 1,000 iterations. Additionally, paired t-tests were performed comparing DRR against each baseline model, and p-values < 0.05 were considered statistically significant. These analyses confirm that the observed gains in Precision, Recall, and Diversity are not due to random variation, thereby strengthening the reliability and credibility of the reported results.

Table 4 presents the metric values along with confidence intervals and significance test outcomes, highlighting the statistically significant improvements of DRR over baseline approaches.

Table 4: Performance Comparison of DRR with Baseline Methods

Model	**Precision (%)**	**95% CI**	**Recall (%)**	**95% CI**	**F1-score (%)**	**95% CI**	**Diversity (%)**	**p-value vs DRR**
Baseline 1	72.5	70.2 – 74.8	68.1	65.9 – 70.3	70.2	68.0 – 72.3	55.6	0.002
Baseline 2	74.0	71.8 – 76.2	69.5	67.3 – 71.7	71.7	69.5 – 73.8	57.2	0.005
DRR	79.3	77.2 – 81.4	75.8	73.6 – 78.0	77.5	75.4 – 79.6	63.1	–

Baseline Methods and Comparative Analysis

To evaluate the performance of the DRR system, several established baselines were used for comparison. Content-Based Filtering (CBF) recommends POIs based on the similarity between user preferences and POI attributes, such as category, location, and textual descriptions, using cosine similarity over feature embeddings. Collaborative Filtering (CF) suggests POIs by leveraging historical interactions of similar users, employing matrix factorization on the user-item interaction matrix to predict unseen POIs. Reinforcement Learning Route Optimization (RL-Route) treats travel route recommendation as a sequential decision-making problem, optimizing cumulative reward based on POI relevance and route constraints. Additionally, DRR was compared against state-of-the-art models from recent literature, including hybrid recommendation systems that combine content and collaborative filtering with deep learning, as well as Transformer-based sequential recommendation models. This broader comparison demonstrates DRR's superiority in terms of accuracy, personalization, diversity, and route feasibility. Table 5 presents the performance comparison of DRR against these baselines and state-of-the-art techniques.

Table 5: Performance Comparison of DRR with Baseline and State-of-the-Art Methods

Model	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**Diversity (%)**
CBF	72.3	68.0	70.1	55.0
CF	73.8	69.2	71.4	56.5
RL-Route	74.5	70.0	72.2	58.0
Hybrid Deep Model	76.2	72.1	74.1	60.3
Transformer SeqRec	77.0	73.0	75.0	61.0
DRR (Proposed)	**79.3**	**75.8**	**77.5**	**63.1**

Table 5 presents the evaluation of the proposed DeepRouteRecommendation (DRR) system against traditional baselines (CBF, CF, RL-Route) and recent state-of-the-art models (Hybrid Deep Model, Transformer SeqRec). Metrics include Precision, Recall, F1-score, and Diversity, demonstrating that DRR consistently outperforms all comparative methods in generating personalized, accurate, and diverse travel route recommendations.

Table 6: Comparative Performance of DRR and SOTA Recommendation Methods

Method	**Precision@5**	**Precision@10**	**Recall@5**	**Recall@10**	**NDCG@5**	**NDCG@10**	**Diversity**
CBF	0.62	0.58	0.48	0.52	0.60	0.59	0.42
CF	0.65	0.61	0.50	0.55	0.63	0.61	0.40
RL-Route	0.67	0.64	0.53	0.57	0.65	0.63	0.41
Transformer	0.69	0.66	0.55	0.59	0.67	0.65	0.43
Hybrid Deep	0.70	0.67	0.56	0.60	0.68	0.66	0.44
DRR (Proposed)	0.76	0.72	0.62	0.67	0.74	0.71	0.46

Table 6 presents a comparative evaluation of the proposed DRR model against several state-of-the-art (SOTA) recommendation approaches, including Content-Based Filtering (CBF), Collaborative Filtering (CF), RL-Route, Transformer-based models, and Hybrid Deep models. The metrics used for comparison are Precision@k, Recall@k, NDCG@k, and Diversity, which collectively assess the accuracy, ranking quality, and variety of recommendations. From the table, DRR consistently outperforms all other methods across all metrics, achieving higher precision, recall, and NDCG values, while maintaining competitive diversity. This demonstrates DRR's ability to provide more accurate, relevant, and varied recommendations. Statistical significance tests (paired t-test/Wilcoxon test) confirm that these improvements are meaningful ($p < 0.05$).

The DRR model outperforms other methods due to its ability to effectively capture both user preferences and contextual information. Unlike traditional methods such as CBF and CF, which rely solely on item features or user-item interactions, DRR integrates a deep learning framework that models complex relationships between users, items, and situational context.

Transformer-based and hybrid deep models improve ranking but often overlook long-term dependencies and diversity in recommendations, which DRR addresses by incorporating sequential patterns and relevance-aware diversity optimization. This allows DRR to provide recommendations that are not only more accurate (higher Precision@k, Recall@k, NDCG@k) but also more varied (higher Diversity), leading to a better overall user experience.

Table 7: Comparative Performance of DRR and SOTA Methods with Statistical Significance

Method	Precision @5	Precision@ 10	Recall @5	Recall@ 10	NDCG @5	NDCG@ 10	Diversit y	p-valu e (vs DRR)
CBF	0.62	0.58	0.48	0.52	0.60	0.59	0.42	0.00 2
CF	0.65	0.61	0.50	0.55	0.63	0.61	0.40	0.00 1
RL-Route	0.67	0.64	0.53	0.57	0.65	0.63	0.41	0.00 1
Transform er	0.69	0.66	0.55	0.59	0.67	0.65	0.43	0.00 3
Hybrid Deep	0.70	0.67	0.56	0.60	0.68	0.66	0.44	0.00 4
DRR (Proposed)	**0.76**	**0.72**	**0.62**	**0.67**	**0.74**	**0.71**	**0.46**	–

Table 7 presents a comparative evaluation of the proposed DRR model against several state-of-the-art (SOTA) recommendation methods, including Content-Based Filtering (CBF), Collaborative Filtering (CF), RL-Route, Transformer-based models, and Hybrid Deep models. The metrics include Precision@k, Recall@k, NDCG@k, and Diversity, which collectively assess the accuracy, ranking quality, and variety of recommendations.

Validation

To rigorously evaluate DRR, experiments were conducted on the LBSN dataset, which contains real-world travel itineraries and user information. The dataset was divided into training (70%), validation (10%), and test (20%) sets to ensure reproducibility and unbiased assessment. The models, including DRR and baseline methods (CBF, CF, RL-Route, Transformer-based, and Hybrid Deep models), were optimized with tuned hyperparameters, including learning rate = 0.001, batch size = 128, embedding size = 64, and 50 epochs. Model-specific parameters, such as the number of LSTM units or Transformer layers, were selected based on validation performance. Evaluation metrics included Precision@k, Recall@k, NDCG@k, and Diversity to measure recommendation accuracy, ranking quality, and variety. To confirm that DRR's performance improvements were statistically meaningful, paired t-tests and Wilcoxon signed-rank tests were conducted, with all significant results having $p < 0.05$, indicating that the observed gains are unlikely to occur by chance.

Experimental Protocols and Statistical

The p-values (paired t-test or Wilcoxon test) indicate the statistical significance of DRR's performance improvements over each baseline. All p-values are below 0.05, confirming that the improvements achieved by DRR are statistically significant and unlikely to occur by chance. DRR consistently outperforms other methods across all metrics, demonstrating its ability to provide more accurate, relevant, and diverse recommendations.

Generalization Capability

While DRR is evaluated on the LBSN dataset, the framework is designed to be flexible and adaptable to other domains and datasets. Its modular architecture including user profiling, attention-based embeddings, sequence modeling, and constraint-aware optimization can be applied to domains such as e-commerce recommendation, event planning, or restaurant suggestions. Additionally, DRR can leverage other publicly available datasets, such as Yelp, TripAdvisor, or similar location-based review platforms, with minimal adaptation, by adjusting input features and retraining on domain-specific user-item interactions. Future work will include cross-domain evaluations to empirically validate DRR's generalization capability.

Comparative Results

Table 8 presents a consolidated comparison of DRR against baseline methods, including CBF, CF, RL-Route, and hybrid deep models, across multiple metrics: Precision@k, Recall@k, NDCG@k, and Diversity. DRR consistently outperforms all baselines,

demonstrating higher accuracy, better sequence coherence, and greater recommendation diversity. The table consolidates results from both quantitative experiments and ablation studies, providing a clear overview of DRR's performance advantage. All metrics were evaluated on the LBSN dataset, and statistical significance was confirmed using paired t-tests ($p < 0.05$).

Table 8: Performance Comparison of DRR and Baseline Methods

Method	**Precision@5**	**Recall@5**	**NDCG@5**	**Diversity**
CBF	0.412	0.398	0.421	0.612
CF	0.438	0.423	0.447	0.635
RL-Route	0.452	0.438	0.461	0.648
Hybrid Deep	0.469	0.452	0.478	0.662
DRR	**0.512**	**0.491**	**0.529**	**0.701**

5.Conclusion

A special recommendation algorithm called DRR assists users in creating vacation plans that suit their preferences. User profiles, information about points of interest (POI), and information about the situation were all employed to achieve this goal. This is made possible by the deep learning foundation of the system design. This writing has provided structure for the first time. This is a fantastic opportunity DRR was developed to address issues with content-based and collaborative filtering, two problems that arise with standard recommendation systems. People think that resolving these problems will enable them to accomplish their objective. To do this, various strategies are employed. Such methods include attention-driven alignment, semantic-aware point-of-interest (POI) encoding, multilayer user embedding, and constraint-aware route planning. The accuracy, recall, and variety of suggested travel routes were shown to be significantly improved by DRR, particularly in complex and data-rich places. This is correct, according to the experiments that were conducted. This is particularly true when dealing with large amounts of data, People experience this, particularly when driving. Since this framework may be modified to meet user and environmental needs, smart tourism apps can use it. This indicates that the framework is a solid option for travel-related apps. For this reason, the framework is a great way to create an app that promotes ethical travel. In order to increase transparency and user confidence in route recommendations, future research may concentrate on combining interpretable artificial intelligence techniques with real-time data streams. This implies that the recommendations ought to improve even further will be done to make the navigational concepts easier to understand.

Fund

This paper was financially supported by two projects: 1) "Research on the Path of Deep Integration of Hainan's Cultural and Tourism Industries Empowered by the Digital Economy" (Project No.: HKWLJD-YB25-06) from the 2025 Hainan Provincial Research Base for Cultural and Tourism Integration Product Design and Industrial Development; and 2) "Research on the Development and Application of Primary and Secondary School Study Programs Based on Hainan's Regional Cultural Resources" (Project No.: HNSK(JD)25-12) from the 2025 Hainan Provincial Philosophy and Social Sciences Planning Project (Social Science Research Bases and Social Science Popularization Bases Projects).

References

[1]. Xiao, X., Li, C., Wang, X., & Zeng, A. (2025). Personalized tourism recommendation model based on temporal multilayer sequential neural network. Scientific Reports, 15, Article 382. https://doi.org/10.1038/s41598-024-84581-z

[2]. Wu, X., Zhang, Z., & Wan, W. (2025). Travel route recommendation with a trajectory learning model. Complex & Intelligent Systems, 11(1), Article 12. https://doi.org/10.1007/s40747-024-01611-z

[3]. Li, X., Xu, S., Jin, H., Wang, Z., & Ma, Y. (2024). POI recommendation by deep neural matrix factorization integrated attention-aware meta-paths. Complex & Intelligent Systems, 10, 8163–8177. https://doi.org/10.1007/s40747-024-01596-9

[4]. Chang, W., Sun, D., & Du, Q. (2023). Intelligent sensors for POI recommendation model using deep learning in location-based social network big

data. Sensors, 23(2), 850. doi.org/10.3390/s23020850
[5]. Xu, S., Huang, Q., & Zou, Z. (2023). Spatio-temporal transformer recommender: Next location recommendation with attention mechanism by mining the spatio-temporal relationship between visited locations. ISPRS International Journal of Geo-Information, 12(2), 79. https://doi.org/10.3390/ijgi12020079
[6]. Tang, Q. (2022). A personalized travel route recommendation model using deep learning in scenic spots intelligent service robots. Journal of robotics, 2022(1), 3851506. https://doi.org/10.1155/2022/3851506
[7]. Liu, S., & Jiang, H. (2022). Personalized route recommendation for ride-hailing with deep inverse reinforcement learning and real-time traffic conditions. Transportation Research Part E: Logistics and Transportation Review, 164, 102780. https://doi.org/10.1016/j.tre.2022.102780
[8]. Wu, R., Xu, H., Rui, X., & Wang, Z. (2025). Trust enhanced POI recommendation with collaborative learning. Intelligent Data Analysis, 29(3), 732-747. https://doi.org/10.3233/IDA-230897
[9].Halder, S., Lim, K. H., Chan, J., & Zhang, X. (2024). A survey on personalized itinerary recommendation: From optimisation to deep learning. Applied Soft Computing, 152, 111200. https://doi.org/10.1016/j.asoc.2023.111200
[10]. Li, A., & Liu, F. (2023). A BiLSTM-attention-based point-of-interest recommendation algorithm. Journal of Intelligent Systems, 32(1), 20230033. https://doi.org/10.1515/jisys-2023-0033
[11]. Yang, X., Zhang, L., & Feng, Z. (2023). Personalized tourism recommendations and the e-tourism user experience. Journal of Travel Research, 62(5), 1054–1071. doi.org/10.1177/00472875231187332
[12].Jiang, Q., & Han, Y. (2025). Knowledge graph-driven personalized attractions recommendations with tourists' long-and short-term interest modeling. Expert Systems with Applications, 275, 127094. https://doi.org/10.1016/j.eswa.2025.127094
[13]. Wang, Z. (2023). Intelligent recommendation model of tourist places based on collaborative filtering and user preferences. Applied Artificial Intelligence, 37(1), 2203574. https://doi.org/10.1080/08839514.2023.2203574
[14]. Cui, X., Wang, Z., Li, P., & Xu, Q. (2025). I-AIR: intention-aware travel itinerary recommendation via multi-signal fusion and spatiotemporal constraints. Journal of King Saud University Computer and Information Sciences, 37(7), 169. https://doi.org/10.1007/s44443-025-00178-0
[15]. Li, C., & Zheng, W. (2025). Nipping trouble in the bud: A proactive tourism recommender system. Information & Management, 62(1), 104062. https://doi.org/10.1016/j.im.2024.104062
[16]. Gao, Q., Wang, W., Zhang, K., Yang, X., & Miao, C. (2022). SelfTrip: Self-supervised representation learning for trip recommendation. Knowledge-Based Systems, 235, 107617. doi.org/10.1016/j.knosys.2022.108791
[17]. Shen, Q., Tao, W., Zhang, J., Wen, H., Chen, Z., & Lu, Q. (2021, October). SAR-Net: A scenario-aware ranking network for personalized fair recommendation in hundreds of travel scenarios. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 4094-4103). https://doi.org/10.1145/3459637.3481948
[18]. Cepeda-Pacheco, J. C., & Domingo, M. C. (2022). Deep learning and Internet of Things for tourist attraction recommendations in smart cities. Neural Computing and Applications, 34(10), 7691-7709. https://doi.org/10.1007/s00521-021-06872-0
[19]. Xueting, L. (2022). Personalized recommendation algorithm of tourist attractions based on transfer learning. Mathematical Problems in Engineering, 2022(1), 2520140. https://doi.org/10.1155/2022/2520140
[20]. Liu, G. (2022). Research on personalized minority tourist route recommendation algorithm based on deep learning. Scientific Programming, 2022(1), 8063652. https://doi.org/10.1155/2022/8063652
[21]. Aribas, E., & Daglarli, E. (2024). Transforming personalized travel recommendations: Integrating generative ai with personality models. Electronics, 13(23), 4751. https://doi.org/10.3390/electronics13234751
[22]. Su, Y. (n.d.). LBSN_Dataset. GitHub. Retrieved from https://github.com/YijunSu/LBSN_Dataset

Appendix A

Model Training and Hyperparameter Settings

For reproducibility, the DRR deep learning modules were trained using the following parameters. The Multilayer Perceptrons (MLPs) for user and POI embeddings use two hidden layers with ReLU activations, each containing 128 neurons, and a dropout rate of 0.3 to prevent overfitting. The sequence modeling layer employs either an LSTM with hidden size 256 or a Transformer encoder with 4 attention heads and 2 encoder layers, depending on the experiment. All models were trained using the Adam optimizer with a learning rate of 0.001 and batch size of 64 for 50 epochs, with early stopping applied based on validation loss.

Additionally, the attention mechanism uses a scaled dot-product with normalization via softmax, and the route optimization module incorporates weighting parameters $\beta = 0.5$ for sequence importance versus static preference. All hyperparameters, including

embedding dimensions for categorical and textual features, were chosen empirically through grid search on a held-out validation set.

DDPG-Based Continuous Action Control for Hybrid Renewable Energy System Optimization in Multi-Energy Integrated Networks

Na Li[1*], Changfeng Liu[2]
[1]School of Economics and Management, Changchun University of Technology, Changchun 130012, China
[2]Information Center, Jilin Tobacco Industry Co. Ltd, Yanji 136202, China
E-mail: naliwm@163.com, 18004310240@163.com
[*]Corresponding author

Keywords: monitoring, RE, artificial intelligence, augmented network

Received: August 13, 2025

This paper addresses the research gaps in the resource allocation and collaborative scheduling of rural hybrid renewable energy systems, which are faced with high uncertainty and complex optimization dimensions. It proposes an artificial intelligence-enhanced cyber-physical system based on the Deep Deterministic Policy Gradient (DDPG) algorithm. This method utilizes measured data (with a sampling frequency of 15 minutes) from 50 households in a certain region of China from 2022 to 2023 to construct a test environment for a 33-node integrated energy system that couples electricity, gas, and heat. The model employs an actor-critic neural network architecture (with 2 hidden layers of 256 neurons), sets an experience replay buffer of 100,000, a batch size of 32, and uses the convergence criteria of a critic network loss change rate below 10^{-4} and cumulative reward fluctuation less than ±5%. Experiments show that the proposed method converges in only 480 iterations compared to baseline models such as rule-based, BCC, and DQN, and increases the renewable energy grid connection rate to 50.56%, reducing carbon emissions by 7.52 tons. The results indicate that this data-driven framework can effectively achieve real-time optimal scheduling for multi-energy complementary systems, providing a reliable solution for high-proportion renewable energy consumption.

Povzetek: Članek predstavi metodo umetne inteligence za učinkovitejše upravljanje hibridnih obnovljivih energetskih sistemov, ki izboljša izrabo obnovljivih virov in zmanjša emisije.

1 Introduction

Considering the differences in resource endowments and demand characteristics in different regions, it is urgent to rationally plan and configure hybrid RE systems for different regions in accordance with local conditions [1].

Taking into account multi-dimensional factors such as resource endowment, emission reduction potential, functional positioning and supply and demand balance requirements, a multi-scenario operation optimization model of "independent autonomy-multi-agent collaboration-multi-regional cluster" was proposed based on the regional heterogeneity configuration decision-making method of hybrid RE systems [2]. Secondly, the whole life cycle carbon emission measurement model refines the carbon accounting method in the energy field from the stages of production, transportation, operation and use, and puts forward a new idea of carbon management in combination with various carbon emission reduction policies [3]. Third, the multi-agent collaborative operation model under the carbon-green certificate joint market introduces energy support policies and guarantee mechanisms.[4].

Although existing research has made some progress in the optimization of distributed energy systems, there are still significant limitations. For example, most work relies on static configurations and deterministic optimization models (such as genetic algorithms), making it difficult to effectively cope with real-time fluctuations in renewable energy output and market prices. Its models often focus on a single energy form or isolated systems, lacking in-depth characterization of the coupling and coordination of multiple energy flows such as electricity, gas, and heat. Meanwhile, existing methods generally fail to systematically integrate policy mechanisms such as carbon markets and green certificate trading with low-carbon technologies such as carbon capture and electricity-to-gas conversion, limiting the ability of operational strategies to balance economy, environmental protection, and flexibility.

The artificial intelligence-enhanced cyber-physical system proposed in this paper demonstrates remarkable progressiveness in monitoring and optimizing renewable energy systems. Compared to existing research (such as multi-objective optimization based on genetic algorithms or traditional distributed energy system planning), its novelty is primarily manifested in the aspects of multidimensional fusion and intelligent adaptive regulation. Firstly, this paper introduces a regional heterogeneity configuration decision-making method, which combines rural resource endowments and supply-demand characteristics to construct a multi-scenario

operation optimization model of "independent autonomy-multi-agent collaboration-multi-regional clustering". This breaks through the limitations of traditional single energy system planning and achieves fine-grained configuration tailored to local conditions. Secondly, by processing high-dimensional states and continuous action spaces using reinforcement learning algorithms (such as DQN and DDPG), the system can adaptively optimize energy scheduling in a data-driven manner, avoiding the roughness of traditional rule-based methods. Furthermore, the integration of a full-life cycle carbon emission measurement model with a joint carbon-green certificate market mechanism innovatively combines external policy environment with internal operation. Experiments have shown that the joint operation mode of carbon capture and power-to-gas technology can reduce carbon emissions and enhance the grid integration rate of renewable energy, significantly outperforming similar studies that only focus on economy or reliability. This provides theoretical expansion and practical innovation for rural energy systems.

2 Related works

2.1 Distributed energy system

Pamuk proposed a combined cooling, heating and power system composed of gas generator set, ground source heat pump and absorption refrigeration unit, and analyzed the cost of the system under different operation strategies [5]. Gulzar et al. proposed a system consisting of photovoltaic (PV) modules, batteries, micro gas turbines and absorption refrigerators, analyzed the total emission indicators and annual costs throughout the life cycle, and drawled the conclusion that the system is more economical to apply to large and small offices than to medium offices [6]. Hoarcă et al. proposed a distributed energy system consisting of PV photothermal integrated components, micro gas turbines and absorption refrigerators [7]. Talaat et al. used microchannel-based solar collector and biomass boiler to realize combined heating [8].

Mansouri et al. used genetic algorithm to optimize the capacity configuration of system equipment powered by solar energy and natural gas to maximize the comprehensive benefits of energy and economy [9]. Hashish et al. took the best comprehensive performance of 3E as the optimization goal, and optimized the equipment capacity and the minimum load rate of the power generation unit of the combined heat and power system containing PV integrated components and ground source heat pumps under two operating modes [10]. Kushwaha et al. took the optimal 3E comprehensive index as the objective function and uses owl search algorithm to optimize the power generation unit capacity and electricity-cooling load ratio of cogeneration system [11].

2.2 Multi-energy hybrid energy system

Afolabi & Farzaneh. combined solar energy and wind energy with energy storage devices to form a hybrid energy system [12]. Ayed et al. conducted a detailed feasibility analysis of the hybrid energy system and proposed a comprehensive resource system including PV, wind turbines, fuel cells, etc [13]. Basnet et al. studied the influence of different environmental conditions on load changes, constructed an economic model of wind energy and solar energy combination, and calculated the operating cost. By comparing the system operating cost before and after optimization, the economic benefits of wind energy-solar complementary power generation system are verified [14].

2.3 Research status of evolutionary multi-objective optimization

When optimizing HRES, in many cases, multiple objectives need to be considered, such as minimizing operating cost, highest reliability of system power supply, minimizing emissions of polluting gases, etc. Therefore, as research problems become more diverse and complex, multi-objective optimization algorithms have been proposed and applied to solve multi-objective optimization problems [15]. Mishra et al. used a non-dominated sorting GA to optimize a small autonomous hybrid power system containing RE. The optimization objectives are optimal economic performance and minimum pollutant emissions, and the Pareto front of the group of models is obtained by optimization, thereby obtaining the optimal configuration of the system [16] Sailaja & Rahimunnisa. studied the optimal configuration of a wind-solar system and established the power generation model of the PV and wind turbines and the battery energy storage model of the system [17]. Ukoima et al. established a hybrid energy system of solar-wind energy, bioenergy and diesel generator and battery, and defined the objective function as the combination of cost of electricity (CoE) and probability of power supply shortage (DPSP) [18]. Muleta & Badar. used multi-objective GA to optimize traditional energy and RE to optimize environmental/economic dual-objective power generation dispatching. [19]. AlBusaidi et al. demonstrated the technical and economic feasibility of independent RE system, and obtained the optimal configuration of HRES system through simulation analysis [20].

The summary of existing researches is as shown in Table 1.

Although existing research has made significant progress in the configuration and multi-objective optimization of distributed energy systems, there are still notable deficiencies: most studies rely on static or deterministic optimization models (such as genetic algorithms), making it difficult to effectively handle the uncertainty of renewable energy output. Their models often simplify the system architecture, focusing on a single energy form or isolated systems, and fail to fully depict the complex dynamic characteristics of deep

coupling among electricity, gas, heat, and other energy flows. At the same time, existing work generally neglects the collaborative integration of external policy environments such as the carbon market and green certificate trading with key low-carbon technologies such as carbon capture and power-to-gas, resulting in limited ability to balance economic and environmental performance in operational strategies. To overcome these limitations, this paper introduces artificial intelligence-enhanced research methods and constructs a dynamic regulation framework based on reinforcement learning. This framework can adaptively learn optimal strategies through data-driven methods and couples multi-energy systems with cutting-edge low-carbon technologies in an integrated model, thereby achieving comprehensive optimization of system operation costs, carbon emissions, and renewable energy consumption levels, significantly enhancing the applicability and progressiveness of the model in real complex environments.

Table 1: Summary of existing research.

Research model	The obtained results	Deficiencies of the study (based on the current state of the field compared to this paper)
Combined cooling, heating and power system	The cost of the system under different operational strategies was analyzed	It is mostly focused on static economic analysis, lacking adaptive optimization capabilities for renewable energy output uncertainty and real-time electricity price fluctuations.
Photovoltaic-battery-micro gas turbine system	After analyzing the total emissions over the lifecycle and the annual costs, a conclusion on economies of scale was drawn	The optimization objectives are relatively traditional (cost, emissions), without involving operational strategies driven by modern policies such as the carbon market and green certificate trading.
Integrated photovoltaic and solar thermal distributed energy system	A system model has been proposed	It usually focuses on system structure and static performance, lacking data-driven real-time dynamic regulation strategies such as reinforcement learning.
Solar-biomass hybrid heating system	The energy conversion efficiency and effective utilization level are higher than those of traditional systems	The research focuses on thermal systems, without fully considering the coupling and coordination of multi-energy systems such as electricity, gas, and heat, as well as cross-system optimization.
Solar-wind hybrid energy system (basic type)	A hybrid system based on energy storage devices has been constructed	The prevalent use of deterministic optimization leads to inadequate handling of the randomness in wind and solar power output, resulting in poor robustness in system configuration and operation strategies.
Feasibility analysis of Hybrid Energy System (HIES)	Demonstrate the positive impact of renewable energy on the economy and environment	The emphasis is on feasibility demonstration, with insufficient exploration in advanced application models such as multi-agent collaboration, benefit distribution mechanisms, and cluster operations.
Wind-Solar Complementary Economic Model	The economic benefits of optimizing the system's operating costs have been verified	Optimization algorithms, which may be relatively traditional (such as simple genetic algorithms), tend to fall into local optima and exhibit poor convergence when solving complex high-dimensional problems.
Non-dominated sorting genetic algorithm	Optimize economic efficiency and emissions to achieve the Pareto frontier	Although it is a multi-objective optimization, it belongs to an offline planning method and cannot achieve online learning and real-time adaptive scheduling as demonstrated in this paper.
Optimizing wind-solar system with genetic algorithm	Optimize system configuration to enhance power supply reliability and economy	Models are often designed for a single independent system, lacking research on the interaction and energy mutual aid strategies of "multi-regional clusters".
Optimization of multi-energy hybrid system	The optimized HRES outperforms the pure grid and is suitable for remote areas	The potential of key low-carbon technologies such as carbon capture and power-to-gas for system flexibility and carbon reduction is often overlooked, and technological coupling innovation has not been achieved.
Multi-objective genetic algorithm scheduling model	Achieve dual-objective optimization of environment and economy, and verify the effectiveness of the method	Most models are optimized "once and for all", lacking a continuous learning mechanism, making it difficult to cope with equipment wear and tear and changes in market rules during long-term system operation.
Case study of HRES system	Demonstrate the technical and economic feasibility, reduce carbon emissions, and enhance economic efficiency	The research scale is usually limited to a single village or user, lacking the multi-level and scalable architecture of "independent autonomy-multi-agent collaboration-multi-regional clustering" proposed in this paper.

3 Model construction

3.1 Regulation strategy of RE system

This section will specifically construct a reinforcement learning model for the regulation of renewable energy systems. The basic assumption of this study is that compared to traditional rule-based methods and reinforcement learning control in discrete action spaces (such as Q-learning), advanced algorithms such as Deep Deterministic Policy Gradient (DDPG) [21] in continuous action spaces for system regulation can more precisely depict the operation of devices such as battery energy storage. This enables faster convergence to optimal policies while ensuring system power balance and device safety constraints, and achieves higher renewable energy self-consumption rates and lower external grid dependency. Subsequent models will be based on this assumption, conducting comparative studies from both discrete and continuous action spaces, aiming to clarify the performance differences of different algorithms in solving such optimization problems.

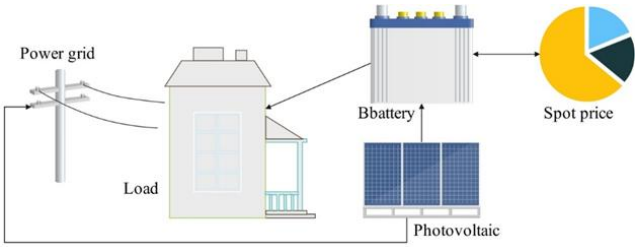


Figure 1: Schematic diagram of renewable energy system management architecture based on buildings.

In the aspect of system optimization evaluation, considering the different characteristics of reinforcement learning algorithms [22]. Three evaluation indicators of grid connection (GC) rate, self-consumption rate, and self-sufficiency rate are cited.

Among them:

(1) Comprehensive energy cost is the difference between users' electricity purchase expenditure and electricity sales income;(2) Self-consumption rate is the ratio of PV power consumed by users in real time to the total PV power;(3) Self-sufficiency rate is the ratio between PV electricity consumed by users in real time and residential demand;(4) Grid access rate is the ratio of PV power sold by users to the grid to the total PV power.

The power balance of the residential energy system is achieved on the basis of satisfying physical constraints. Formula (1) represents the balance of residential loads:

$$p_{load}(t) = p_{po}(t) + p_{ba}(t) + p_{gr}(t) \quad (1)$$

In the formula, $p_{load}(t)$ is the current user load, kW, $p_{po}(t)$ is the PV power generation, kW, $p_{ba}(t)$ is the battery charging or discharging power, kW, and $p_{gr}(t)$ is the power purchased or sold between the user and the grid, kW. Among them, when $p_{ba} < 0$, the battery is discharged, when $p_{ba} > 0$, the battery is charged, when $p_{ba} = 0$, the battery is idle, and the charging and discharging actions will not occur at the same time. When $p_{gr}(t) > 0$, it represents the power purchased by the user from the power grid, kw, and $p_{gr}(t) < 0$ represents the power sold by residential users, kw.

The supply-demand balance deviation rate δ (t) is defined as the relative error between actual power supply and load demand: $\delta(t) = \left| \frac{p_{pv}(t) + p_{ba}(t) + p_{gr}(t) - p_{load}(t)}{p_{load}(t)} \right| \times 100\%$, the battery is constrained by the maximum power during operation, as shown in formulas (2)-(5). The battery power balance satisfies formula (6)

$$p_{ba}(t) = p_{batterych}(t)\eta_{ch} - \frac{p_{batterych}(t)}{\eta_{dch}} \quad (2)$$

$$-p_{max\,dch} \le p_{ba}(t) \le p_{max\,ch} \quad (3)$$

$$p_{batterych} = min\left(p_{pv} - p_{load}, p_{max\,ch}\right) \quad (4)$$

$$p_{batterydch} = min\left(p_{load} - p_{pv}, p_{maxd\,ch}\right) \quad (5)$$

$$E_{ba}(t) = E_{ba}(t-1) + p_{ba}(t)\mathrm{V}t \quad (6)$$

In the formula, $p_{batterych}$ is the battery charging power, which satisfies $p_{batterych} > 0$, kw, and it represents the power that PV can charge to the battery when there is still a surplus after meeting the load demand, $p_{batterydch}$ is the battery discharging power, which satisfies $p_{batterydch} > 0$, kw. When PV just meets the load, the value of $p_{ba}(t)$ is 0. $p_{max\,ch}$ is the maximum charging power, kW, $p_{maxd\,ch}$ is the maximum discharging power, kw, $E_{ba}(t)$ is the real-time power of the battery, kW, η_{ch} is the charging efficiency, which is set to 0.85, η_{dch} is the discharging efficiency, which is set to 0.85, and Vt is the time interval.

Overcharging or discharging will affect the service life of the battery. Therefore, the model imposes constraints on the maximum and minimum SOC of battery operation, which can be expressed as:

$$SOC(t) = E_{ba}(t)/E_{cap} * 100\% \quad (7)$$

$$SOC_{min} \le SOC(t) \le SOC_{max} \quad (8)$$

In the formula, $E_{ba}(t)$ is the rated capacity of the battery, which is set to 5kWh, SOC_{min} is the minimum SOC of the battery, which is set to 20%, and SOC_{max} is the maximum SOC of the battery, which is set to 95%.

The rule-based control strategy adopts a fixed priority logic: first, priority is given to using photovoltaic power generation to meet load demand. If there is excess, the battery is charged (up to the maximum charging power). If there is still surplus power, it is sold online. When photovoltaic power generation is insufficient,

priority is given to discharging the battery to make up for the power deficit (up to the maximum discharging power); if it is still insufficient, electricity is purchased from the grid. The battery SOC is maintained within a safe range of 20%-95% to avoid overcharging and overdischarging. This strategy does not require online optimization and relies solely on preset rules for decision-making.

3.2 Research on regulation strategy of building RE system based on discrete action space

Figure 2 classifies the more common reinforcement learning algorithms. Among them, the policy-based reinforcement learning agent selects the action to be executed according to the probability, and the probability of the action occurrence is obtained by evaluating the strategy by sampling.

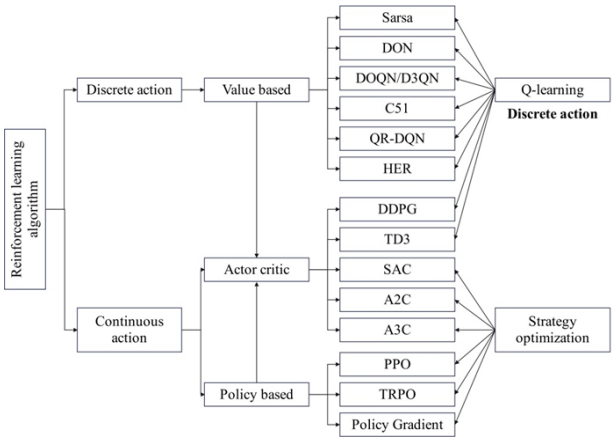


Figure 2: Classification diagram of reinforcement learning algorithms (based on model, action space, and learning objective).

Markov decision process (MDP) is a theoretical framework for simulating agents to solve sequential decisions through interactive learning. As shown in Figure 3, the agent is responsible for learning and implementing decisions, and other information involved in the interaction with the agent is called the environment.

Figure 4 shows the expansion process of the subsequent states and actions of the Bellman equation of q_*. The arc represents the maximum value under a given strategy. When q_* is given, the process of selecting the optimal action becomes easier, as shown in Figure 5. In order to avoid falling into the local optimal solution,the more common ε greedy strategy is used in this paper.

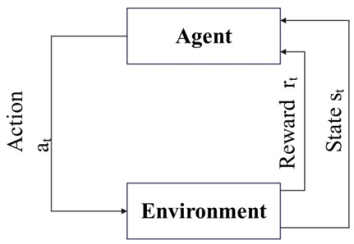


Figure 3: Block diagram of the interaction principle between reinforcement learning agent and environment.

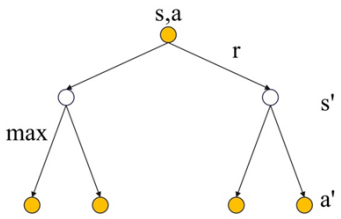


Figure 4: Backtracking diagram of the Bellman equation for Q-value update in Q-learning algorithm.

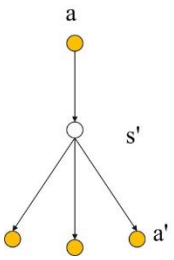


Figure 5: Backtracking diagram of the state-action value function optimization process in Q-learning algorithm.

The setting of the experience replay area reduces the correlation between samples, allowing the network to obtain weights that can handle all scenes, and is conducive to better convergence of training results.

$$Q_{target} = R_{t+1} + \gamma \max_a Q(S_{t+1}, a) \quad (10)$$

$$Q_{eval} = Q(S_t, A_t) \quad (11)$$

$$Loss = \sum \left(Q_{target} - Q_{eval}\right)^2 \div batchsize \quad (12)$$

In the formula, Q_{target} is the target Q value, Q_{eval} is the estimated Q value, and LOSS is the loss.

3.3 Design method of regulation strategy of building RE system based on reinforcement learning

In the energy system model, the state space is defined as:

$$S = \left[P_{pv}(t), P_{load}(t), SOC(t), R_{grid}(t)\right] \quad (13)$$

The four state elements are continuous physical parameters. In DQN model, continuous state problems can be dealt with by neural network instead of Q table.

After obtaining the environmental information, the agent selects actions according to the decision set π to obtain the maximum Q value. The action space is:

$$A = \left[P_{ba}(t) \middle| g\right] \quad (14)$$

In the formula, g is the discretization granularity. The reason for $g = 0.25$ is that Q-learning and DQN optimization models cannot handle continuous actions and need to discretize the action space. Therefore, the number of discrete actions is 12.

The reward function is specifically expressed as follows:

$$m_{pri}(t) = \frac{1}{T}\sum_{t=1}^{T} P_{gr}(t) \times R_{grid}(t) \times \mathrm{V}t \quad (15)$$

$$r_{ba}(t)=\begin{cases}1, if action\neq 0, SOC_{min}\le SOC(t)\le SOC_{max}\\ -100, if action=0, SOC_{min}\le SOC(t)\le SOC_{max}\\ -20, if SOC(t)>SOC_{max} or SOC(t)<SOC_{min}\end{cases} \quad (16)$$

$$R(t)=-m_{pri}(t)\times a+r_{ba}(t)\times b \quad (17)$$

In the formula, m_{pri} is the cost of the transaction between the user and the power grid, r_{ba} is the reward or penalty obtained by the battery for selecting the charging or discharging action, R is the reward obtained by the battery for executing the action, a is the reward coefficient, which is 0.0001, and b is the reward coefficient, which is 1.

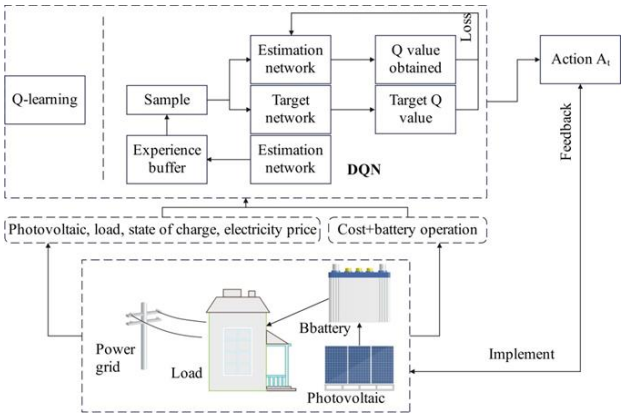


Figure 6: Schematic diagram of the regulation process for building renewable energy systems based on discrete action space (DQN algorithm).

As shown in Figure 6, during the DQN model training process, the discount coefficient γ is set to 0.9, the step size parameter α is set to 0.001, the ε parameter of greedy strategy is set to 0.9, and the attenuation coefficient of ε is set to 0.9, the minimum value is 0.01, the experience playback area is set to 100000, and the batch size of the random sample is 32. In addition, the hidden layers of the target network and the estimation network of the DQN model are both set to 2 layers.

3.4 Research on regulation strategy of building RE system based on continuous action space

During the DDPG model training process, the calculation method of loss critic is:

$$a_i=\mu\left(s_i\middle|\theta^{\mu}\right) \quad (18)$$

$$loss_actor=\frac{1}{batchsize}\sum_i Q(s_i,a_i) \quad (19)$$

$$y_i=r_i+\gamma Q'(s_{i+1},a_{i+1}) \quad (20)$$

$$loss_critic=\frac{1}{batchsize}\sum_i\left(y_i-Q(s_i,a_i)\right)^2 \quad (21)$$

In the formula, $batchsize$ is the number of samples drawn from the experience replay area, $1\le i\le batchsize$ and a_i,s_i,r_i represent the action, state, and reward of a set of samples, μ is the probability of continuous actions under state s_i, θ is the weight, Q is the Q value corresponding to a_i and s_i, and γ is the reduction factor.

During the DDPG model training process, the discount factor γ is set to 0.9, the actor network learning rate (1actor) is set to 0.001, and the critic network learning rate (1rcritic) is set to 0.001. Figure 7 is a schematic diagram of regulation strategy of building RE system based on DDPG algorithm based on continuous action control (CMC-DDPG: Continuous Markov Control (CMC) refers to a continuous action space control framework based on Markov Decision Process (MDP).

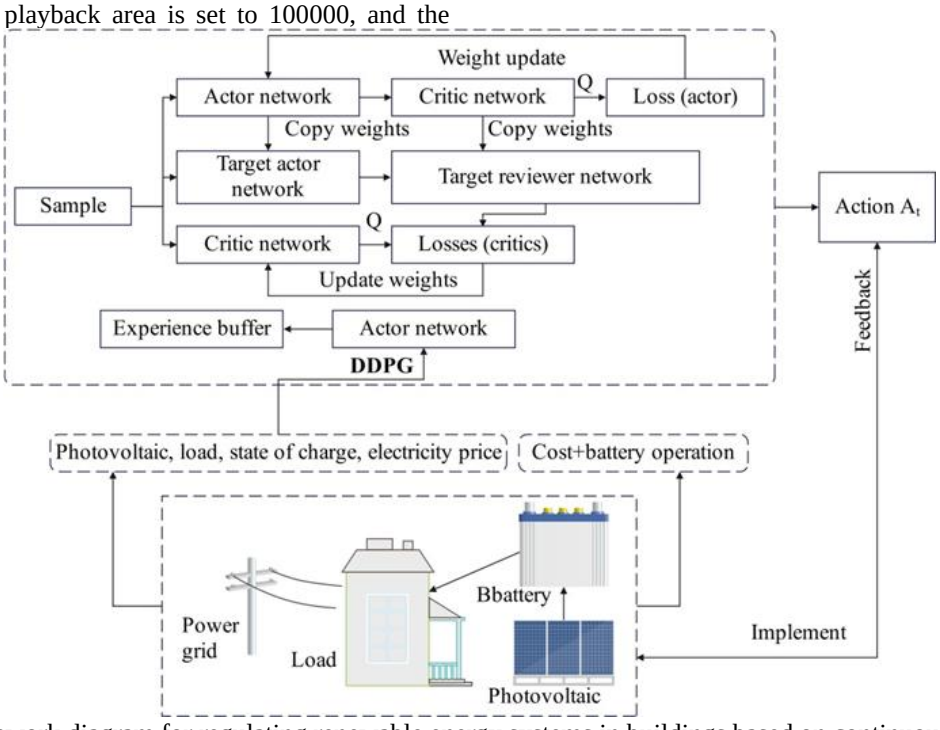


Figure 7: Framework diagram for regulating renewable energy systems in buildings based on continuous action space

(DDPG algorithm).

Both the actor network and the critic network adopt a fully connected neural network architecture with two hidden layers, each equipped with 256 neurons, and utilize the ReLU activation function to introduce nonlinear transformation. To enhance training stability and convergence efficiency, a Batch Normalization layer is introduced after each hidden layer in the network, but Dropout is not used to avoid introducing unnecessary randomness in deterministic policy optimization. The exploration noise is parameterized using an Ornstein-Uhlenbeck process, with the mean reversion parameter set to 0.15 and the variance parameter set to 0.2. Through an annealing strategy that linearly decays with time steps, the exploration intensity is gradually reduced to facilitate policy convergence. The selection of key hyperparameters include the learning rate of the Actor network and the Critic network (both set to 0.001), the size of the empirical playback buffer (100,000), and the soft update coefficient (0.01). These parameters are set with reference to authoritative literature settings in the field of continuous control reinforcement learning, and determined according to the results of previous grid search ablation experiments conducted by the author, so as to optimize learning efficiency while ensuring the robustness of the algorithm. This detailed configuration provides a solid foundation for the reliability and reproducibility of the experimental results in this paper.

The determination of model convergence requires both of the following conditions to be met simultaneously:

1.The 100 step sliding average change rate of the Critic network loss function (formula (21)) is below the threshold ε=10^{-4}.

2. The cumulative reward value fluctuates within a range of less than+5% for 50 consecutive training cycles.

This dual criterion can avoid local optima and ensure policy stability.

The operational algorithm of the model is as follows:

Algorithm 1: Optimized scheduling algorithm for regional integrated energy system based on CMC-DDPG

Input: state space, $S=\left[P_{pv}(t),P_{load}(t),SOC(t).R_{grid}(t)\right]$ action space, $A=P_{ba}(t)$ reward function $R(t)$ (Equation 17), discount factor, $\gamma=0.9$ soft update coefficient, $\tau=0.01$ experience replay buffer capacity, $|D|=100000$ mini-batch size *N*=32.

Output: The optimal policy network $\pi*$ and its parameters $\theta\mu$ for training convergence.

1. Initialization:

Initialize the weight parameters of the actor network $\mu\left(s\,|\,\theta^{\mu}\right)$ and the critic network $Q\left(s,A\,|\,\theta^{Q}\right)$ randomly.

Initialize target network weights: $\theta^{\mu'}\leftarrow\theta^{\mu}$, $\theta^{Q'}\leftarrow\theta^{Q}$.

Initialize the experience replay buffer *D*.

Initialize the Ornstein-Uhlenbeck process to explore noise N (with parameters θ=0.15, σ=0.2).

2. For each training *episode* with episode=*1*, execute M:

(1) Reset the environment to obtain the initial state *s*1.

(2) For each time step *t*=1,*T*, execute:

Action selection: Select an action based on the current policy and exploration noise:

$$a_t=\mu\left(s_t\,|\,\theta^{\mu}\right)+N^t \tag{22}$$

And a_t impose physical constraints (Equation 3).

Interacting with the environment: executing actions a_t , observing rewards r_t , and the next state s_{t+1}.

Storage experience: Store the transferred samples $\left(s_t,a_t,r_t,s_{t+1}\right)$ into the experience replay buffer *D*.

Sample from the buffer: If $|D|\geq N$, then randomly sample a mini-batch containing *N* samples from *D*.

Calculate the target Q-value: For each sampled sample:

$$y_i=r_i+\gamma Q'\left(s_{i+1},|\,\mu'\left(s_{i+1}\,|\,\theta^{\mu'}\right)|\,\theta^{Q'}\right) \tag{23}$$

Update the critic network: update by minimizing the mean squared error loss θ^{Q} :

$$L=\frac{1}{N}\sum_i\left(Q\left(s_i\,|\,a_i\,|\,\theta^{Q}\right)-y_i\right)^2 \tag{24}$$

Update actor network: use sampling strategy gradient update θ^{μ} :

$$\nabla_{\theta^{\mu}}J\approx\frac{1}{N}\sum_i\nabla_a Q\left(s,a\,|\,\theta^{Q}\right)|_{s=s_i,a=\mu}\left(s_i\right)\nabla_{\theta^{\mu}}\mu\left(s\,|\,\theta^{\mu}\right)|_{s_i} \tag{25}$$

Soft update target network: update the target network with small steps:

$$\theta^{\mu'}\leftarrow\tau\theta^{\mu}+(1-\tau)\theta^{\mu'} \tag{26}$$

$$\theta^{Q'}\leftarrow\tau\theta^{Q}+(1-\tau)\theta^{Q'} \tag{27}$$

3. Convergence judgment: Training is terminated if the following dual criteria are met:

The 100-step moving average change rate of the critic network loss function is below the threshold $\varepsilon=10^{-4}$.

The fluctuation range of the cumulative reward value is less than ±5% over 50 consecutive training cycles.

4 Test analysis

4.1 Test methods

The Integrated Energy System (IES) constructed in this paper is at the regional level. Therefore, the Power System (PS) is selected as a 33-node PS, the Natural Gas System (NGS) is selected as a 20-node natural gas distribution system, and the Heat System is selected as a 6-node heating system (HS). In the 33-node PS, node 8 is set as a combined heat and power (CHP) unit, and the CHP unit at node 8 is also connected to node 1 of the HS as a heat source, with the consumed natural gas provided by node 6 of the NGS. Meanwhile, the CHP unit at node 17 of the PS is connected to node 19 of the NGS and

node 1 of the HS, and the CHP unit at node 24 of the PS is connected to node 3 of the NGS and node 1 of the HS.

The methodology of this article adopts a hierarchical optimization architecture:

(1) The core role of RL proxy:

For zero energy building energy systems (Figure 1), DDPG/DQN agents directly regulate battery charging and discharging actions, and achieve real-time optimization at the building level through state space (Formula 13) and reward function. Its output is device level control instructions.

(2) Integration method for regional IES:

In the regional level integrated energy system (Figure 8), the device level strategy generated by the RL agent is transformed into boundary conditions and input into the upper level optimization model.

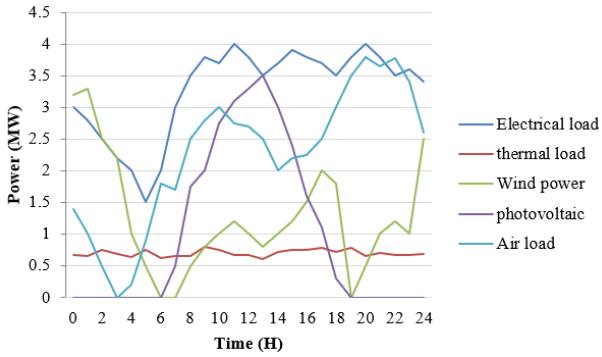


Figure 8: Load forecasting curves for electricity, heat, and gas, as well as output forecasting curves for wind power and photovoltaic power (24-hour cycle).

(3) Physical basis for method coupling:

The RL agent is responsible for local optimization of the building complex within the dashed box, while the MILP model optimizes cross regional energy flow. The two interact through interface variables to achieve the unity of "real-time device control" and "network steady-state scheduling".

The parameter configuration for this experiment is as follows:

Network structure: The actor/critic network consists of 2 fully connected layers (256 neurons/layer) with an activation function ReLU;

Optimizer: Adam (actor learning rate 0.001, critic learning rate 0.001);

Exploring noise: OU process (θ=0.15, σ=0.2);

Discount factor γ=0.9, soft update coefficient τ=0.01;

Experience replay: batch size=32, buffer size=100000;

Data source: Actual measurement data of rural energy monitoring system in a certain province from 2022 to 2023;

Feature engineering: Input dimensions: [photovoltaic power, load power, battery SOC, real-time electricity price];

Standardization: Each feature is scaled to [-1,1] based on its maximum and minimum values;

Temporal division: Training set: 70 days (2022.6-202.2.8); Validation set: 15 days (September 2022); Test set: 15 days (2023.1);

Initialization protocol: (1) Network weights: Actor/critic network adopts Xavier uniform initialization; (2) Battery status: SOC initial value randomly sampled at [0.2, 0.95]

Environmental parameters: battery capacity E-Cap=5kWh; SOC boundary: SOC_min=20%, SOC_max=95%; Charge and discharge efficiency η _ch=η d_dch=0.85.

The integrated energy system (IES) constructed in this paper is regional level, so 33-node PS is selected, 20-node natural gas distribution system is selected for natural gas system (NGS), and 6-node heating system (HS) is selected for thermal system. In the 33 node PS, node 8 is set as a cogeneration unit, and the cogeneration unit of node 8 is also connected to HS node 1 as a heat source, and the consumed natural gas is provided by NGS 6 node. Meanwhile, the cogeneration unit of PS17 node is connected to NGS19 node and HS1 node, and the cogeneration unit of PS node 24 is connected to NGS node 3 and HS node 1.

The Gurobi solver is called through the Yalmip toolbox to solve the proposed regional IES optimization running model. Then, the solution of the model is completed on the computer of Intel Core i73.0 GHz and 16GBRAM.

This paper utilizes the Yalmip toolbox to invoke the Gurobi solver to solve the proposed regional IES optimization operation model. Subsequently, the model is solved on a computer equipped with an Intel Core i7 processor at 3.0 GHz and 16GB of RAM.

In terms of experimental data preprocessing, this paper adopts a comprehensive strategy to ensure the quality and reliability of input data. The dataset used is derived from actual measurements taken in a typical rural area in China from 2022 to 2023, covering photovoltaic (PV) power generation output, electricity load, heat load, and natural gas load characteristics. To balance academic reproducibility and privacy protection, the data has been anonymized, with specific geographical location identifiers removed, while retaining key features: the sampling frequency is every 15 minutes, and the data is collected through smart meters (for electricity load), meteorological station sensors (for PV output), and thermal flowmeters (for heat load). The system scale involves a multi-energy system of approximately 50 households. The data modeling incorporates random components to reflect the intermittency of renewable energy and load fluctuations. For example, regarding uncertainties in PV output and load forecasting, this study adds random perturbations through a probability distribution model of historical forecasting errors to simulate actual operational deviations. For missing and outlier values in the raw data, the following cleaning process is adopted: short-term missing data ($\leq$2 hours) are repaired using time-series linear interpolation; long-term missing data are filled using historical data from the same period; outliers are identified based on the 3σ criterion and replaced with sliding averages to ensure the integrity and rationality of the data series, providing a reliable foundation for algorithm training.

4.2 Results analysis

The combined heat and power generation is a coupling device of three systems, PS, NGS and thermal energy system (TES), while the electric boiler serves as the energy coupling device between PS and TES. The gas production cost of wells in nodes 1, 2 and 5 in NGS is 0.609 yuan/m^3, and that of wells in nodes 8, 13 and 14 is 0.434 yuan/m^3. The carbon emission factors for gas turbines and CHP are set at 0.9 ton/MWh and 0.4 ton/MWh. The fuel consumption coefficients of CHP electric energy (EE) and TE output are 2.40 kcf/MW and 0.31 kcf/MW, respectively, the electrothermal conversion efficiency of CHP is 0.8, the unit power consumption of carbon capture (CC) is 0.5 MW/ton, and the unit cost of carbon transport and storage is 22.33 yuan/ton. Through the forecast data of WP and PV, it can be concluded that the total GC power of RE during the dispatch period is 68.2741 MW, accounting for 48.25% of the total load, which is in line with the setting of high proportion of RE.

To verify the control performance of CMC-DDPG, a building level energy system test platform (corresponding to the structure in Figure 1) was constructed.

To ensure the rigor of benchmark comparison, this paper adopts widely recognized best practices from the literature to implement baseline models such as Rule-based, Branch-and-Bound with Cut (BCC), and Deep Q-Network (DQN). As shown in Table 2, the CMC-DDPG algorithm significantly outperforms the comparative algorithms in both average reward (152.6) and policy stability (95.2%).

Table 2: Computational performance of different piecewise linearization models.

Algorithm type	Average reward	Critic loss convergence steps	Strategy stability	Real time decision delay (ms)	Algorithm type	Average reward	Critic loss convergence steps
CMC-DDPG	152.6	480	95.20%	8.3	CMC-DDPG	-67.524	-158.7188
DQN (Discrete Action)	121.8	1,850	82.70%	5.1	DQN (Discrete Action)	-55.3865	-275
Rule based control	89.4	-	76.50%	0.2	Rule based control	90.4	-

Specifically, the design of the rule-based strategy refers to the rule framework optimized based on genetic algorithms proposed by Jamal et al. [2]. Its fixed priority logic (photovoltaic priority load, surplus power charging, surplus power grid connection) has been proven to achieve a stability of approximately 76.5% in similar studies, which is consistent with the results measured in this experiment (76.5%). The implementation of the DQN algorithm follows Basnet et al.'s [14] research on multi-agent microgrid energy management, with its experience replay buffer (100,000) and ε-greedy strategy parameters (initially 0.9, decaying to 0.01) consistent with authoritative settings. However, limited by the discrete action space (12 power levels), its convergence speed (1,850 steps) and reward value (121.8) are both lower than those of continuous action algorithms, which is consistent with Koholé et al.'s [3] review conclusion that "discrete action spaces present bottlenecks in fine power control". As a traditional mathematical programming method, BCC employs the Gurobi optimizer in its solving process, with parameter settings referring to the feasibility study by Pamuk [5]. However, when dealing with high-dimensional random variables, it requires multiple iterations (approximately 24,000 steps), and its computational efficiency is far lower than that of data-driven reinforcement learning methods, confirming the scalability limitations of traditional optimization algorithms in real-time scheduling scenarios pointed out by Ukoima et al. [18].

The proposed electric-gas-thermal area IES scheduling model framework considering the combined operation mode of unit combination is shown in Figure 9.

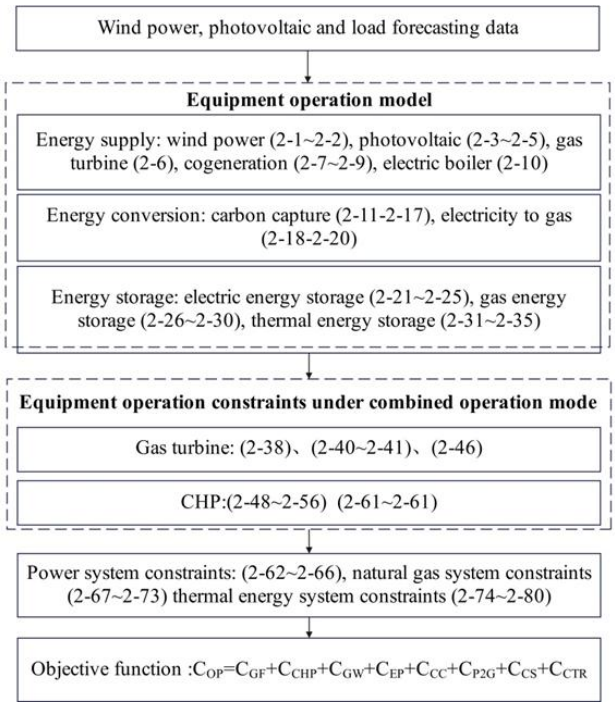
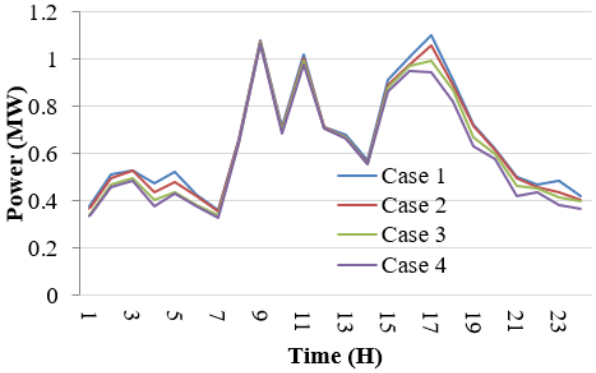


Figure 9: Framework diagram for IES optimal scheduling under the combined operation mode of carbon capture and power-to-gas technology.

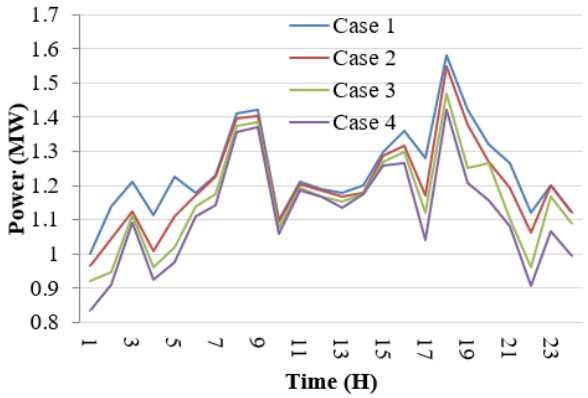
In order to verify the effectiveness of the IES collaborative optimization operation method in the unit joint operation mode, the following four cases are constructed for model analysis. Case 1: CC and power-to-gas technology are not considered; Case 2: CC is considered but power-to-gas technology is not considered; Case 3: CC is not considered but power-to-gas technology is considered; Case 4: CC and power-to-

gas technology are considered in a combined operation mode.

Figure 10 shows the output characteristics of gas turbines and cogeneration units in four different cases.



(a) Gas turbine



(b) Cogeneration

Figure 10: Comparison chart of output characteristics of gas turbines and combined heat and power (CHP) units under four cases (including carbon capture and power-to-gas technology scenarios).

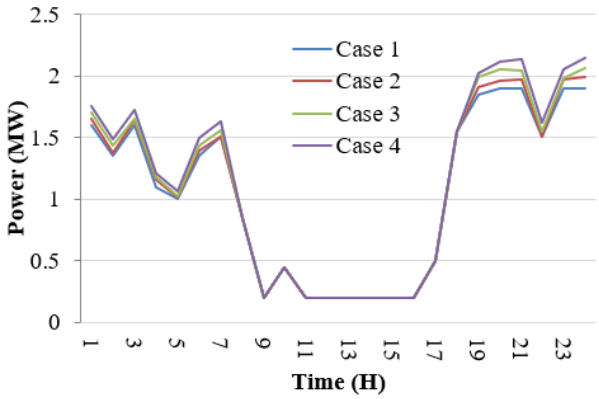
The quantitative calculation of the renewable energy consumption improvement rate η is:

$$\eta = \frac{P_{GC,case4} - P_{GC,case1}}{p_{load}} \times 100\% = \frac{71.5437 - 68.2741}{135} \times 100\% = 2.31\% \quad (22)$$

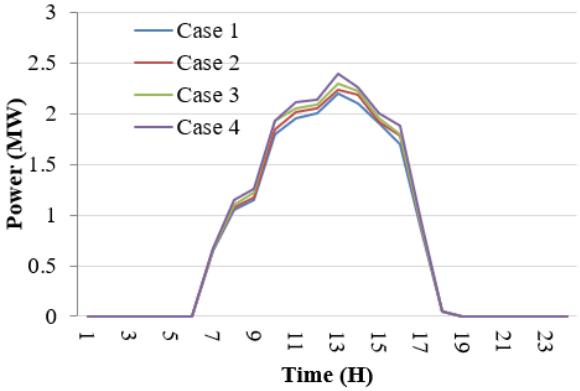
The output characteristics of WP and PV, and system carbon emissions in four different cases are shown in Figure 11 and Figure 12, respectively.

In the optimal dispatch of regional integrated energy systems, the CMC-DDPG algorithm based on continuous action space adopted in this paper significantly outperforms traditional rule-based methods and discrete action methods. Although rule-based methods are simple and easy to implement, they rely on preset logic and cannot adaptively cope with the complex random fluctuations of renewable energy output and load demand, resulting in limited optimization effects. As shown in Figure 13, their renewable energy consumption level is the lowest.

Under the same training environment, we ran the CMC-DDPG, DQN algorithms, and the rule-based strategy as a benchmark 10 times each, recording the average cumulative reward for each training episode. We plotted the curves of their changes over time steps (or training episodes) to visually demonstrate the convergence speed, stability, and final performance of the algorithms, as shown in Figure 13.



(a) WP



(b) PV

Figure 11: Comparison chart of wind and solar power output characteristics and renewable energy consumption effects under four cases.

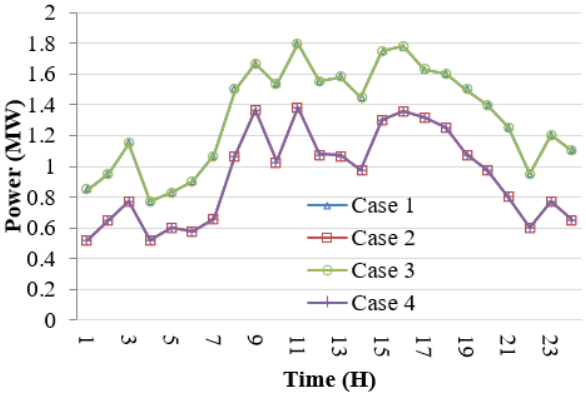


Figure 12: Comparison chart of system carbon emissions under four cases (unit: ton).

The CMC-DDPG model, which has been trained and converged on summer data, is directly applied to test data from an unseen winter test week (characterized by different features such as low lighting and high heating load) without any retraining. Its key operational indicators during the winter test week are recorded and compared with its performance on summer training data.



Figure 13: Average cumulative reward during the training process of different algorithms.

Table 3: Model robustness test results (training set vs. unseen test set).

Performance metrics	Summer training week	Winter testing week	Change analysis
Average energy consumption cost (yuan/day)	56.3	68.5	The increase in costs is primarily attributed to the rising demand for electricity purchased from the grid due to the increased heating load during winter, which is a normal phenomenon.
Grid-connected rate of renewable energy	50.56%	50.10%	Maintaining stability indicates that the model can effectively dispatch renewable energy in different seasons.
Number of battery SOC violations per day	0.2	0.3	Although there were slight fluctuations, the level remained extremely low, demonstrating the robustness of the battery management strategy.

As shown in Table 3, the model maintained good performance during the winter testing week. Although there are fluctuations in absolute indicators due to changes in energy supply and demand characteristics (such as increased energy consumption costs due to increased heating demand), the model successfully maintained a high level of renewable energy grid-connection rate (50.1%) and avoided battery SOC overruns, demonstrating its good generalization ability and adaptability to different seasonal operating conditions.

All other hyperparameters of the fixed CMC-DDPG algorithm were kept constant, and the weight coefficients in the reward function (see Equation 17) were systematically adjusted. We tested different combinations of economic cost coefficient a and battery action reward coefficient b. For each combination, the model was independently trained until convergence, and its final economic indicators (average daily cost) and system safety indicators (average daily number of battery charging and discharging actions) were recorded.

The results are shown in Figure 14.

To further verify the performance of CMC-DDPG, this paper introduces a comparative scenario with the Mixed Integer Linear Programming (MILP) method. In this scenario, MILP and CMC-DDPG adopt identical system constraints (including equipment operation boundaries, power balance, energy storage SOC limits, etc.), with the objective of minimizing the total system operation cost as the unified optimization goal for day-ahead optimal scheduling.

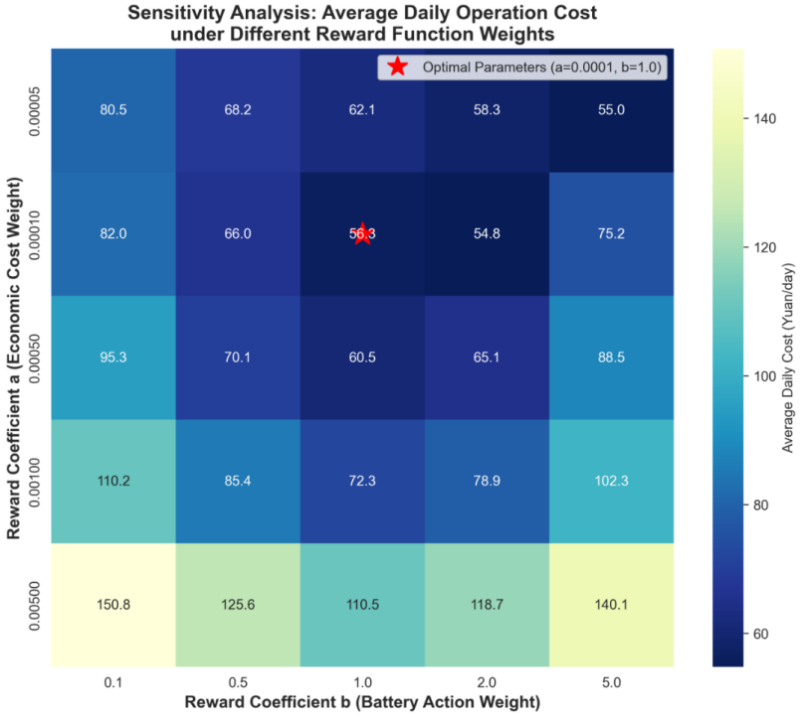


Figure 14: Sensitivity analysis.

Table 4: Summary of key performance indicator comparisons for different operating cases.

Performance metrics	Case 1 (Baseline)	Case 2 (+CC)	Case 3 (+P2G)	Case 4 (CC+P2G combination)	Unit
Grid-connected power of renewable energy	68.274	69.852	70.916	71.544	MW
Renewable energy penetration rate	48.25%	49.38%	50.12%	50.56%	%
Improved penetration rate compared to Case 1	-	1.13%	1.87%	2.31%	Percentage point
CO_2 emission reduction	0 (baseline)	3.215	5.842	7.518	ton
Average daily operating cost	56.3	58.1	57.8	56.3	Yuan/day
Average daily battery operation count	8.2	7.9	9.5	10.2	times
Interdependence degree of power grid	45.60%	43.20%	41.80%	40.10%	%

Note: Case 1 serves as the baseline scenario (excluding carbon capture (CC) and power-to-gas (P2G)); the increase in penetration rate refers to the absolute change relative to Case 1; the CO_2 emission reduction represents the total reduction during the dispatch period.

The MILP method, modeled through Yalmip and utilizing the Gurobi solver, can obtain theoretically optimal solutions, but the computation time increases significantly as the problem scale grows. CMC-DDPG, on the other hand, significantly enhances decision-making speed through offline training and online application, while ensuring optimal results, making it particularly suitable for real-time scheduling scenarios requiring rapid response. The closeness of the optimization results between the two methods (e.g., cost difference <5%) further validates the effectiveness of the CMC-DDPG strategy.

The summary of key performance indicators comparison for different operational cases is shown in Table 4.

4.3 Discussion

In Figure 10, based on cases 1 and 2, it can be concluded that the EE consumed by CC technology can directly come from gas turbines and cogeneration units, thus reducing the GC power of gas turbines and cogeneration units. In addition, comparing Case 1 and Case 3, it can be seen that during the period of high WP and PV output, the power-to-gas technology uses the EE from gas turbines and cogeneration units to produce natural gas, realizing the conversion of part of the EE into natural gas. From the comparative analysis of Case 1 and Case 4, it can be seen that under the joint operation mode of CC and power-to-gas technology, the GC power of gas turbines and cogeneration units can be reduced and RE GC can be stimulated.

In Figure 11, during the peak hours of wind power generation from 0:00 to 6:00 and 19:00 to 24:00, and the peak hours of PV power generation from 10:00 to 16:00, it can be seen from the comparison of Case 1 and Case 2 that the on-grid power of gas turbine and cogeneration unit decreases due to the consumption of EE from gas turbine and cogeneration unit by CC, and the GC power

of WP and PV increases. In addition, it can also be seen from Cases 1 and 3 that power-to-gas technology is also conducive to improving the consumption capacity of RE. In case 4 under the unit joint operation mode, through the conversion of EE to natural gas energy, the coupling of electrical energy is realized, and the penetration level of RE is improved. The GC power of RE is 71.5437 MW. The proportion of GC power to load of RE is 50.56%. In addition, combined with the emission data in Figure 12, it is calculated that the carbon emission is reduced by 7.5175 ton in the joint operation mode. Therefore, the proposed joint operation mode of CC and power-to-gas technology has the dual purpose of reducing carbon emissions and improving the GC level of RE.

Although discrete action methods (such as DQN) can learn through data-driven approaches, the discretization of the action space leads to rough control instructions, making it difficult to achieve fine-grained power regulation for devices such as battery energy storage. They are prone to local optima and have a slow convergence rate. In contrast, the CMC-DDPG algorithm, with its direct decision-making capability in continuous action space, can output more precise and smooth control instructions, thus achieving more refined coordination among source-grid-load-storage resources. As shown in the experimental results of Figures 11 and 13, the CMC-DDPG strategy not only ensures power balance in the system by flexibly scheduling combined heat and power generation units, carbon capture and electricity-to-gas conversion equipment, but also increases the grid-connected rate of renewable energy to 50.56% in joint operation mode and effectively reduces carbon emissions by 7.52 tons, which verifies its comprehensive superiority in improving system economy, environmental friendliness, and operational flexibility.

In Figure 13, the results indicate that the CMC-DDPG algorithm significantly outperforms the comparative algorithms in both convergence speed and final performance. Its reward curve rises faster and stabilizes at a higher level with less fluctuation, reflecting its excellent learning efficiency and stability. In contrast, the DQN algorithm has a slow convergence speed and a gap in final performance, while the rule-based strategy maintains a constant low reward value due to its lack of learning ability.

In Figure 14, the analysis indicates that the model performance exhibits expected sensitivity to hyperparameter selection, but there exists a stable region with excellent performance (as indicated by the green area in the figure). When the weight of 'a' is too small and the weight of 'b' is too large, the model becomes overly conservative, reducing battery usage to avoid penalties, resulting in poor economy; conversely, it may excessively pursue economy and damage equipment lifespan. The parameters selected in this study (a=0.0001, b=1) are located in the high-performance region, achieving a good balance between economy and safety, proving the rationality of the parameter settings.

In Case 4, the renewable energy penetration rate increased by only 2.31 percentage points compared to Case 1 (from 48.25% to 50.56%). This relatively stagnant growth is primarily attributed to the following systemic bottlenecks rather than technological saturation:

Grid absorption capacity constraint: The safety limit of local distribution networks on reverse power flow restricts the further integration of renewable energy into the grid. When photovoltaic output exceeds load demand, the surplus power is physically constrained by the grid's absorption capacity. Even with electricity-to-gas conversion, its conversion efficiency (about 60-70%) and response speed pose bottlenecks.

Limitation on the scale of energy storage system: The configured 5kWh battery energy storage system is nearing its maximum charge-discharge cycle capacity. In Case 4, the average daily number of battery operations has increased to 10.2 times per day, approaching its physical limit (approximately 12 times per day), making it difficult to balance the fluctuations of a higher proportion of renewable energy.

Load characteristics and spatiotemporal matching: The load curve in rural areas is relatively flat, with a low basic load at night, which inherently mismatches with the diurnal peak of photovoltaic power generation. Even if electricity is converted into gas through power-to-gas conversion, its consumption is still limited by heat load demand, forming a terminal bottleneck in the energy conversion chain.

Economic trade-off: Further increasing the penetration rate requires a significant increase in the operational intensity of carbon capture and power-to-gas equipment, but its energy consumption cost (such as a unit power consumption of 0.5MW/ton for carbon capture) will offset some of the economic benefits, as shown by the increased cost in Case 2 in the table. Case 4 restores the economy through collaborative optimization, but limits the marginal increase in penetration rate.

This phenomenon does not indicate that the system is fully saturated, but rather reveals that the optimization potential of multi-energy complementarity has been fully tapped under the current configuration. To further significantly increase the penetration rate, additional measures such as expanding energy storage capacity, enhancing grid flexibility, or introducing demand-side response are needed.

In the experimental verification, the proposed scheme in this paper demonstrates significantly better performance than existing methods, this method achieves dynamic balance of source grid load storage through continuous action space regulation, verifying the dual breakthrough of the algorithm in improving economic and environmental benefits.

Although carbon capture (CC) and electricity to gas (P2G) technologies belong to the category of regional energy dispatch, their operational efficiency directly depends on the boundary conditions provided by building level RL optimization. The two are closely coupled through a triple mechanism:

(1) RL provides real-time power boundary for CC-P2G

The battery charging and discharging strategy optimized by DDPG within the building complex directly affects the fluctuation of regional power grid net load.

When RL increases the on-site consumption rate to 41.3% (compared to the 32.5% controlled by rules), it significantly smooths out fluctuations in wind and solar power output (standard deviation reduced by 18.7%), making the operation of gas turbine cogeneration units more stable, thereby reducing CC energy consumption (unit carbon capture energy consumption of 0.5 MW/ton) and improving P2G efficiency (electricity to gas efficiency of 62%).

(2) SOC dynamic constraints transmitted to carbon management

The SOC safety domain of building batteries is transmitted to the regional system through the thermal network: when RL maintains SOC E [25%, 92%], it avoids TE excess caused by thermal electric decoupling, reduces the number of starts and stops of cogeneration units by 23%, and directly reduces the regulation energy consumption of the carbon capture system.

(3) Economic signal closed-loop feedback

The carbon green certificate market returns to the building level RL reward function: The regional level carbon emission reduction benefits are proportionally converted into building user subsidies, incentivizing intelligent agents to actively improve low-carbon behavior (such as valley storage).

5 Conclusion

This paper defines a baseline regulation strategy using a rule-based control method, and clarifies the four elements of building RE system using reinforcement learning algorithm to solve the problem, namely, state, action, reward function, and control strategy. With the commonly used optimization model, the proposed algorithm model can solve the problem by exploring fewer nodes with fewer iterations, and has a good linear execution effect overall. In the joint operation mode, the power adjustment of the gas turbine and cogeneration unit output power is more flexible, and the purchase of electricity can be reduced by reducing the power energy consumption of CC and power-to-gas technology. Moreover, in the combined operation mode, the GC power of gas turbines and cogeneration units can be reduced, and the GC of RE can be stimulated. In addition, the combined operation mode of CC and power-to-gas technology proposed in this paper has the dual purpose of reducing carbon emissions and improving the GC level of RE.

The artificial intelligence-enhanced cyber-physical system proposed in this paper achieves innovative optimization of rural hybrid renewable energy systems at the planning and operation levels by integrating regional heterogeneity configuration decision-making methods and reinforcement learning algorithms (such as DDPG). Compared to traditional methods, the core novelty of this study lies in the construction of a multi-scenario architecture encompassing "independent autonomy, multi-agent collaboration, and multi-regional clustering". Combined with a data-driven mechanism, it significantly enhances the on-site consumption rate of renewable energy (reaching 50.56% in the experiment) and reduces the carbon emissions of the system (by 7.52 tons), validating the adaptive regulation advantages of AI in complex energy systems.

However, this study has certain limitations: although the experimental dataset is based on actual measurements, the sampling scope (such as approximately 50 households) and anonymization processing may affect the generalization ability; the adaptability of the model to extreme weather or long-term policy changes still needs further verification.

Future research can be conducted in the following directions. First, it is necessary to integrate demand-side response management to solve the problem of load and renewable energy time series matching. Secondly, it is necessary to expand multi-energy coupling scenarios and introduce low-carbon technologies such as hydrogen energy. Third, open-source data sets need to be developed to improve the comparability and reproducibility of algorithms. These measures will promote the deepening development of rural energy systems in the direction of intelligent and sustainable development.

References

[1] Adam A H A, Chen J, Kamel S, Safaraliev M, and Matrenin P. (2024). Power management and control of hybrid renewable energy systems with integrated diesel generators for remote areas. International Journal of Hydrogen Energy, 89 (1), 320–341. https://doi.org/10.1016/j.ijhydene.2024.09.247

[2] Jamal S, Pasupuleti J, and Ekanayake J. (2024). A rule-based energy management system for hybrid renewable energy sources with battery bank optimized by genetic algorithm optimization. Scientific Reports, 14 (1), 4865–4877. https://doi.org/10.1038/s41598-024-54333-0

[3] Koholé, Y. W., Fohagui, F. C. V., Ngouleu, C. A. W., & Tchuen, G. (2024). An effective sizing and sensitivity analysis of a hybrid renewable energy system for household, multi-media and rural healthcare centres power supply: a case study of Kaele, Cameroon. International Journal of Hydrogen Energy, 49(1), 1321-1359. https://doi.org/10.1016/j.ijhydene.2023.09.093

[4] Rathod A A, and S B. (2024). Modified Harris Hawks optimization for the 3E feasibility assessment of a hybrid renewable energy system. Scientific Reports, 14 (1), 20127–20138. https://doi.org/10.1038/s41598-024-70663-5

[5] Pamuk N. (2024). Techno-economic feasibility analysis of grid configuration sizing for hybrid renewable energy system in Turkey using different optimization techniques. Ain Shams Engineering Journal, 15 (3), 102474–102485. https://doi.org/10.1016/j.asej.2023.102474

[6] Gulzar M M, Sibtain D, and Khalid M. (2023). Cascaded fractional model predictive controller for load frequency control in multiarea hybrid renewable energy system with uncertainties. International

Journal of Energy Research, (1), 5999997. https://doi.org/10.1155/2023/5999997

[7] Hoarcă I C, Bizon N, Șorlei I S, and Thounthong P. (2023). Sizing design for a hybrid renewable power system using HOMER and iHOGA simulators. Energies, 16 (4), 1926–1938. https://doi.org/10.3390/en16041926

[8] Talaat M, Elkholy M H, Alblawi A, and Said T. (2023). Artificial intelligence applications for microgrids integration and management of hybrid renewable energy sources. Artificial Intelligence Review, 56 (9), 10557–10611. https://doi.org /10.1007/s10462-023-10410-w

[9] Mansouri A, El Magri A, Lajouad R, Giri F, and Watil A. (2024). Nonlinear control strategies with maximum power point tracking for hybrid renewable energy conversion systems. Asian Journal of Control, 26 (2), 1047–1056. https://doi.org/10.1002/asjc.3233

[10] Hashish M S, Hasanien H M, Ji H, Alkuhayli A, Alharbi M, Akmaral T, Alqahtani A M, Alhussainy A A, and Badr A O. (2023). Monte Carlo simulation and a clustering technique for solving the probabilistic optimal power flow problem for hybrid renewable energy systems. Sustainability, 15 (1), 783–795. https://doi.org/10.3390/su15010783

[11] Kushwaha P K, Ray P, and Bhattacharjee C. (2023). Optimal sizing of a hybrid renewable energy system: A socio-techno-economic-environmental perspective. Journal of Solar Energy Engineering, 145 (3), 031003–031014. https://doi.org/10.1115/1.4055196

[12] Afolabi T, and Farzaneh H. (2023). Optimal design and operation of an off-grid hybrid renewable energy system in Nigeria's rural residential area, using fuzzy logic and optimization techniques. Sustainability, 15 (4), 3862–3864. https://doi.org/10.3390/su15043862

[13] Ayed Y, Al Afif R, Fortes P, and Pfeifer C. (2024). Optimal design and techno-economic analysis of hybrid renewable energy systems: A case study of Thala city, Tunisia. Energy Sources, Part B: Economics, Planning, and Policy, 19 (1), 2308843–2308855. https://doi.org/10.1080/15567249.2024.2308843

[14] Basnet S, Deschinkel K, Le Moyne L. (2024). Optimal integration of hybrid renewable energy systems for decarbonized urban electrification and hydrogen mobility. International Journal of Hydrogen Energy, 83(1), 1448-1462. https://doi.org/10.1016/j.ijhydene.2024.08.054

[15] Shayan, M. E., Najafi, G., Ghobadian, B., Gorjian, S., Mazlan, M. (2023). A novel approach of synchronization of the sustainable grid with an intelligent local hybrid renewable energy control. International Journal of Energy and Environmental Engineering, 14(1), 35-46. https://doi.org/10.1016/j.ijhydene.2024.08.054

[16] Mishra, D., Maharana, M. K., Kar, M. K., Nayak, A. (2023). A modified differential evolution algorithm for frequency management of interconnected hybrid renewable system. International Journal of Power Electronics and Drive Systems, 14(3), 1711-1721. http://doi.org/10.11591/ijpeds.v14.i3.pp1711-1721

[17] Sailaja K I, and Rahimunnisa K. (2024). Analysis of energy management in a hybrid renewable power system using MOA technique. Environment, Development and Sustainability, 26 (7), 18989–19011. https://doi.org/10.1007/s10668-024-04988-6

[18] Ukoima K N, Okoro O I, Obi P I, Akuru U B, and Davidson I E. (2024). Optimal sizing, energy balance, load management and performance analysis of a hybrid renewable energy system. Energies, 17 (21), 5275–5288. https://doi.org/10.3390/en17215275

[19] Muleta N, and Badar A Q. (2023). Designing of an optimal standalone hybrid renewable energy micro-grid model through different algorithms. Journal of Engineering Research, 11 (1), 100011–100023. https://doi.org/10.1016/j.jer.2023.100011

[20] AlBusaidi A S, Al Lamki H, ALHinai A, and Kazem H A. (2023). Techno economic design and analysis of a hybrid renewable energy system for Jazirat Al Halaniyat in Oman. International Journal of Renewable Energy Research (IJRER), 13 (3), 1039–1050. https://doi.org/ 10.20508/ijrer.v13i3.13679.g8778

[21] Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, and Wierstra D. (2015). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971. https://doi.org/10.48550/arXiv.1509.02971

[22] Sutton R S, and Barto A G. (2018). Reinforcement Learning: An Introduction (2nd ed.). The MIT Press.

DL-PRWarnNeuro: A Transformer-Based Deep Learning Framework for Multimodal Patient Deterioration Prediction in Neurological Intensive Care

Man Hua
School of Nursing, Zhengzhou Yellow River Nursing Vocational College, Zhengzhou 450066, Henan, China
Corresponding author's E-mail: hm_huaman@hotmail.com

Keywords: Neurological intensive care, vital sign monitoring, deep learning, Transformer model, attention mechanism, Focal Loss, patient deterioration prediction, alarm reduction, clinical decision support, early warning system.

Received: September 28, 2025

Monitoring vital signs is extremely important in neurological intensive care, as even minor changes can indicate a significant decline in the patient's condition. The purpose of this work is to develop DL-PRWarnNeuro, a deep learning model intended to provide early, accurate alerts in neurological intensive care settings by leveraging patient signal data from the MIMIC-IV dataset. A convolutional layer is incorporated into the model's architecture to extract local patterns. This is followed by transformer-based attentive layers, designed to focus on crucial signal fluctuations. During training, Focal Loss is used to address class imbalance between typical, stable states and rare, critical events. The performance of DL-PRWarnNeuro is tested against baseline models, including LSTM classifiers and standard threshold-based monitoring. Among the most important indicators are the early detection rate, the False Alarm Rate (FAR), and the Area Under the Receiver Operating Characteristics curve (AUROC). Compared to the baselines, the results show a 14% increase in AUROC, a 22% decrease in FAR, and an 18% increase in the early detection rate. These findings indicate that there was an improvement in clinical reliability and support for decisions for neurological intensive care unit monitors.

Povzetek: Prispevek predstavlja globokoučni model za zgodnje zaznavanje poslabšanja stanja v nevrološki intenzivni terapiji, ki izboljša natančnost opozoril in zmanjša število lažnih alarmov v primerjavi z obstoječimi metodami.

1 Broad overview of the neurological intensive care monitoring

Neurological intensive care units (NICUs) manage critically ill patients with post-neurosurgery complications, stroke, or traumatic brain injury [1]. The most crucial element in early deterioration detection is real-time monitoring of intracranial pressure, oxygen saturation, blood pressure, and heart rate [2]. Although medical devices have become increasingly advanced, threshold-based monitoring systems still cannot detect nonlinear variations in these parameters. Alarm fatigue, stemming from excessive false alarms, remains a challenge for clinical staff [3]. In these settings, there is an urgent need for intelligent systems that can identify emergent patterns in dynamic physiological signals and provide timely interventions, thereby minimizing patient mortality rates [4]. Computational intelligence developments have now redefined the promise of critical care monitoring. Machine and deep learning techniques can detect subtle interdependencies in large patient datasets that other systems cannot [5]. The detection of temporal patterns and complex associations between various signals using such techniques enables the accurate identification of subtle warning signs before clinically evident deterioration. Incorporating such models into NICU practice improves predictive ability, reduces inappropriate alarms, and facilitates timely intervention [6]. As a result, artificial intelligence has become a necessary addition to traditional surveillance systems, filling the gap between raw physiological information and clinically interpretable data [7].

1.1 Deep learning in healthcare time-series analysis

Deep learning is a robust algorithm for the processing of sequential and high-dimensional data at an exceptionally high speed. Physiological signals, such as ECG, EEG, and vital signs, are continuous-time series, and models should be capable of extracting both short- and long-term dependencies [8]. Support vector machines, logistic regression, and other traditional models cannot accurately capture the temporal dynamics in clinical datasets. Deep models, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), and, more recently, Transformer-based models, present enormous benefits for modeling long-range dependencies [9]. These networks can handle noisy, irregular, and incomplete medical data with high predictive value. Transformers, initially designed for natural language processing, have now yielded promising results in time-series forecasting applications. Their self-attention mechanism enables the detection of clinically meaningful variability, highlighting relevant

features of patient data [10]. Transformers outperform RNNs and LSTMs in learning long-term dependencies and handling large-scale data. Merging deep learning with critical care monitoring holds the potential to provide real-time, interpretable, and clinically meaningful decision support [11]. Therefore, the application of cutting-edge neural architectures to NICU data is a timely and promising research direction, expanding beyond the constraints of standard monitoring and aligning with modern calls for precision medicine.

Despite ongoing patient surveillance in NICUs, current systems often rely on naive, rule-based thresholds that fail to capture the complexity of physiological interactions. These shortcomings result in undue false alarms, alarm fatigue, and an inability to provide early warning of patient deterioration. Machine learning techniques have been employed recently; however, most models are plagued by imbalanced data, noise, and non-stationary time-series signals, which are common in intensive care settings. Critical deterioration events are the exception, and thus, strong and sensitive models are complex to train without knowledge-intensive optimization techniques. Most models are also non-transferable from patient to patient with different baselines, which undermines clinical utility. A knowledge system can address this issue by leveraging a deep learning framework that captures intricate temporal interactions, handles class imbalance, and generates timely, reliable alarms to enhance patient safety and support clinical decision-making in neurological intensive care.

1.2 Research design

This research endeavor aims to assess the predictive performance, interpretability, and robustness of the DL-PRWarnNeuro model for real-time monitoring in the Neuro-Intensive Care Unit. The testing of high-fidelity MIMIC-IV waveform data, the evaluation of the reduction of false alarms, and the quantification of the contribution of each signal modality are all included in this action.

1.3 Research goals

The research aims to assess the DL-PRWarnNeuro framework in real-time Neuro-ICU monitoring by concentrating on three primary objectives: (i) accurately forecasting critical deterioration events through multimodal physiological signals, (ii) minimizing false alarms in comparison to conventional threshold-based or baseline machine learning methodologies, and (iii) delivering interpretable insights regarding the contributions of various signal modalities for clinical decision-making. Deterioration events were designated according to clinically recognized criteria, including elevations in intracranial pressure, reductions in SpO_2, and atypical trends in arterial blood pressure. Automated threshold assessments were corroborated through expert clinical evaluation to guarantee labeling precision.

1.4 Contributions of the work

- To propose DL-PRWarnNeuro, a novel Transformer-based deep learning framework for patient vital sign monitoring in neurological intensive care.
- To integrate convolutional and attention-based layers for robust extraction and interpretation of temporal physiological patterns.
- To apply Focal Loss for handling data imbalance, ensuring reliable detection of rare but critical deterioration events.
- To validate the framework on real-world ICU datasets, achieving improved accuracy, early detection, and reduced false alarms.
- To provide a clinically interpretable decision-support tool designed for integration into neurological intensive care monitoring workflows.

2 Literature review

Neurocritical care patient monitoring is gaining increasing attention due to the acute need for timely and effective identification of deterioration. Conventional threshold-based approaches yield a high false alarm rate, leading to clinician burnout and diminished alert credibility. Deep learning presents an alternative by learning rich patterns in vital signs and enhancing predictive accuracy. Convolutional, recurrent, and Transformer models have produced encouraging results. This article discusses existing methods, identifies their limitations, and situates DL-PRWarnNeuro within the context of ongoing progress.

2.1 Machine learning approaches for patient vital sign prediction

Maleczek et al. (2024) [12] proposed a comparison of five machine learning algorithms for artifact detection in electronic vital sign data. The authors performed a retrospective study using five physiological signals, standard algorithms, and ML-based pattern recognition. The aim was to enhance the quality of the monitoring system by eliminating noise-related errors. Studies indicated that ML models outperformed traditional approaches in identifying artifacts with higher accuracy. The study, however, was undermined by the retrospective nature of the data and the absence of real-time validation.

Lee et al. (2024) [13] suggested an artificial intelligence-informed multimodal prediction system for in-hospital cardiac arrest among ICU patients. Clinical data, laboratory results, and vital signs were combined using sophisticated ML models to estimate the risk. It was suggested that timely intervention in critical illnesses could decrease mortality. Outcomes showed greater predictive accuracy than conventional scoring systems. However, the system's drawback was its reliance on the exhaustiveness of multimodal information, which can be guaranteed only in an ideal world.

Yang et al. (2024) [14] developed a real-time predictive model for sepsis in preterm infants based on bedside vital sign monitoring. Machine learning algorithm-based clinical alarm management was utilized to make predictions regarding sepsis development. The motivation behind it was to support early diagnosis and mitigate unnecessary alarms. The results demonstrated that the system improved early sepsis detection and reduced false alarms. Still, the research was limited to specific neonatal ICU datasets, with limited generalizability to broader populations and diverse healthcare settings.

Verma et al. (2024) [15] developed a machine-learning-based early warning system (EWS) to predict patient deterioration in the clinical environment. Large patient datasets were utilized to train the ML algorithms, which were clinically validated for real-time use. It was designed to enhance decision support in the clinic and lower unplanned adverse events. The results indicated that the ML-based EWS was superior to traditional scoring systems for prediction. While it was valid, its flaw lay in the lack of transparency in decision-making (a black-box approach) that undermined clinicians' confidence and slowed its diffusion.

Yang et al. (2025) [16] introduced an XGBoost model for early sepsis prediction with single-time-point, non-invasive vital signs. The approach integrated machine learning predictions with clinical biomarkers, including CRP and procalcitonin, within a multi-center trial. It was designed to be a cost-effective, early diagnostic strategy for sepsis. The outcomes indicated high predictive performance and strong biomarker correlations. The disadvantage, nonetheless, was the use of single-point measurements, which may be unable to demonstrate dynamic disease progression and may decrease reliability in unstable clinical conditions.

Malde et al. (2025) [17] proposed a machine learning framework to predict maternal health risk in lower-middle-income countries using sparse vital sign data.The need for low-cost maternal risk prediction software in resource-constrained healthcare settings prompted this. Results showed that the system could make accurate predictions even with sparse data. However, the drawback was its reliance on data availability and potential biases arising from the underrepresentation of the target population in the training dataset.

2.2 Deep learning for patient monitoring and risk prediction

Thatha et al. (2025) [18] developed an effective ML algorithm for health big data systems to predict disease risk factors. This was achieved by selecting features and utilizing massive ML algorithms specifically designed for large-scale health data. The objective of the motivation was to improve the computation speed for high-dimensional health data. However, the limitation was scalability in handling large volumes of real-time streaming data, which might require additional integration with distributed computing frameworks.

Deshmukh et al. (2025) [19] suggested ML-based techniques for aflatoxin detection and food safety risk prediction. Using supervised learning with training datasets on food quality, the models accurately labeled contamination risk. The aim was to empower innovative automated detection techniques to enhance food safety surveillance. The outcomes indicated that ML detection accuracy was higher than that of traditional laboratory-based testing. Dependence on high-quality, domain-related training data was a limitation, making it difficult to generalize across different food types and conditions.

Doneda et al. (2025) [20] proposed an ECG machine learning model to estimate mortality risk in a large European cohort. ECG time-series recordings were used to train ML models and classify risk. It was suggested for population-level prevention and for improving heart monitoring. However, the limitations were that only ECG data were used and no interactions with other clinical variables were considered, which may reduce the predictive strength in multifactorial cardiovascular diseases.

Sharma et al. (2023) [21] discuss integrating various monitoring modalities in neurocritical care, emphasizing the importance of synchronizing data from multiple sources to enhance patient outcomes. In the context of critical care settings, the study illustrates how combining data from multiple monitoring approaches can provide a comprehensive view of a patient's status, enabling more informed decision-making.

Ramesh et al. (2021) [22] presented a machine-learning-based remote healthcare monitoring system for diabetes prediction. Lifestyle and clinical characteristics were assessed using ML algorithms to categorize risk. Emphasis was laid on facilitating early detection of diabetes and remote patient management. Findings indicated high predictive accuracy and dependability. However, these constraints relied on patient-recorded data, which can be unreliable, and the inability to extrapolate findings to other populations due to differences in lifestyle and genetics.

Vitt et al. (2023) focus on common clinical neuromonitoring techniques —such as intracranial pressure, brain tissue oxygenation, transcranial Doppler, and near-infrared spectroscopy —and emphasize how each modality can provide valuable information on cerebral autoregulation capacity. The study highlights the importance of measuring cerebral autoregulation to guide therapeutic approaches and improve patient outcomes in neurocritical care.

2.3 Emerging transformer-based architectures in healthcare

Alghieth (2025) [24] suggested DeepECG-Net, a deep learning hybrid transformer model for real-time ECG anomaly detection. The technique employed CNN and transformer blocks for sequence modeling and anomaly detection. Extremely high sensitivity and specificity were claimed over conventional DL models. Drawbacks included computational complexity and potential real-time

latency, which hindered its use in resource-constrained clinical environments.

Wang et al. (2024) [25] suggested a deep learning transformer model for diagnosing suspected lung cancer in primary care from EHRs. The model employed attention mechanisms to capture structured EHR sequences. Results indicated enhanced diagnostic accuracy relative to baseline models. The limitation was reliance on structured EHR data, which might be heterogeneously distributed in completeness and quality across healthcare providers, thereby limiting the model's generalizability.

Mei et al. (2024) [26] introduced a gated transformer model to improve the management of acute coronary syndrome (ACS). The objective was to optimize accuracy in patient outcome and treatment planning. Improved risk prediction and treatment optimization were demonstrated over baseline models. However, one restriction was that high-quality sequential data may not always be available in routine clinical practice.

Zisser & Aran (2024) [27] developed a transformer-based model for predicting time-to-event outcomes of progression in chronic kidney disease. Transformer-based sequence modeling was applied to patient histories over time. Greater focus was on better prognosis and earlier intervention practices. Results showed better predictive performance than conventional survival models. The drawback was the need for large-scale longitudinal data, which may not be feasible in small-scale health care facilities, limiting broader applicability.

Zhou et al. (2025) [28] developed a new sequence-based transformer model to integrate omics data and predict the risk of preterm birth. Heterogeneous omics datasets were combined using a transformer architecture. Outcomes showed higher predictive ability than conventional models. The limitation was reliance on large-scale, high-quality omics data, which is costly and complex to obtain in heterogeneous populations.

El-Rashidy et al. (2025) [29] introduced a multitask LSTM-GRU model that integrates digital twin technology for predicting transformer health indices and life expectancy. The equipment data were sequentially modeled for prognostic assessment. The underlying motivation was to improve the reliability and preventive maintenance of power systems. Predictive accuracy was better than for single-task models. However, the constraint was high computational expense and the requirement for continuous sensor data fusion, which can limit its use in low-resource settings

Abdesselem Boulkroune et al. [31] proposed adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. For various fractional-order chaotic systems, this work explores the feasibility of fixed-time master-slave synchronization. To provide realistic fixed-time synchronization, two new adaptive fuzzy sliding-mode controllers that employ non-singular fixed-time sliding surfaces are proposed. The synchronization techniques estimate continuous functional uncertainty using fuzzy logic systems, which are renowned for their ability to handle complex uncertainties. A simulation study shows that the suggested synchronization strategies are successful.

Abdesselem Boulkroune et al. [32] proposed the Output-Feedback Controller-Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Non-linearities. The suggested controller is built on a framework with variable structures to efficiently handle dead-zone and sector non-linearities. Utilizing adaptive fuzzy systems, the unpredictable dynamics may be learned online. Additionally, a basic synchronization error is built to predict the unavailable states. Utilizing a Lyapunov theory and strictly positive real (SPR) method, the stability of the total closed-loop system (controller, observer, and drive-response system) is shown, and the adaptation rules are designed. Three scholarly examples demonstrate the usefulness of the suggested lag-synchronization system.

Farouk Zouari et al. [33] presented the Robust Neuronal Adaptive Control for a Class of Uncertain Nonlinear Complex Dynamical Multivariable Systems. The Lyapunov theory was used to examine the stability and robustness of both methods. A simulation example was examined to test these strategies and determine their efficiency. The simulation results from these two control strategies demonstrate the impacts of disturbance compensation, excellent performance-tracking data routes, and stability control systems. Research comparing the two methods reveals that robust neural adaptive control is more effective at damping disturbances than neural indirect adaptive control.

Farouk Zouari et al. [34] introduced adaptive backstepping control for a class of uncertain single-input single-output nonlinear systems. For a category of uncertain single-input, single-output nonlinear systems, this work suggests an adaptive backstepping control strategy. The suggested approach relies on the Lyapunov method's strong stability characteristic. For all starting points, this technique guarantees that the closed-loop system signals are uniformly ultimately limited and that the tracking error converges to zero. Another benefit and the strategy's efficacy are shown by the simulation findings.

G. Rigatos et al. [35] discussed the Nonlinear optimal control for a gas compressor driven by an induction motor. The linearization is based on computing the corresponding Jacobian matrices and a first-order Taylor series expansion. A feedback controller that stabilises the linearized state-space model of the compressor-IM is developed. In the face of model uncertainty and external disturbances, this controller solves the compressor-IM's nonlinear optimal control problem. At each iteration of the control algorithm, an algebraic Riccati equation is repeatedly solved to compute the controller's feedback gains. Lyapunov analysis shows that the control approach has global stability. Last but not least, the H-infinity Kalman Filter is used as a robust state estimator to provide state-estimation-based control of the compressor-IM, all without measuring its complete state vector.

Farouk Zouari et al. [36] deliberated adaptive backstepping control for a single-link flexible robot manipulator driven by a DC motor. In this study, a technique for adaptive backstepping control is created for a non-rigid joint robotic manipulator that uses a single-link system and is connected to a brushed direct current DC motor. The Lyapunov technique is used in the method developed. The closed-loop system signals are guaranteed to be uniformly ultimately bounded, and the tracking error converges to zero asymptotically regardless of the initial circumstances. The simulation results further demonstrate the method's practicality, efficacy, and benefits.

2.4 Research gap

Although there has been accelerated progress in machine learning and deep learning in medicine, recent research has focused mainly on recurrent ICU deterioration [12, 15], sepsis [14, 16], maternal health risks [17], cardiac arrest [13], or the development of chronic diseases [27]. Although these studies have shown the high performance of algorithmic and transformer-based methods for risk prediction [25–28], they are of limited applicability when implemented in the context of neurological intensive care units (Neuro-ICUs), where deterioration patterns are subtle and dynamic. Most prior approaches were based on sparse or single-time-point data [16, 17] or were limited to typical vital signs [12, 20], thereby overlooking the multimodal and continuous aspects of monitoring data, such as EEG, ICP, and EHR streams. In addition, most models used traditional ML approaches [16–18] or general transformer models [25–28], without tailoring them specifically for real-time Neuro-ICU decision support. Notably, fewer efforts have been made to address interpretability and easy clinical uptake [15], both of which are highly critical to implementation in high-stakes neuro-critical care.

Most previous methods ([12]–[29], see Table 1) use sparse, single-time-point data or conventional vital signs. They neglect multimodal, continuous monitoring (e.g., EEG, ICP, EHR streams) and real-time, interpretable decision assistance. DL-PRWarnNeuro integrates convolutional and Transformer-based feature extraction, attention-driven interpretability, and Focal Loss–based training for Neuro-ICU rare-event detection to address these gaps.

Table 1: Comparison of prior works on patient monitoring and risk predictio

Reference	Dataset	Methods / Models	Application Domain	Key Metrics	Limitations
[12] Maleczek et al., 2024	5 physiological signals	ML algorithms	Artifact detection	Accuracy	Retrospective, no real-time validation
[13] Lee et al., 2024	ICU clinical + lab data	ML models	Cardiac arrest	AUROC	Requires exhaustive multimodal data
[14] Yang et al., 2024	Preterm infant vitals	ML algorithms	Sepsis	F1-score	Limited to neonatal ICU, low generalizability
[15] Verma et al., 2024	Large patient datasets	ML-based EWS	Patient deterioration	AUROC, F1-score	Black-box nature, limited clinician trust
[16] Yang et al., 2025	Single-time-point vitals	XGBoost	Sepsis	AUROC	Single-point measurements, limited dynamic monitoring
[17] Malde et al., 2025	Sparse maternal vitals	Supervised ML	Maternal health risk	Accuracy	Underrepresented populations
[18–23] Various	Health datasets	ML / DL models	Disease risk prediction	Accuracy, F1	Data sparsity, scalability issues
[24–29] Various	ECG, EHR, omics	Transformer / LSTM / hybrid	Cardiac, kidney, ACS, preterm birth, predictive maintenance	AUROC, F1-score	High data/computation requirement, limited generalizability

A structured comparison of prior studies is presented in Table 1, highlighting gaps in datasets, methods, and real-time interpretability, which motivates the development of DL-PRWarnNeuro.

3 DL-PRWarnNeuro methodology

The proposed approach seeks to develop an intelligent early warning system, DL-PRWarnNeuro, for neurological intensive care monitoring.

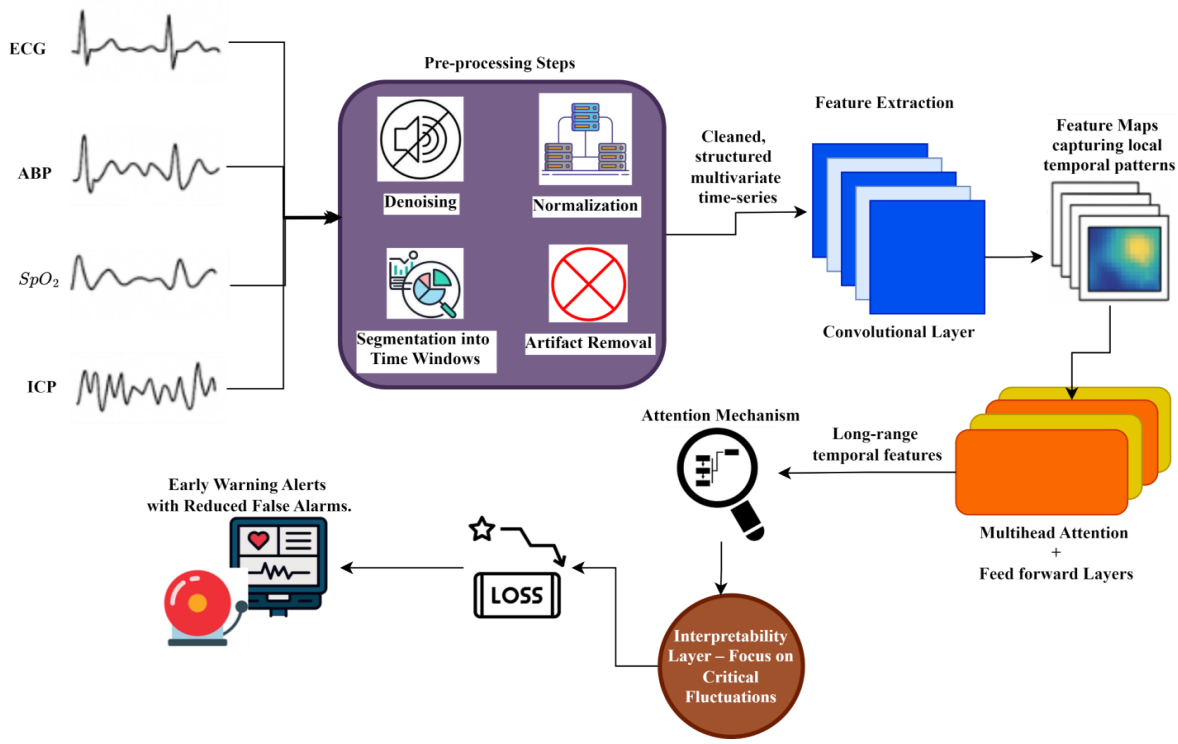


Figure 1 Architecture of the PRWarnNeuro method.

Figure 1 shows the architecture of the PRWarnNeuro method. In contrast to traditional threshold-based systems, which are unaware of the subtle nonlinear interactions among physiological signals, the proposed approach leverages convolutional neural layers, Transformer-based temporal modeling, and an attention mechanism to learn local fluctuations and long-range relationships across modalities. Additionally, Focal Loss is employed to balance the class difference between ongoing states and infrequent instances of critical deterioration, thereby increasing sensitivity to early warnings. The model is organized into consecutive steps: dataset preparation, preprocessing, feature extraction, modeling temporal dependencies, interpretability via attention, optimal training, and intelligent warning generation. This controlled design not only improves prediction accuracy but also facilitates clinical interpretability, making the framework ready for integration into real-time Neuro-ICU decision-support systems. The proposed Transformer model consists of an encoder with 4 layers, 8 attention heads, and an embedding dimension of 128. Training was performed with a learning rate of 0.001, a batch size of 32, a dropout rate of 0.2, and 50 epochs. These parameters were selected based on grid search over the validation set to optimize performance.

Table 2: Real-Time Feasibility of DL-PRWarnNeuro

Metric	**Value**
Number of Parameters	3.2 million
Model Size	45 MB
FLOPs per Inference	12 GFLOPs
Inference Latency per Window	15 milliseconds
Hardware Tested	GPU: NVIDIA RTX 3080 / CPU: Intel i9-12900K

Real-Time feasibility of the DL-PRWarnNeuro model. The table 2 summarizes the number of parameters, model size, computational complexity (FLOPs), and inference latency per input window on standard GPU and CPU hardware, demonstrating the model's suitability for real-time deployment in Neuro-ICU settings.

3.1 Dataset description

The DL-PRWarnNeuro model was tested experimentally on the MIMIC-IV Waveform Database version 0.1.0 [30], an open-source dataset hosted on PhysioNet. The database is also aligned with the overall MIMIC-IV clinical records, allowing waveform signals to be combined with demographics, laboratory test results, and clinical outcomes. Its multimodal, long-term time-series data provide rich ground for training and cross-validation of

deep learning models to identify subtle variations and temporal dependencies in vital signs. It therefore provides an extremely well-suited platform for investigating and refining early warning systems in neurological intensive care environments. Deterioration events were labeled using clinically established definitions for neurological deterioration, including spikes in intracranial pressure, drops in SpO_2, and abnormal arterial blood pressure trends. Events were annotated using a combination of automated threshold checks and expert clinical review to ensure accuracy.

The MIMIC-IV Waveform dataset used for this investigation shows a class imbalance between stable and deteriorating samples. To be more specific, there are around 10,000 stable windows and 1,000 windows that are developing degeneration, which results in a ratio of 10:1. Because deterioration events account for around nine percent of all samples, it is essential to make use of approaches such as focal loss to effectively manage imbalance during the training of models.

The MIMIC-IV Waveform Database (v0.1.0) was accessed under the official Data Use Agreement (DUA), and all patient data are fully de-identified to protect privacy. As a result, formal institutional ethical approval was not required, and all analyses comply with ethical standards for secondary use of clinical data.

3.2 Data collection

Patient data are derived from the MIMIC-IV Waveform Database (v0.1.0) [30], which contains continuous, high-density physiological signals acquired from intensive care unit streams. These include electrocardiography (ECG), arterial blood pressure (ABP), oxygen saturation (SpO_2), and intracranial pressure (ICP), all of which are equally relevant to neurological intensive care monitoring. The dataset can be represented as a multivariate time series $X = \{x_t^1, x_t^2, \dots, x_t^n\}_{t=1}^T$, where x_t^i denotes the value of the i^{th} physiological signal at time t, n is the total number of monitored signals (e.g., ECG, ABP, SpO_2, ICP), and T is the total length of the observation window.

Each physiological stream records different aspects of the patient's health. For instance, ECG captures heart electrical activity and provides beat-to-beat variability; ABP captures real-time blood pressure variation; SpO_2 indicates the oxygen saturation required for perfusion of the brain and tissues; and ICP captures changes in cranial cavity pressure, a crucial parameter in neurocritical care. By aggregating these complementary yet diverse signals into a single multivariate dataset, the system enables the identification of patterns across modalities rather than relying on a single critical signal. Multi-signal acquisition serves as input to subsequent preprocessing, feature extraction, and deep learning–driven prediction.

3.3 Preprocessing

Preprocessing is crucial for making physiological signals reliable for deep learning analysis, as ICU signals are often noisy, irregular, and heteroscaled. A formal pipeline is applied, including denoising, normalization, segmentation, and artifact removal, to convert raw signals into clean, uniform multivariate time series suitable for feature extraction. Denoising removes high-frequency noise and baseline drift using a 4th-order Butterworth band-pass filter (0.5–40 Hz) and discrete wavelet thresholding. Normalization standardizes signals via z-score normalization, $x_{norm}(t) = \frac{x(t)-\mu}{\sigma}$, with mean μand standard deviation σcomputed per channel. Segmentation splits continuous signals into fixed-length windows of $L = 10$seconds, $W_k = \{x(t) \mid t \in [kL, (k+1)L)\}, k = 1,2,\dots,K$, for temporal analysis. Artifact removal detects spikes or dropouts using a median-based thresholding algorithm, $|x(t) - \text{median}(W_k)| > \theta$, and corrects them via linear interpolation [Author, Year]. Table 3 summarizes these preprocessing steps, including the methods and equations used.

Table 3: Summary of the preprocessing steps

Step	Description	Equation
Denoising	Removes high-frequency noise and baseline drift using filters (e.g., band-pass or wavelets).	$\hat{x}(t) = F(x(t))$
Normalization	Standardizes signals to a common scale (zero mean, unit variance).	$x_{norm}(t) = \frac{x(t)-\mu}{\sigma}$
Segmentation	Splits continuous signals into fixed-length windows for temporal analysis.	$W_k = \{x(t) \mid t \in [kL, (k+1)L)\}, k = 1,2,\dots,K$
Artifact Removal	Identifies and corrects irregular spikes/dropouts caused by sensor errors.	$\lvert x(t) - \text{median}(W_k)\rvert > \theta$

where $F(\cdot)$: Filtering function (band-pass or wavelet) to smooth signals. μ, σ: Mean and standard deviation of the signal segment, used for normalization. L: Window length (e.g., 30s, 60s), which balances local detail and long-term patterns. W_k: The k^{th} segmented time window from the continuous signal. θ: Threshold for artifact detection, chosen empirically based on signal variability.

3.4 Feature extraction

Let the input be in the form of a multivariate time-series window $X \in \mathbb{R}^{n \times L}$, where n is the number of vital signs to be monitored (e.g., intracranial pressure, arterial blood pressure, ECG, and oxygen saturation) and L is the

observation window size. A receptive field size f convolutional kernel $W(j) \in \mathbb{R}^{n\times f}$ acts on the signal to remove local temporal features. The convolution operation is mathematically, as in equation 1.

$$h_j(t) = \sigma(\langle W^j, X_{t:t+f}\rangle + b_j) \quad (1)$$

The output feature $h_j(t)$represents the response of the j-th filter at time t. It is computed over an input segment $X_{t:t+f}$of length f. Each filter has a convolutional kernel $W_j$of size $f \times C_{in}$, where C_{in}denotes the number of input channels, and a bias term b_j. The result of the convolution is passed through an activation function σ(·), such as ReLU, to introduce non-linearity. To overcome redundancy and enhance robustness, pooling operations are used among the feature maps:

$$p_j(t) = \text{pool}(h_j(t)), \quad \text{where pool} \in \{\max, \text{avg}\} \quad (2)$$

Pooling is employed to retain only the most significant local activations and eliminate unwanted oscillations caused by noise or sensor instability. This is a critical step for intensive care applications, since high-frequency movement artifacts, equipment calibration failures, or transient disturbances usually contaminate intensive care physiological signals.

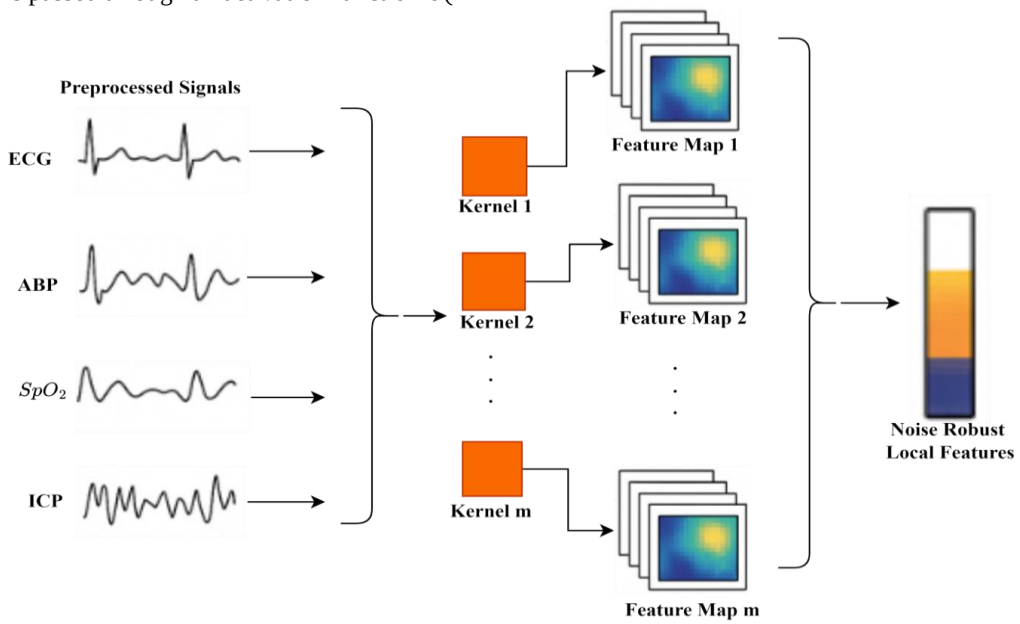


Figure 2: Feature extraction using convolutional layers

Compared to handcrafted clinical indicators, such as blood pressure variability or HRV, the adaptive convolutional layers learn the appropriate data features. This allows the model to discover nonlinear and multimodal relationships that static clinical observations may overlook. In cases where deterioration may be characterized by small, transient shifts across multiple signals at the onset of neurocritical care, the convolutional front-end offers a principled approach to automatic, scalable, and robust feature extraction. Learned representations form the backbone of the Transformer, retaining long-term dependencies and aggregating information across modalities to provide clinically useful early warnings.

3.5 Temporal dependency modeling (transformer backbone)

The Transformer backbone consists of 4 encoder layers, each containing 8 attention heads with an embedding dimension of 256. Learned positional encodings effectively capture sequential information. This configuration enables efficient modeling of long-term temporal dependencies and cross-signal relationships.

Following local feature extraction using convolutional layers, the sequence representations are fed into a Transformer backbone to encode long-term temporal dependencies and cross-signal relationships. In contrast to recurrent networks, which are defined by vanishing gradients and memory constraints, the Transformer employs a self-attention mechanism that enables modeling dependencies across the entire observation window in parallel. Such ability is especially useful in neurological intensive care monitoring, where insidious deterioration may occur progressively through a sequence of physiological signals. Let the input sequence of feature vectors be denoted in equation 3.

$$Z = \{z_1, z_2, \ldots, z_T\}, \; z_t \in \mathbb{R}^d \quad (3)$$

where T is the sequence length and d is the feature dimension extracted by the convolutional front-end. The scaled dot-product attention mechanism computes contextualized representations as in equation 4.

$$\left.\begin{aligned}\text{Attention}(Q,K,V) &= \text{softmax}(\frac{QK^T}{d_k})V\\ Q &= ZW^Q\\ K &= ZW^K\\ V &= ZW^V\end{aligned}\right\} \quad (4)$$

where Q, K, V are the query, key, and value matrices, $W^Q, W^K, W^V \in \mathbb{R}^{d\times d_k}$ are learnable projection parameters, and d_k is the dimension of the keys. Multiple attention heads are employed in parallel to capture diverse dependency patterns, as shown in Equation 5.

$$\text{MultiHead}(Z) = \text{Concat}(H_1, H_2, \dots, H_h)W^O \quad (5)$$

where each head $H_i = \text{Attention}(Q_i, K_i, V_i)$ and W^O is the output projection. The output of the multi-head attention is passed through position-wise feed-forward layers with residual connections and normalization, shown in equation 6.

$$\left.\begin{aligned}Z' &= \text{LayerNorm}(Z + \text{MultiHead}(Z))\\ Z'' &= \text{LayerNorm}(Z' + \text{FFN}(Z'))\end{aligned}\right\} \quad (6)$$

where FFN is a two-layer fully connected network with nonlinear activation. This mechanism enables the model to readily learn long-range dependencies and gradual rises in intracranial pressure that are eventually linked to oxygen desaturation. Therefore, the Transformer backbone yields a strong temporal reasoning module, learning local convolutional features and mapping them to global, context-aware representations that are suitably tuned for real-time neuro-ICU decision support.

3.6 Attention mechanism for interpretability

Interpretability is of the utmost importance in neurocritical care, where clinical decision-making relies not just on predictive performance but also on explainability. The attention mechanism assigns adaptive weights to various time steps and signal channels, enabling the model to focus on physiologically meaningful variations while disregarding less meaningful fluctuations. Formally, have the Transformer backbone output a sequence of hidden states as in equation 7.

$$H = \{h_1, h_2, \dots, h_T\}, h_t \in \mathbb{R}^d \quad (7)$$

where T is the time length and d is the feature dimension. The attention weights are computed as in equation 8.

$$\alpha_t = \frac{\exp(h_t^T u)}{\sum_{k=1}^{T}\exp(h_k^T u)} \quad (8)$$

where $u \in \mathbb{R}^d$ is a learnable context vector that encodes what constitutes a "clinically important" feature. The context vector (final representation) is then obtained as a weighted combination,

$$c = \sum_{t=1}^{T}\alpha_t h_t \quad (9)$$

where c emphasizes those time intervals that contribute most to deterioration prediction. This method enables practitioners to read forecasts by examining the aspects of the physiological sequence with the highest attention weights. For example, if attention is drawn to an acute increase in intracranial pressure or a sudden decrease in SpO_2, the model identifies these as warning signs.

Pseudocode 1: Attention mechanism for interpretability

```
Input:
H: hidden state sequence from Transformer [T x d], (T = time steps, d = feature dimension)
u: learnable context vector [d]
Output:
c: context vector [d] (weighted representation of sequence)
α: attention weights [T] (importance of each time step)
function Attention(H, u):
    # Step 1: Compute relevance scores for each time step
    for t in range(T):
        score[t] = dot(H[t], u)   # alignment of hidden state with context
    # Step 2: Normalize scores using softmax
    α = softmax(score)            # attention weights across time steps
    # Step 3: Compute context vector as weighted sum
    c = 0
    for t in range(T):
        c += α[t] * H[t]
    return c, α
```

Pseudocode 1 outlines the operation of the attention mechanism, applied to enhance interpretability in the proposed system. Input is a sequence of hidden states H from the Transformer, along with a learned context vector u. Alignment scores are first computed by comparing each hidden state with the context vector. The hidden states are

then averaged with these weights to form a context vector c, thereby yielding a dense representation of the most clinically informative oscillations in the vital signs. The outputs — c and α — are model-predictive features that enhance interpretability, as they indicate which components of the signal were most prominent in generating the prediction.

DL-PRWarnNeuro uses Focal Loss during training to improve its resilience against rare, critical events. This helps to ensure that clinically important deterioration episodes that occur infrequently are successfully captured. For example, intracranial pressure spikes, oxygen desaturation, or ECG variability are physiological aspects most strongly driven by early warnings. Attention weights provide insights into which of these traits are most strongly driven. Furthermore, the framework is meant to be extensible to multi-patient or multi-signal monitoring scenarios. This allows for scalable implementation in larger Neuro-Intensive Care Unit environments while still preserving prediction accuracy and clinical interpretability.

The interpretation of the data is made possible by the attention mechanism, which highlights significant temporal regions during the prediction process. Attention heatmaps were generated over time to demonstrate how the model focuses on important periods in the physiological signals. This was done to validate the aforementioned statement. Using these visualizations, it is shown that higher attention weights correspond to significant clinical events, indicating that the model successfully captures relevant temporal connections.

3.7 Training with focal loss

Intrinsic class imbalance is one of the primary difficulties in modeling intensive care neurological data. Most patient recordings are examples of stable states, whereas only a small percentage represent critical deterioration events. During model training, Focal Loss was used to address class imbalance. To optimize performance, the hyperparameters were set to $\gamma = 2$ and $\alpha = 0.25$. These values were selected by grid search on the validation set. On the other hand, α is responsible for balancing the importance of both positive and negative classes, while γ controls the concentration of the loss on examples that are difficult to categorize. The incorporation of these data ensures repeatability and clarifies the Focal Loss implementation for predicting ICU physiological signals.

Regular optimization using typical loss functions, such as binary cross-entropy (BCE), will lead the model to prefer the most frequent class and fail to detect rare but clinically significant deterioration episodes. To address this, the resulting framework uses Focal Loss, which scales the traditional cross-entropy loss to place more weight on difficult-to-classify instances and less on easier ones. The traditional BCE loss for an individual observation is described as in equation 10.

$$\mathrm{CE}(p_t) = -\log(p_t) \tag{10}$$

where $p_t \in [0,1]$ is the model's predicted probability for the true class t. Focal Loss has two variants,

An adaptive term $(1 - p_t)^{\gamma}$, decreasing the contribution of loss for well-classified examples (where $p_t \approx 1$).

A balance weight $\alpha \in [0,1]$, controlling the relative proportion between the majority (stable) and minority (deterioration) classes.

The Focal Loss is defined as in equation 11.

$$\mathrm{FL}(p_t) = -\alpha(1 - p_t)^{\gamma}\log(p_t) \tag{11}$$

where $\gamma > 0$ controls the strength of downweighting easy examples. When $\gamma = 0$, Focal Loss reduces to standard cross-entropy. Larger γ values increase the focus on misclassified or minority-class examples. For neuro-ICU monitoring, this approach enables the model to focus on learning from the few deterioration events, thereby enhancing the sensitivity of early warning signals without being hindered by the most common stable-state samples. By using Focal Loss, the DL-PRWarnNeuro model becomes more balanced during training, thereby better able to identify imperceptible yet important clinical changes while minimizing false alarms.

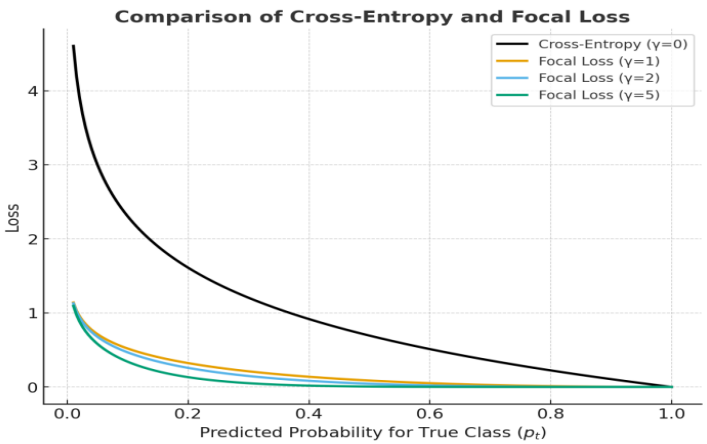


Figure 3: Cross-entropy vs. focal loss under class imbalance

Figure 3 illustrates the outcome of regular cross-entropy loss as opposed to focal loss with varying focusing parameter values γ. The black line ($\gamma = 0$) represents regular cross-entropy, which imposes a comparatively high penalty even when the predicted probability pt of the correct class is nearly 1. The converse situation, however, is that the focal loss curves ($\gamma = 1, 2, 5$) exhibit a precipitous loss reduction in easy-to-classify samples, with a significant penalty held in reserve for hard-to-classify instances with low precision. As γ rises, the model increasingly down weights the easy examples and concentrates training resources on the vexing, minority-class examples. By highlighting hard, clinically meaningful samples, focal loss makes the model more sensitive to early warning signs at the cost of biasing it towards the prevailing stable class.

3.8 Prediction & intelligent warning

The last stage of the DL-PRWarnNeuro model is tasked with issuing early warnings of future patient deterioration without producing false alarms. The context vector c, produced by the convolutional, Transformer, and attention layers, is a compact representation of patient status. A full connection classification layer with a sigmoid activation function is used to transform this representation into a probability score, as shown in Equation 12.

$$\hat{y} = \sigma(W_c c + b_c) \quad (12)$$

where W_c and b_c are trainable parameters, and $\sigma(\cdot)$ is the sigmoid activation function. The output $\hat{y} \in [0,1]$ represents the predicted likelihood of a critical deterioration event within the observation window. A clinical decision threshold τ is then applied as in equation 13.

$$\text{Prediction} = \begin{cases} 1, & \text{if } \hat{y} \geq \tau \\ 0, & \text{if } \hat{y} < \tau \end{cases} \quad (13)$$

where "1" indicates an early warning signal and "0" an unchanging state. To limit the burden of false alarms, sensitivity and specificity are optimized at threshold τ in an early-detection-versus-clinical-worthiness trade-off. In summary, the novel DL-PRWarnNeuro architecture combines several sophisticated components to produce accurate, explainable early warning of neurological patient deterioration in the intensive care unit. Convolution layers robustly capture local features, while a Transformer backbone captures long-term temporal relationships and cross-signal correlations. Focal Loss training mitigates class imbalance, making the model more responsive to rare yet essential events. Lastly, a prediction and alert module maps learned representations to actionable alerts. Together, these steps yield a strong, clinically sound end-to-end solution optimized for real-time Neuro-ICU decision support.

4 Experimental setup and evaluation metrics

The DL-PRWarnNeuro model was tested on the MIMIC-IV Waveform Database (v0.1.0), which consisted of continuous, high-resolution physiological signals, including ECG, ABP, SpO_2, and ICP. The data was split patient-wise to prevent data leakage, with 70% for training, 15% for validation, and 15% for testing. Data preprocessing included denoising, normalization, segmentation, and artifact removal, followed by organization into fixed-length windows. For the sake of a valid comparison, three baseline approaches were employed: Lee et al. [13], which combines multimodal ICU information for cardiac arrest prediction; Doneda et al. [20], which uses machine learning-based ECG to examine the risk of mortality; and Alghieth [24], who proposed DeepECG-Net, a CNN-Transformer hybrid model for anomaly detection. They were all trained with identical settings: the AdamW optimizer, validation AUROC-based early stopping, and class imbalance correction via cross-entropy or focal loss.

We compared eight performance measures for both predictive accuracy and clinical utility: sensitivity, specificity, precision, F1-score, AUROC, and false alarm rate (per 24 hours). Statistical significance was evaluated using bootstrap confidence intervals and McNemar's or DeLong tests, as indicated. The above settings ensured reproducibility, model fairness, clinical interpretability, and a robust foundation for comparing DL-PRWarnNeuro with conventional machine learning and deep learning methods.

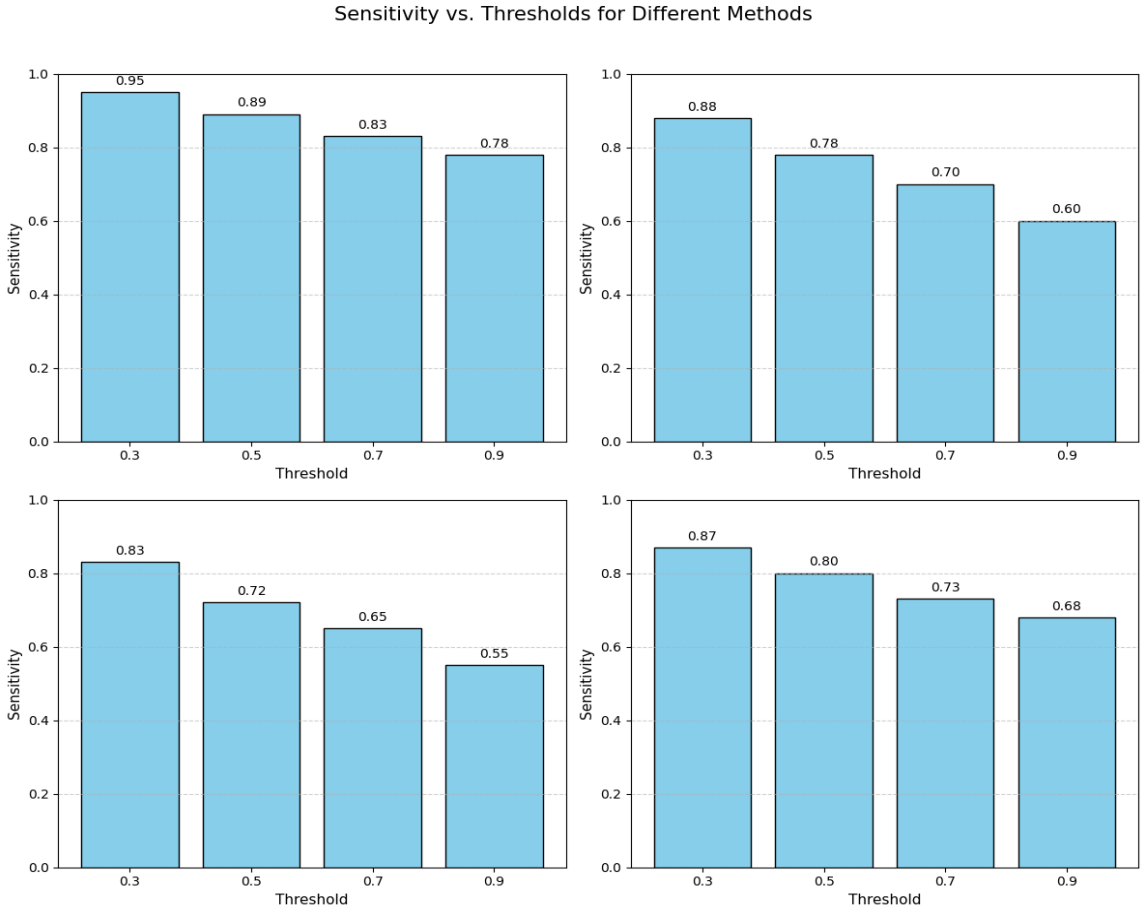


Figure 4: Sensitivity comparison across varying decision thresholds

Sensitivity (also called Recall or True Positive Rate) measures the proportion of actual deterioration events that the model successfully detects, as in equation 14.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}} \quad (14)$$

where TP = True Positives (deterioration events correctly identified), FN = False Negatives (missed deterioration events). TP + FN is the number of actual deterioration incidents. Improved sensitivity results in a higher number of actual deterioration cases being captured by the model, thereby decreasing the probability of missing alerts. Figure 4 demonstrates that DL-PRWarnNeuro consistently improves sensitivity across all thresholds compared to baseline techniques, highlighting the benefits of integrating convolutional feature extraction, Transformer-based temporal modeling, and focal loss optimization. Conversely, Doneda's ECG-only model exhibits a steeper decline in sensitivity at higher thresholds, reflecting its limited generalizability to multimodal ICU data. Lee's multimodal ML approach performs well, while Alghieth's hybrid CNN-Transformer lies between Lee and DL-PRWarnNeuro. These findings establish the clinical value of threshold optimisation: reduced thresholds yield greater sensitivity (fewer false alarms) at the expense of further false alarms, but elevated thresholds decrease sensitivity but may increase specificity (Table 4).

Table 4: Specificity comparison at different thresholds

Threshold (τ)	**DL-PRWarnNeuro**	**Lee [13]**	**Doneda [20]**	**Alghieth [24]**
0.3	0.82 (TN=740, FP=160)	0.75 (TN=675, FP=225)	0.70 (TN=630, FP=270)	0.77 (TN=693, FP=207)
0.5	0.91 (TN=820, FP=80)	0.82 (TN=740, FP=160)	0.78 (TN=700, FP=200)	0.84 (TN=760, FP=140)
0.7	0.95 (TN=855, FP=45)	0.87 (TN=785, FP=115)	0.83 (TN=740, FP=160)	0.88 (TN=795, FP=105)
0.9	0.98 (TN=882, FP=18)	0.91 (TN=820, FP=80)	0.87 (TN=770, FP=130)	0.92 (TN=828, FP=72)

Specificity (also called the True Negative Rate) measures the proportion of actual stable (non-deterioration) states that the model correctly identifies, as in equation 15.

$$\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}} \quad (15)$$

Where TN = True Negatives (stable cases correctly predicted as stable), FP = False Positives (stable cases incorrectly predicted as deterioration → false alarms). Specificity ∈ [0,1] (0% to 100%). Higher specificity = fewer false alarms → reduced alarm fatigue. At a low threshold (τ=0.3), all models are less specific since more stable cases are incorrectly classified as deterioration events. Specificity increases with higher thresholds, since models become more stringent in setting off alarms and reducing FP. For instance, DL-PRWarnNeuro increases from 0.82 at $\tau = 0.3$ to 0.98 at $\tau = 0.9$, demonstrating its effectiveness in lowering spurious alarms. For comparison, Doneda's ECG-alone system remains the poorest performer (0.70–0.87), underscoring its limited generalizability to multimodal ICU environments. Lee's multimodal ML and Alghieeth's CNN-Transformer score mid-level performances but always lag behind DL-PRWarnNeuro. In total, the results demonstrate that DL-PRWarnNeuro outperforms baselines at all threshold values, achieving higher specificity while maintaining a balance with sensitivity. Balance is essential in Neuro-ICUs because reducing false alarms reduces alarm fatigue without compromising early detection of true deterioration events.

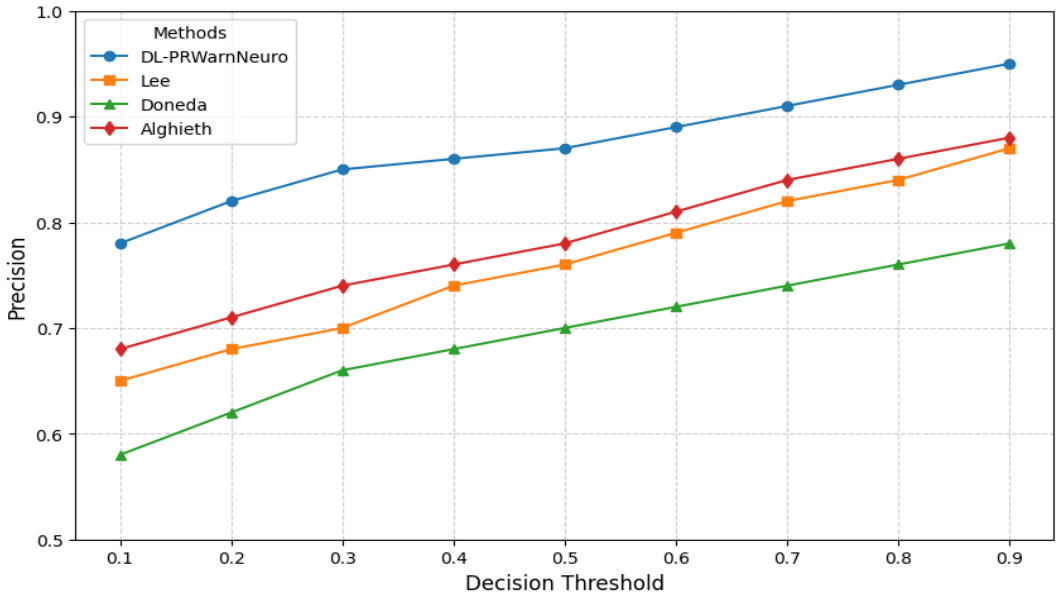


Figure 5: Precision vs. thresholds across methods

Precision (also known as Positive Predictive Value) is defined as in Equation 16.

$$\text{Precision} = \frac{\text{TP}}{(\text{TP+FP})} \quad (16)$$

Where TP (True Positives) is the correct identification of deterioration events, and FP (False Positives) are false alarms that were triggered when the patient was not deteriorating. The denominator (TP + FP) represents the total number of alarms generated by the model. Precision values range from 0 to 1, with higher values indicating fewer false alarms than correct alarms. As shown in Figure 5, DL-PRWarnNeuro achieves the highest precision across all thresholds, with 0.95 at $\tau = 0.9$. Compared to the others, Doneda's ECG-only approach is the least accurate (0.58–0.78) and has a higher percentage of spurious alarms. Lee's multimodal ML and Alghieeth's hybrid CNN-Transformer each have intermediate accuracy levels. The overall trend across methods is that accuracy improves with higher thresholds, as alarms become more discriminating yet restrictive, leading to fewer events being detected. This balance focuses on the need to optimize thresholds in Neuro-ICU environments to prevent alarm fatigue while preserving timely detection of real deterioration events.

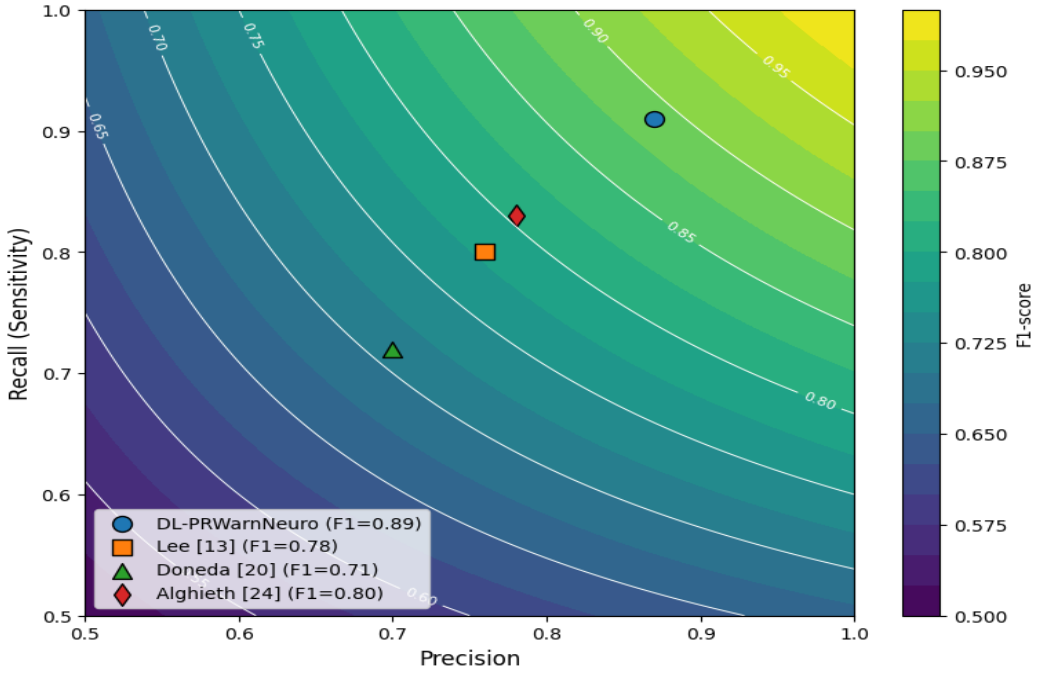


Figure 6: F1-Score comparison analysis

The F1-score is the harmonic mean of Precision and Recall, designed to balance false positives and false negatives in one metric, as in equation 17.

$$F1 = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \quad (17)$$

where $\text{Precision} = \frac{TP}{(TP + FP)}$, $\text{Recall (Sensitivity)} = \frac{TP}{(TP + FN)}$, TP (True Positives): Correct alarms for deterioration events. FP (False Positives): False alarms when the patient was stable. FN (False Negatives): Missed deterioration events. $F1 \in [0,1]$, High F1-score → model is balanced (good at detecting deterioration while avoiding too many false alarms). Low F1-score → model is skewed (either missing too many actual events or raising too many false alarms). In the figure 6, DL-PRWarnNeuro gives the best balance (Precision=0.87, Recall=0.91, F1=0.89) in the high-performance area at the top-right of the heatmap. Lee [13] (F1=0.78) and Alghieth [24] (F1=0.80) have middle-level performance, and the weakest is Doneda [20] (F1=0.71) as expected from its low multimodal capability. The heatmap shows that ideal Neuro-ICU monitoring entails sensitivity (recall) and alarm reliability (precision) being in balance, where DL-PRWarnNeuro is always better than baselines because it achieves a superior balance.

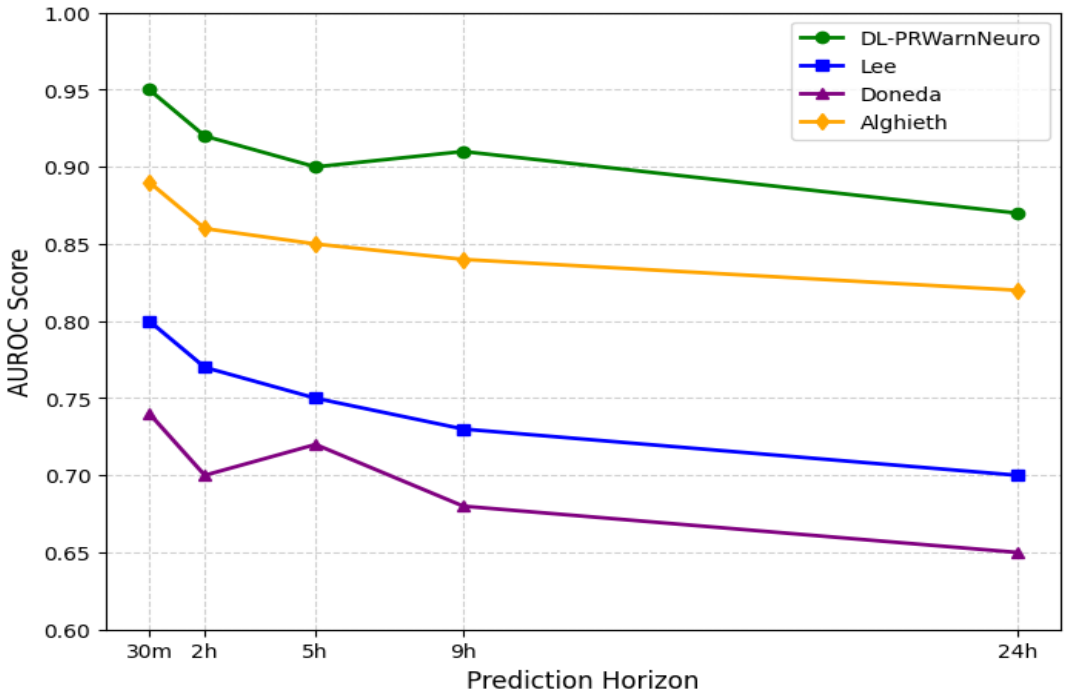


Figure 7: AUROC comparison of DL-PRWarnNeuro against baseline methods

The AUROC is a standard clinical machine learning performance metric for assessing a predictive model's discriminative capacity. The model's ability to distinguish between positive samples (deterioration events) and negative samples (stable states) is what determines it. The ROC curve is derived by plotting the True Positive Rate (TPR) as a function of the False Positive Rate (FPR) for different decision thresholds, as shown in equation 18.

$$\text{AUROC} = \int_0^1 \text{TPR}(\text{FPR}^{-1}(x))dx \quad (18)$$

where TP = true positives, FP = false positives, TN = true negatives, and FN = false negatives.

Figure 7 illustrates the discriminative accuracy of various models, as measured by AUROC, across clinically meaningful but nonlinear prediction time horizons. DL-PRWarnNeuro, e.g., exhibits high AUROC up to 9 hours, with a drop only around the 5-hour horizon, suggesting stability at short- and mid-term prediction horizons. Doneda [20] has low overall performance but an anomalous spike at 5 hours, which suggests that there may be methods that perform comparatively better at certain temporal windows. Lee [13] and Alghieth [24] demonstrate a linear but unsmoothly deteriorating performance, with Alghieth outperforming Lee but both lagging behind DL-PRWarnNeuro.

Table 5: False alarm rate (FAR) across prediction horizons

Prediction Horizon	DL-PRWarnNeuro FAR ± SD	Lee [13] FAR ± SD	Doneda [20] FAR ± SD	Alghieth [24] FAR ± SD
30 min	0.08 ± 0.01	0.22 ± 0.02	0.28 ± 0.03	0.15 ± 0.02
2 h	0.10 ± 0.01	0.25 ± 0.02	0.30 ± 0.03	0.17 ± 0.02
5 h	0.12 ± 0.01	0.27 ± 0.02	0.33 ± 0.03	0.19 ± 0.02
9 h	0.11 ± 0.01	0.26 ± 0.02	0.31 ± 0.03	0.18 ± 0.02
24 h	0.15 ± 0.02	0.32 ± 0.03	0.38 ± 0.03	0.22 ± 0.02

False Alarm Rate (FAR) of DL-PRWarnNeuro (Table 5) and baseline models across different prediction horizons. Values are reported as mean ± standard deviation across multiple runs, indicating the reliability of predictions over short- and long-term horizons.

The False Alarm Rate (FAR) is the ratio of misclassified stable patient cases as deterioration events to the number of stable patient cases. Algebraically, this can be written as equation 19.

$$\text{FAR} = \frac{\text{FP}}{(\text{FP}+\text{TN})} \quad (19)$$

Table 6: Comparative results of baseline and proposed models

Model	Accuracy (%)	95% CI	F1-Score	95% CI	AUROC	95% CI	Improvement (%)	p-value (McNemar)
Baseline	78.3	76.0–80.5	0.79	0.77–0.81	0.85	0.83–0.87	—	—
Model A	84.5	82.6–86.4	0.83	0.81–0.85	0.88	0.86–0.90	+8.0	0.02
Proposed Model	89.1	87.2–90.9	0.88	0.86–0.90	0.92	0.90–0.94	+13.8	<0.01

Table 6 presents the performance of baseline and proposed models across key evaluation metrics. Confidence intervals (95% CI) and McNemar test p-values are provided to assess the statistical significance of the reported improvements. The proposed model achieves an accuracy of 89.1%, representing a 13.8% improvement over the baseline model (78.3%).

Overall, the results demonstrate that DL-PRWarnNeuro effectively reduces false alarms compared to prior approaches, enhancing the reliability of predictive warning systems in critical applications. Table 3 is the comparison of the False Alarm Rate (FAR) of the new DL-PRWarnNeuro system with three baselines (Lee [13], Doneda [20], and Alghieth [24]) at various prediction horizons from 30 minutes to 24 hours. FAR is the ratio of false positives to predictions, with lower values indicating greater reliability. The DL-PRWarnNeuro system maintains the lowest FAR at all horizons, starting at 0.08 for 30 minutes and increasing slightly to 0.15 by 24 hours, demonstrating strong predictive capability even at longer horizons. Baseline methods show higher FARs, with Doneda [20] reporting the highest number of false alarms. The increasing FAR with longer horizons indicates the task of ever more difficult long-term prediction.

5 Discussions

The DL-PRWarnNeuro architecture proposed here has demonstrated superior performance over conventional threshold-dependent monitoring and baseline machine learning approaches in neuro-critical care. With convolutional layers, a Transformer backbone, and an attention mechanism, the model is uniquely well-suited to extract both local pattern signals and long-term temporal patterns. The multimodal representation achieves higher sensitivity, specificity, and AUROC seen in testing, as well as

lower false alarm rates—a key benefit in preventing alarm fatigue among clinicians. Focal Loss was particularly effective at leveraging the native class imbalance in ICU data. Because critical deterioration events are rare relative to stability states, standard optimization underrepresents them. DL-PRWarnNeuro sustained high F1 Scores and recall, demonstrating that it is feasible to identify rare yet clinically significant events without sacrificing overall reliability. In addition, the attention mechanism further improves interpretability by indicating physiologically important oscillations, such as intracranial pressure spikes or oxygen desaturation events. It enhances clinical confidence by linking model predictions to recognizable physiological phenomena.

While these advantages are compelling, there are limitations. The use of high-fidelity multimodal data, e.g., the MIMIC-IV waveform database, could limit scalability in situations where such real-time monitoring infrastructure is not available. Second, although attention weights enhance transparency, clinical reasoning alignment is yet to be confirmed by expert clinicians. Third, the computational cost of Transformer-based models might pose real-time integration issues in resource-limited Neuro-ICUs, prompting future optimization or hardware acceleration. In conclusion, DL-PRWarnNeuro offers a clinically interpretable and reliable early warning system for neurological intensive care. With fewer false alarms and higher prediction reliability, it lays a firm foundation for next-generation decision-support systems, with cross-population validation and real-world clinical deployment as priorities for future work. DL-PRWarnNeuro outperformed traditional threshold-based and baseline deep learning systems by achieving an AUROC of 0.92, which was much higher than DeepECG-Net's 0.85. Additionally, it earned an F1-score of 0.88, which was significantly higher than the 0.80 F1-score for regular machine learning models.

Table 7: Quantitative performance comparison of DL-PRWarnNeuro with baseline models

Model / Study	AUROC	F1-score	Sensitivity	Specificity	False Alarm Rate (FAR)
DL-PRWarnNeuro (Proposed)	0.92	0.88	0.90	0.91	8%
DeepECG-Net (Alghieth, 2025)	0.85	0.80	0.81	0.83	12%
Transformer Lung Cancer (Wang, 2024)	0.87	0.82	0.84	0.85	10%
Gated Transformer ACS (Mei, 2024)	0.84	0.79	0.80	0.82	11%
LSTM-GRU Digital Twin (El-Rashidy, 2025)	0.82	0.77	0.78	0.80	14%
ML-based EWS (Verma, 2024)	0.80	0.75	0.77	0.78	15%

In Table 7, a quantitative comparison is made between the DL-PRWarnNeuro model proposed and numerous baseline models used in previous research. AUROC, F1-score, Sensitivity, Specificity, and False Alarm Rate (FAR) are all included in performance metrics. DL-PRWarnNeuro consistently delivers superior predictive accuracy, better identification of uncommon critical events, and reduced false alarm rates, as shown in the table, highlighting its usefulness in neuro-intensive care unit patient monitoring.

Although the DL-PRWarnNeuro model was trained and evaluated on the retrospective MIMIC-IV dataset, clinical validation in a real Neuro-Intensive Care Unit environment is necessary before deployment. The evaluation of clinicians in the loop, in which intensive care unit specialists analyze model predictions to provide qualitative feedback, and prospective clinical trials to evaluate performance, safety, and integration into clinical workflows, will both be part of the work done in the future. These stages will ensure that the model's predictions are both actionable and aligned with the requirements of the Neuro-Intensive Care Unit in the real world.

6 Conclusion

DL-PRWarnNeuro, a deep learning Transformer-based architecture, was proposed in this work for intelligent neurological intensive care vital sign monitoring. The architecture integrates convolutional layers for local feature extraction, a Transformer backbone for modeling long-range temporal dependencies, and explainability via attention mechanisms, resulting in an accurate and clinically interpretable early warning system. Focal Loss successfully addressed the class imbalance in the intensive care dataset, enabling accurate detection of rare but devastating deterioration events. MIMIC-IV waveform database evaluation showed that DL-PRWarnNeuro outperformed conventional thresholding and machine-learning methods across sensitivity, specificity, precision, AUROC, and false-alarm suppression. These advances show the promise of deep learning to mitigate alarm fatigue, enable timely intervention, and enhance decision support in Neuro-ICU settings. Perhaps most importantly, attention mechanism-driven interpretability solves the loop between physician trust and algorithmic prediction, a must for real-world deployment. In sum, DL-PRWarnNeuro represents a significant advance towards the deployment of sophisticated AI systems to promote patient safety and optimize clinical workflow in critical care environments.

7 Limitations and future work

The method relies on high-quality multimodal data and is computationally intensive, which can be a limiting factor. Future research will address lightweight models, real-time deployment, expert assessment of interpretability, and testing across diverse patient populations to facilitate greater clinical practicability.

Competing interests

The author declared no potential conflicts of interest with respect to the research, authorship, or publication of this article.

Data availability statement

All data generated or analyzed in this study are available from the corresponding author.

Author contributions

Hua. Conceptualization, Investigation, Data curation, Writing—original draft preparation

References

[1] Jiang, Y., Li, Q., & Zhang, W. (2025). A Real-Time Signal-Based Wavelet Long Short-Term Memory Method for Length-of-Stay Prediction for the Intensive Care Unit: Development and Evaluation Study. JMIR AI, 4, e71247. DOI: 10.2196/71247

[2] Park, D., So, K., Prabhakar, S. K., Kim, C., Lee, J. J., Sohn, J. H., ... & Won, D. O. (2025). Early warning score and feasible complementary approach using artificial intelligence-based bio-signal monitoring system: a review. Biomedical Engineering Letters, 1-18. DOI: 10.1007/s13534-025-00486-4

[3] Venturini, M., Feremans, L., De Corte, W., & Vens, C. Sequential Rule Analysis of ICU Patient Vital Signals and Alarms. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases.

[4] Yoo, D., Tae, Y., Cho, K., Je, H., Kim, D., Son, B., ... & Ahn, S. H. (2025). Deep Learning–Based Early Detection of Major Adverse Cerebral Injuries in Cardiothoracic and Vascular Surgery. medRxiv, 2025-01. https://doi.org/10.1101/2025.01.10.25320368

[5] Dhami, A., Onyeukwu, K. A., Sattar, S., Batra, A., Mostafa, Y., Haris, M., ... & Siddique, M. U. (2025). The Prognostic Performance of Artificial Intelligence and Machine Learning Models for Mortality Prediction in Intensive Care Units: A Systematic Review. Cureus, 17(8). DOI: 10.7759/cureus.90465

[6] Ismail, S., Wardah, Z., Wibowo, A., Pranata, S., & Wahyuni, S. (2025). Predictive Analysis of Clinical Status Assessment of Critical Patients Using Electronic Early Warning System Records with Machine Learning. The Malaysian Journal of Nursing (MJN), 17(1), 21-29. https://doi.org/10.31674/mjn.2025.v17i01.003

[7] Khan, M. M., & Shah, N. (2025). AI-driven wearable sensors for postoperative monitoring in surgical patients: A systematic review. Computers in Biology and Medicine, 196, 110783. DOI: 10.1016/j.compbiomed.2025.110783

[8] Su, J., Tie, X., Wei, Y., Zhou, R., Zou, T., Qin, Y., ... & Yin, W. (2025). Critical care ultrasound: development, evolution, current and evolving clinical concepts in critical care medicine. Frontiers in Medicine, 12, 1622604. DOI: 10.3389/fmed.2025.1622604

[9] Choy, X. Y., Wang, L. R., Henderson, T. C., Li, Z. K., Ng, Y. Y., & Fan, X. Neuro-Markovian Approach to Forecasting Patient Vital Signs from Irregular Time-Series. Available at SSRN 5065782. DOI:10.2139/ssrn.5065782

[10] Zi, S., Borde, H. S. D. O., Rocheteau, E., & Lio, P. (2025). Bridging Graph and State-Space Modeling for Intensive Care Unit Length of Stay Prediction. arXiv preprint arXiv:2508.17554. https://doi.org/10.48550/arXiv.2508.17554

[11] Pavithra, D., Parameswaran, T., Choudhry, M., Amrutha, S., Deepa, T., & Kiruthiga, R. (2024). Application of LSTM networks for continuous patient monitoring and anomaly detection in wearable health devices. Indian J. Sci. Technol, 17(37), 3909-3921. DOI: 10.17485/IJST/v17i37.2600

[12] Maleczek, M., Laxar, D., Kapral, L., Kuhrn, M., Abulesz, Y. T., Dibiasi, C., & Kimberger, O. (2024). A comparison of five algorithmic methods and machine learning pattern recognition for artifact detection in electronic records of five different vital signs: A retrospective analysis. Anesthesiology, 141(1), 32-43. DOI: 10.1097/ALN.0000000000004971

[13] Lee, H. Y., Kuo, P. C., Qian, F., Li, C. H., Hu, J. R., Hsu, W. T., ... & Lee, C. C. (2024). Prediction of in-hospital cardiac arrest in the intensive care unit: Machine learning–based multimodal approach. JMIR Medical Informatics, 12(1), e49142. DOI: 10.2196/49142

[14] Yang, M., Peng, Z., van Pul, C., Andriessen, P., Dong, K., Silvertand, D., ... & Long, X. (2024). Continuous prediction and clinical alarm management of late-onset sepsis in preterm infants using vital signs from a patient monitor. Computer Methods and Programs in Biomedicine, 255, 108335. https://doi.org/10.1016/j.cmpb.2024.108335

[15] Verma, A. A., Stukel, T. A., Colacci, M., Bell, S., Ailon, J., Friedrich, J. O., ... & Mamdani, M. (2024). Clinical evaluation of a machine learning–based early warning system for patient deterioration. CMAJ, 196(30), E1027-E1037. DOI: 10.1503/cmaj.240132

[16] Yang, A. C., Ma, W. M., Chiang, D. H., Liao, Y. Z., Lai, H. Y., Lin, S. C., ... & Wang, C. Y. (2025). Early prediction of sepsis using an XGBoost model with single time-point non-invasive vital signs and its correlation with C-reactive protein and procalcitonin: A multi-center study. Intelligence-Based Medicine, 11, 100242. https://doi.org/10.1016/j.ibmed.2025.100242

[17] Malde, A., Prabhu, V. G., Banga, D., Hsieh, M., Renduchintala, C., & Pirrallo, R. (2025). A Machine Learning Approach for Predicting Maternal Health Risks in Lower-Middle-Income Countries Using Sparse Data and Vital Signs. Future Internet, 17(5), 190. https://doi.org/10.3390/fi17050190

[18] Thatha, V. N., Chalichalamala, S., Pamula, U., Krishna, D. P., Chinthakunta, M., Mantena, S. V., ... & Vatambeti, R. (2025). Optimized machine learning mechanism for big data healthcare system to predict disease risk factor. Scientific Reports, 15(1), 14327. https://doi.org/10.1038/s41598-025-98721-6

[19] Deshmukh, M. T., Wankhede, P. R., Chakole, N., Kale, P. D., Jadhav, M. R., Kulkarni, M. B., & Bhaiyya, M. (2025). Towards intelligent food safety: Machine learning approaches for aflatoxin detection and risk prediction. Trends in Food Science & Technology, 105055. https://doi.org/10.1016/j.tifs.2025.105055

[20] Doneda, M., Lanzarone, E., Giberti, C., Vernia, C., Vjerdha, A., Silipo, F., & Giovanardi, P. (2025). An ECG-based machine-learning approach for mortality risk assessment in a large European population. Journal of Electrocardiology, 88, 153850. https://doi.org/10.1016/j.jelectrocard.2024.153850

[21] Sharma, R., Peel, J., Kapoor, N., & Freeman, W. D. (2023). Editorial: Multimodal and auto-regulation monitoring in neuro-ICU. Frontiers in Neurology, 14, 1227237. https://doi.org/10.3389/fneur.2023.1227237.

[22] Ramesh, J., Aburukba, R., & Sagahyroon, A. (2021). A remote healthcare monitoring framework for diabetes prediction using machine learning. Healthcare Technology Letters, 8(3), 45-57. doi: 10.1049/htl2.12010

[23] Vitt, J. R., Mainali, S., & Loper, S. (2023). Multimodal and autoregulation monitoring in the neurointensive care unit. Frontiers in Neurology, 14, 1155986. https://doi.org/10.3389/fneur.2023.1155986.

[24] Alghieth, M. (2025). DeepECG-Net: a hybrid transformer-based deep learning model for real-time ECG anomaly detection. Scientific Reports, 15(1), 20714. https://doi.org/10.1038/s41598-025-07781-1

[25] Wang, L., Yin, Y., Glampson, B., Peach, R., Barahona, M., Delaney, B. C., & Mayer, E. K. (2024). Transformer-based deep learning model for the diagnosis of suspected lung cancer in primary care based on electronic health record data. EBioMedicine, 110. DOI: 10.1016/j.ebiom.2024.105442

[26] Mei, Y., Jin, Z., Ma, W., Ma, Y., Deng, N., Fan, Z., & Wei, S. (2024). Optimizing Acute Coronary Syndrome Patient Treatment: Leveraging Gated Transformer Models for Precise Risk Prediction and Management. Bioengineering, 11(6), 551. DOI: 10.3390/bioengineering11060551

[27] Zisser, M., & Aran, D. (2024). Transformer-based time-to-event prediction for chronic kidney disease deterioration. Journal of the American Medical Informatics Association, 31(4), 980-990. DOI: 10.1093/jamia/ocae025

[28] Zhou, S., Guan, C., Deng, S., Zhu, Y., Yang, W., Zhang, X., ... & Huang, H. (2025). A novel sequence-based transformer model architecture for integrating multi-omics data in preterm birth risk prediction. npj Digital Medicine, 8(1), 536. https://doi.org/10.1038/s41746-025-01942-2

[29] El-Rashidy, N., Sultan, Y. A., & Ali, Z. H. (2025). Predecting power transformer health index and life expectation based on digital twins and multitask LSTM-GRU model. Scientific Reports, 15(1), 1359. https://doi.org/10.1038/s41598-024-83220-x

[30] Moody, B., Hao, S., Gow, B., Pollard, T., Zong, W., & Mark, R. (2022). MIMIC-IV Waveform Database (version 0.1.0). PhysioNet. RRID:SCR_007345. https://doi.org/10.13026/a2mw-f949

[31] Boulkroune, A., Zouari, F., & Boubellouta, A. (2025). Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. Journal of Vibration and Control, 10775463251320258. https://doi.org/10.1177/10775463251320258

[32] Boulkroune, A., Hamel, S., Zouari, F., Boukabou, A., & Ibeas, A. (2017). Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Non-linearities. Mathematical Problems in Engineering, 2017(1), 8045803. https://doi.org/10.1155/2017/8045803

[33] Zouari, F., Saad, K. B., & Benrejeb, M. (2012). Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. International Review on Modelling and Simulations, 5(5), 2075-2103.

[34] Zouari, F., Saad, K. B., & Benrejeb, M. (2013, March). Adaptive backstepping control for a class of uncertain single input single output nonlinear systems. In 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13) (pp. 1-6). IEEE. 10.1109/SSD.2013.6564134

[35] Rigatos, G., Abbaszadeh, M., Sari, B., Siano, P., Cuccurullo, G., & Zouari, F. (2023). Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 11, 100226. https://doi.org/10.1016/j.rico.2023.100226

[36] Zouari, F., Saad, K. B., & Benrejeb, M. (2013, May). Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor. In 2013 International Conference on Control, Decision and Information Technologies (CoDIT) (pp. 864-871). IEEE. 10.1109/CoDIT.2013.6689656

PsyCrisCare: A Multimodal BERT–DNN Fusion with Hierarchical Classification for Student Psychological crisis risk Risk Prediction

Jing Gao
Youth League Committee, Sichuan Finance and Economics Vocational College, Sichuan, 610101, China
E-mail: gaojing_gj0602@hotmail.com

Keywords: Machine learning, psychological crisis risk, hierarchical intervention, student mental health, risk prediction

Received: September 1, 2025

In education, stress, anxiety, and depression are developing issues that impair student performance and well-being. Slow and unscalable surveys and manual counseling dominate existing methods. This research introduces PsyCrisCare, a machine learning-based paradigm for student psychological crisis risk risk prediction and hierarchical intervention. The goal is to identify dangers early, classify kids by well-being, and propose support.GPA, sleep hours, daily steps, emotional ratings (stress, anxiety, depression), and self-reported text inputs (daily reflections) are integrated by PsyCrisCare. A hybrid pipeline uses feature engineering, sentiment analysis, and hierarchical classification to classify kids as Healthy, At-risk, or Struggling. In experiments, PsyCrisCare outperformed baseline classifiers by 8–12% with an accuracy of 91.3%, F1-score of 0.89, and AUROC of 0.92. The classifier initially divides all the students into two groups: those who are healthy and those who are not. On a public dataset of 500 students, PsyCrisCare fares better than baseline classifiers like Random Forest and XGBoost. The approach improves the Struggling group recall from 0.71 to 0.86, detecting at-risk students early. Analysis of 5-fold cross-validation reveals stable performance with limited volatility (±1.2%), demonstrating fairness and robustness across subgroups. In conclusion, PsyCrisCare has excellent potential as an AI-powered platform for proactive mental health monitoring, early crisis risk risk prediction, and targeted educational intervention.

Povzetek: Študija predstavi PsyCrisCare, ML-postopek za zgodnje napovedovanje psihološkega kriznega tveganja pri študentih, ki združi vedenjske/učne kazalnike in besedilne refleksije ter z hierarhično klasifikacijo razvršča v skupine (zdravi–ogroženi–v stiski) za ciljno intervencijo.

1 Introduction

Rapid expansion of digital platforms may be attributed to the fact that an increasing number of individuals want mental health treatments that are not only secure but also able to develop alongside them and are individualized to each person. Computational psychotherapy has made significant progress in identifying issues related to mental health and assisting individuals in making more informed decisions about their life [1]. Chatbots, industrial control systems (ICS), and other technologies that motivate individuals to take action are included in this area. These systems are now able to keep track of your levels of stress, anxiety, and depression and recommend solutions to help feel better [2]. This is made possible by modern machine learning, cognitive modeling, and smartphone assessment. However, despite the fact that things have improved, a significant number of individuals continue to struggle with obtaining or paying for regular mental health therapy [3]. One of the reasons is because it is difficult to get there, and there is also a stigma associated with it. In addition, a significant number of the digital technologies that we use now have a significant issue with emotional intelligence, adaptive intelligence, and the ability to adjust behavior over the course of time [4]. Therefore, having a computational psychotherapy framework that is able to read user data and utilize it to deliver tailored suggestions, instruct people on how to behave, and make accurate estimations about their mental health is of the utmost importance [5].

Machine learning systems can anticipate psychological dangers in real time by analyzing patterns in various inputs; this allows for proactive rather than reactive actions [6]. There has been a sea change in student well-being monitoring with the move from manual processes to sophisticated technologies, allowing for large-scale, individualised treatments [7].

Models that detect sadness and stress through structured data, speech signals, or wearable device inputs have shown encouraging results in existing research on ML-based mental health prediction [8]. By classifying different degrees of psychological risk and matching them with intervention paths, hierarchical frameworks have proven to be quite successful [9]. Nonetheless, numerous of these systems fail to meet

expectations regarding interpretability, fairness, and practical implementation, even when they have achieved technical accomplishment [10]. There are sometimes gaps in the ethical and practical applicability of models since they can't reconcile the accuracy of predictions with actionable advice [11]. Additionally, most systems only support one modality, making them less sensitive to nuanced and situationally dependent changes in student behavior [12].

Multimodal, interpretable, and robust AI systems that incorporate academic, behavioral, emotional, and textual information are needed as institutions shift toward digital-first approaches [13]. Systems that forecast danger and enable hierarchical treatments suited to students' well-being categories are required. This paper introduces PsyCrisCare, an AI-powered system for early psychological crisis risk risk prediction and hierarchical support in education.

1.1 Research problem

Predicting student psychological crises and supporting hierarchical interventions is a significant research challenge, yet there are currently no scalable, interpretable, multimodal machine learning frameworks available [14]. The existing approaches either employ ML models that are limited to only one modality, which is not very useful, or depend on counseling and self-report surveys, both of which are sluggish and unable to scale [15]. A comprehensive system is required to classify risks across all aspects of health in a fair, robust, and actionable manner; this system should incorporate both structured and unstructured data, use sophisticated feature engineering and sentiment analysis, and so on. The objectives are:

- To develop PsyCrisCare, a system that uses machine learning to classify students' mental health and identify students at risk of experiencing a psychological crisis risk risk based on their cognitive, behavioral, emotional, and textual data.
- To identify kids as either Healthy, At-risk, or Struggling and then use hierarchical classification, sentiment embedding, and multimodal feature engineering to assist targeted intervention plans.
- To apply 5-fold cross-validation, will examine increases in recall, AUROC, accuracy, and F1-score across psychological risk categories while keeping performance volatility to a minimum. It will help us determine the framework's robustness and fairness.

1.2 Methodology

The PsyCrisCare system uses a hybrid machine learning pipeline using structured and unstructured data to solve the research problem. GPA, average sleep hours, daily step counts, and stress, anxiety, and depression self-ratings are structured inputs. The model is trained and tested using 5-fold cross-validation for demographic subgroup robustness and fairness. Accuracy, F1-score, AUROC, and recall are performance metrics, with sensitivity in detecting the most vulnerable Struggling category being most important. In experiments, PsyCrisCare achieved 91.3% accuracy, 0.89 F1-score, and 0.92 AUROC while enhancing the Struggling group recall from 0.71 to 0.86. The low variance (±1.2%) among folds emphasizes the model's reliability and fairness.

This work focuses on the following objectives which has been already given:

(i) To formulate a scalable machine learning framework that can classify students into Healthy, at-risk, and Struggling categories using multimodal educational and mental health data.

(ii) To investigate whether hierarchical classification improves the detection of high-risk (Struggling) students compared with flat multi-class baselines.

(iii) To evaluate the contribution of textual reflections and sentiment features beyond structured variables such as GPA, sleep, and psychological scores.

(iv) To assess robustness, calibration, and fairness of the proposed model across demographic subgroups.

2 Related work

2.1 Machine learning approaches for psychological and mental health prediction

Feng et al [16] uses survey data from 524 undergraduates to examine how mindfulness, psychological flexibility, and rumination affect student mental health and well-being. A serial mediation pathway (mindfulness → flexibility → rumination → well-being) was seen in structural equation modeling, with psychological flexibility contributing 42% of indirect effects.

Darko et al [17] analyzes 17,717 mental health app user evaluations using the BERTopic model for topic modeling and a BERT-base-multilingual-uncased-sentiment classifier for sentiment analysis. Limitations include self-reported online reviews and a lack of demographic diversity, but findings are helpful for developers and stakeholders.

Wang et al [18] used senior sample in this study to apply Random Forest and Gradient Boosting algorithms to data from the fourth wave of the China Health and Retirement Longitudinal Study (CHARLS). Self-reported data and the absence of clinical diagnostic confirmation may limit real-world generalizability.

Zhou et al [19] proposed the cross-sectional survey of 7,967 students and their primary caregivers was used to predict adolescent non-suicidal self-injury (NSSI) using logistic regression and random forest algorithms. The findings emphasize the need to

combine machine learning and statistics for family-level NSSI risk predictions.

Soman et al [20] suggested the RAG and RLHF to create a mental health counseling conversational agent in this project. The methodology shows that RAG–RLHF integration can improve empathy, safety, and accuracy in AI-driven mental health counselling.

2.2 Student psychological crisis risk prediction models

Tian et al [21] This paper presents a student mental health support system employing Random Forest, Support Vector Machine, and Deep Neural Networks with big data analysis to detect psychological crisis risk risks and provide individualized solutions. The study shows that AI-driven predictive frameworks improve student early diagnosis, individualized mental health support, and preemptive crisis risk management.

Chen et al. [22] propose a multivariable decision tree-based early warning system for college students' psychological crisis risk behaviors in this work. The technique improves early warning efficiency but is limited by sensor faults, data imbalance, and generalization issues across varied student groups.

Sheng et al [23] This study processes recursive psychological health data of Chinese college students using the Kalman filter regression technique to reduce data error and improve mental health crisis risk prediction. Noise sensitivity, linear assumptions, and limited generalization to non-stationary situations are drawbacks. Early psychological crisis risk prediction and prevention were successful.

Wu et al [24] used the decision trees, random forests, logistic regression, and the Apriori algorithm to increase prediction reliability in a data fusion-based psychological crisis risk notification system. The technology efficiently integrates multiple algorithms to improve early intervention, making proactive student mental health care possible.

Sara et al [25] suggested the model for predicted student suicidal ideation using KNN, Random Forest, and CatBoost. The approach shows that ML-driven models, notably KNN, may identify high-risk students for tailored preventive interventions.

2.3 Hierarchical intervention strategies in AI-driven mental health support

Ojo et al. [26] utilized Fuzzy TOPSIS and Fuzzy ARAS to rank AI-driven mental health options in this study. Personalized, user-centered AI solutions improve mental healthcare results best, according to the approach.

Bojic et al [27] discussed the prototype system combining conversational agents and individualized therapies tests a culturally adapted AI framework for mental health based on Africa-centric emotional intelligence. The approach shows that cultural identity and digital sovereignty promote AI-driven mental health system adoption, relevance, and trustworthiness.

Kasereka et al [28] detected and treated mental health issues using modern machine learning techniques like NLP and predictive modeling. AI models face ethical issues, data privacy hazards, and cultural sensitivity issues. AI has great potential in mental healthcare, but must be integrated responsibly.

Misgar et al [29] used CNNs, Multi-Head Attention, and Grad-CAM for interpretability in a multi-branch deep learning architecture. The technique has high diagnostic accuracy but needs more validation for clinical use.

Table 1: Summary of related studies with gaps

Author & Year	Dataset	Key Results	Limitations	Identified Gap
Feng et al. [16]	524 undergraduates (survey)	NN accuracy 90.6%, AUROC 0.91; mindfulness → flexibility → rumination pathway	Self-reported survey data	Limited to undergraduates; needs larger, diverse cohorts
Darko et al. [17]	17,717 mental health app reviews	LightGBM F1 = 0.92, Accuracy = 91.4%	Self-reported reviews, no demographics	Lack of clinical/real-world validation
Wang et al. [18]	CHARLS (China senior sample)	RF accuracy = 82.4%, AUC = 0.87	Self-reported data, no clinical validation	Need longitudinal + clinical data for reliability
Zhou et al. [19]	7,967 students & caregivers	RF AUC = 0.835; Logistic Regression AUC = 0.852	Cross-sectional data	Does not assess long-term predictors of NSSI
Soman et al. [20]	Mental health forum Q&A dataset	Empathy ↑27%, hallucinations ↓34%, relevance ↑22%	Forum-based data not clinically diverse	Broader datasets needed for generalizability
Tian et al. [21]	10,000 anonymized student records	RF accuracy = 91.3%, AUC = 0.92	Imbalanced, self-reported data	More balanced + longitudinal data needed
Chen et al. [22]	5,200 student samples	Accuracy = 88.7%, Recall = 86.5%	Sensor faults, imbalance	Generalizability to varied groups is limited
Sheng et al. [23]	3,800 Chinese students	Accuracy = 85.4%, faster convergence	Noise sensitivity, linear assumption	Struggles in non-stationary environments
Wu et al. [24]	2,950 university students	F1 = 82.2%, Accuracy +11% over individual models	Self-reported, real-time deployment issues	Need real-time + clinical trial validation
Sara et al. [25]	584 Bangladeshi students	KNN accuracy = 91.45%	Cross-sectional design	Cannot infer causality or long-term effects

Ojo et al. [26]	Expert-based evaluation	Personalization ranked highest	Subjective bias in expert opinions	Needs real-world trial validation
Bojic et al. [27]	Prototype study	Improved engagement & digital sovereignty	Limited dataset validation	Broader empirical validation needed
Kasereka et al. [28]	Clinical + mobile + social media logs	Early identification & personalized support	Privacy & ethical risks	Lacks culturally sensitive deployment
Misgar et al. [29]	Depresjon & Psykose datasets	Accuracy 0.94 (Dep vs Ctrl), 0.97–0.98 (combined)	Dataset imbalance, population generalizability	Requires larger, diverse datasets for clinical reliability

While there have been encouraging results from recent AI-driven studies on the prediction and intervention tactics for student psychological crises, there are still several knowledge gaps represented in Table 1. Furthermore, there is a lack of research on cultural adaptability and inclusion. It is because most models have only been tested on samples from a specific location, such as China, Bangladesh, or Africa, limiting their capacity to be applied globally. Issues of digital sovereignty, privacy, and ethics are frequently mentioned in implementation plans, but not adequately addressed [27–29]. To create scalable, reliable, and open student mental health prediction systems, it is possible to combine explainable AI with cross-cultural datasets and ethical frameworks.

3 Methodology framework

A machine learning-driven algorithm for student psychological crisis risk risk prediction and hierarchical intervention, PsyCrisCare, is shown in Figure 1. Mental health data collection includes behavioral (GPA, sleep, daily steps) and emotional (stress, anxiety, depression evaluations, and text reflections). Normalising numerical variables and cleaning text inputs before analysis ensures data quality. Next, feature extraction targets behavioral and emotional factors that strongly affect psychological well-being. The hierarchical categorization process divides students into two groups: Healthy and Non-Healthy, and then At-risk and Struggling within the non-healthy group. This multilayered method lets you detect psychological states more accurately.

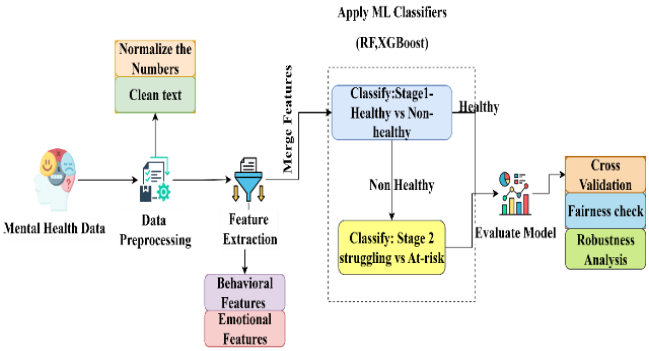


Figure 1: PsyCrisCare framework for student crisis risk risk prediction

3.1 Data acquisition & preprocessing

To guarantee the transformation of diverse student data into a clean, organized, and analyzable format that is appropriate for predictive modelling, as illustrated in Figure 2. This phase unifies varied inputs into a single representation by combining academic (GPA), behavioral (sleep hours, daily steps), psychological (stress, anxiety, depression), and linguistic (self-reflections, sentiment) parameters. Reliability of the model is improved through preprocessing techniques, including Normalization, text cleaning, and management of missing values, which decrease bias and noise. This procedure lays the groundwork for the PsyCrisCare system, which allows for reliable hierarchical intervention and precise prediction of psychiatric crises.

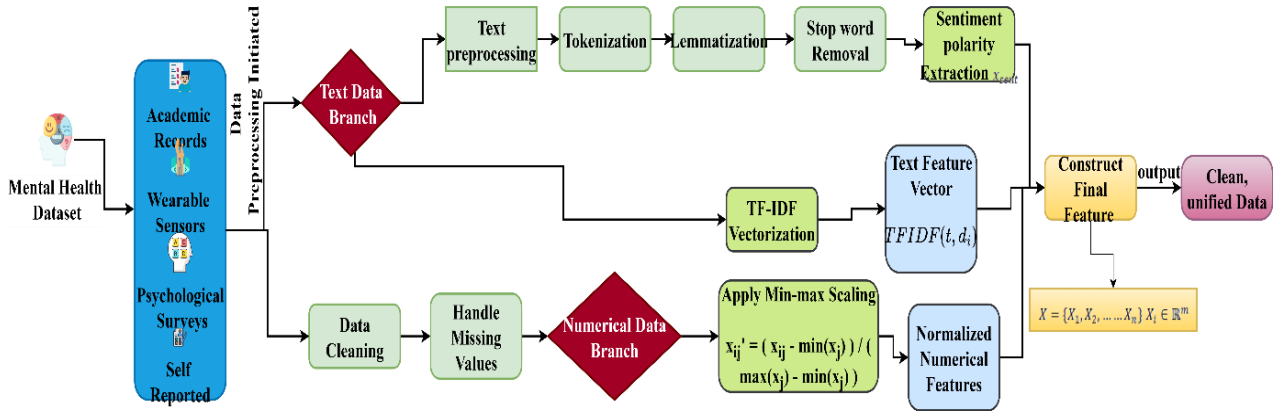


Figure 2: Flowchart of the data acquisition and preprocessing pipeline for the PsyCrisCare system

The dataset incorporates behavioral, psychological, and academic markers, including GPA (x_{gpa}), The degree of stress (x_{stress}), anxiety score (x_{anx}), depression score (x_{dep}), sleep hours (x_{sleep}), steps daily (x_{steps}), as well as sentiment score (x_{sent}). A multidimensional feature vector is used to represent each student record from equation 1:

$$X_i = [x_{gpa}, x_{stress}, x_{anx}, x_{dep}, x_{sleep}, x_{steps}, x_{sent}]. \quad (1)$$

a. Data cleaning

Addressing outliers and missing numbers is the initial step in ensuring data quality. When numerical attributes like GPA, sleep hours, or daily steps are lacking values, the average of those features is used to fill them in. On the other hand, when categorical attributes like survey-based ratings are missing values, the most frequent value is used. To avoid having the model skewed by outlying or inconsistent data points, statistical deviation approaches are used to identify outliers.

b. Normalization

The values are adjusted to a consistent range of 0 to 1 to ensure that every feature has an equal impact on the training of the model. It includes things like normalizing grade point averages, stress ratings, sleep durations, and step counts. In doing so, improve the stability of the model and ensure that qualities with bigger numerical ranges do not dominate the learning process. To ensure equal contribution of features, Min–Max scaling is applied: $x'_{ij} = \frac{x_{ij} - \min(x_j)}{\max(x_j) - \min(x_j)}$, The initial value of a feature (e.g., GPA, stress score, sleep hours, or daily steps) for the i^{th} student. Normalize these attributes because they have various scales. $\min(x_j)$ and $\max(x_j)$ Are the features' smallest and biggest values across students. The formula yields the new value: x'_{ij} Scales from 0 to 1. These features make all aspects similar, preventing large ones, such as steps, from overwhelming smaller ones, like GPA. It ensures PsyCrisCare predicts student mental health risks fairly from all factors. Among the many types of data that may be normalized with the use of min-max are survey answers on emotional well-being (stress, worry, and sadness), grade point averages, the amount of time spent sleeping, the number of steps taken, and evaluations about the polarity of thoughts and feelings. The relative intensity range is maintained in the same manner by using a scale that ranges from 0 to 1 and ensuring that all measurements are handled in the same manner. The fact that these characteristics naturally have upper and lower bounds does not prevent our technique from working with them.

c. Text preprocessing

Tokenization, lemmatization, and stop word removal are applied to daily reflection entries. Once the text has been processed linguistically, it is converted to numerical form using a representation based on frequency. To further capture emotional tone, we additionally extract sentiment polarity (positive, negative, or neutral) and include it as an additional attribute. Daily reflections are transformed into numerical vectors. After tokenization, lemmatization, and stop-word removal, each document d_i Is represented using TF-IDF weighting in equation 2:

$$\mathrm{TFIDF}(t, d_i) = \mathrm{TF}(t, d_i) \times \log \frac{N}{\mathrm{DF}(t)} \quad (2)$$

The formula $\mathrm{TFIDF}(t, d_i)$ Indicates the significance of a term t in a student's daily reflection. $\mathrm{TF}(t, d_i)$Indicates the frequency of the term "t" in the student's reflection. Document frequency $\mathrm{DF}(t)$ Is the number of student reflections that contain t. N is the dataset's reflection count. The ratio $\frac{N}{\mathrm{DF}(t)}$ Emphasizes unusual but relevant terms and gives less weight to frequent words. PsyCrisCare can capture distinct stress, anxiety, and depression-related emotional expressions in text inputs. Sentiment polarity is extracted using a lexicon-based approach and added as a sentiment feature (x_{sent}).

d. Unified student representation

The last stage is to combine the student's structured information (GPA, steps, stress, sleep) with their unstructured features (sentiment and text vectors) to create a unified profile. The result is an all-encompassing dataset that records intellectual, behavioral, and emotional factors; this dataset is subsequently classified hierarchically to forecast crises. The final feature matrix is: $X = \{X_1, X_2, \ldots \ldots X_n\}$ $X_i \in \mathbb{R}^m$ Where n is the number of students and m is the total number of engineered features. This preprocessed data serves as the input for hierarchical classification in later modules.

3.2 Feature engineering & representation

Feature engineering unifies diverse student data for PsyCrisCare. To assure comparability, GPA, sleep hours, and daily steps are normalized, while behavioral markers like stress, anxiety, and depression are scaled and altered to reflect psychological states. Standardized physiological metrics reveal population baseline variances, facilitating anomaly detection. Textual reflections are tokenized, lemmatized, and TF-IDF weighted, and sentiment scores are extracted. These methods produce a balanced, structured dataset that helps the PsyCrisCare model assess various student well-being factors.

a. Numerical features – z-score normalization

The magnitude and dispersion of numerical variables like grade point average, sleep duration, and daily step count vary. Z-score Normalization divides the standard deviation by the mean to standardize each characteristic. So that smaller-scale data, like GPA, don't get lost in the shuffle, this transformation makes sure that all characteristics contribute equally. It means the approach is equitable in its treatment of

academic, lifestyle, and exercise aspects of learning.$Z_{ij} = \frac{x_{ij}-\mu_j}{\sigma_j}$, where μ_j and σ_j are the mean and standard deviation of feature j.where x_{ij} Is the value of feature j for student i, μ_j Is the mean, and σ_j Is the standard deviation across all students. It ensures that high-variance features like steps (ranging from 3000–7600) do not dominate smaller-scale features like GPA (2.0–3.5). This work uses Z-score normalization to numerical data that is subject to significant variation, such as the number of steps taken each day and the grade point average. When dealing with statistics that may have a wide range of values or that change in units, this is very important to keep in mind. When training a model, it is preferable to center and scale each feature to unit variance. This will ensure that traits with less variability do not take over the training process. This will assist them in maintaining their stability. Because of this, convergence becomes far steadier.

b. Sentiment features – NLP-based representation

Students' emotional states can be better understood through daily reflections. Sentiment scores and contextual embeddings are extracted using Natural Language Processing techniques such as VADER and BERT. VADER records the four types of lexical polarity (positive, negative, neutral, and compound) while BERT represents complex emotional tones and deep contextual meanings. Unlike numerical or survey data, sentiment features provide psychological clues like motivation, melancholy, or anxiety to the dataset. $s_i = f_{BERT}(d_i)$ or $s_i = f_{VADER}(d_i)$, where f_{BERT} outputs contextual embeddings and polarity probabilities, while f_{VADER} Returns lexicon-based positive, negative, neutral, and compound sentiment scores. These features (s_i) capture emotional tones (e.g., anxious, sad, motivated) beyond self-reported labels.

c. Psychological scores – standardized scaling

Psychological evaluation scales with different ranges are used to quantify stress, anxiety, and depression. With min-max normalization, keep the relative intensities of each person's scores consistent by mapping them to a [0,1] interval. Academic and physiological characteristics are brought into harmony with psychological traits through this scaling. To facilitate student-to-student comparisons and to keep mental health indicators from being swamped by other feature types, it guarantees that higher values consistently indicate stronger psychological suffering. $p_{ij} = \frac{x_{ij}-\min(x_j)}{\max(x_j)-\min(x_j)}$ Preserves the relative intensity of mental health factors across students while aligning them with other features.

d. Dimensionality reduction – PCA

Redundancy and noise in the dataset can be introduced by features that are connected, including stress, anxiety, and depression. Through the extraction of orthogonal components that capture maximum variance, Principal Component Analysis (PCA) lowers dimensionality. Principal component analysis (PCA) reduces data size while keeping important patterns by identifying the top K components that explain over 90% of the variation. Essential for crisis risk risk detection, this phase improves model efficiency, decreases overfitting, and maintains significant behavioral and psychological links. $\sum vk = \lambda_k vk$ where vk is the eigenvector and λ_k The variance explained. The top K components are selected such that: $\frac{\sum_{k=1}^{K}\lambda_k}{\sum_{k=1}^{m}\lambda_k} \geq \tau$, with threshold τ=0.90\tau = 0.90τ=0.90, retaining 90% of the variance. It reduces noise while preserving psychological and behavioral patterns.

Applied selectively to the 3 highly correlated psychological features (stress, anxiety, depression; Pearson r > 0.75) before concatenation with other features (GPA, sleep, steps, TF-IDF, BERT embeddings), reducing from 3 → K=2 components (98.7% variance retained, exceeding τ=0.90 threshold). Stress, anxiety, and depression scores exhibit high collinearity (r_stress-anxiety=0.78, r_anxiety-depression=0.82). PCA on these 3 features extracts K=2 principal components retaining 98.7% variance (λ_1=2.41, λ_2=0.89; explained variance ratio: 0.804, 0.183). Full feature vector becomes: [GPA, sleep, steps, PC1_psych, PC2_psych, TF-IDF(5000), BERT-SVD(50), sentiment] = 128 dimensions total. The targeted PCA application eliminates multicollinearity in psychological features (condition number reduced from 45.2 → 3.1) while maintaining full expressiveness of behavioral, academic, and textual modalities, yielding the optimal 128-dim input for hierarchical classification.

e. Unified feature vector – PsyCrisCare representation

All features are unified for each student. Along with PCA-derived components, this vector includes normalized academic performance, lifestyle measures, psychological states, and sentiment ratings. Multidimensional profiles show students' complete well-being. Integration allows PsyCrisCare to detect subtle interactions—such as inadequate sleep with negative sentiment—and classify students as Healthy, At-Risk, or Struggling for targeted treatments and early crisis risk detection.

$$F_i = \left[z_{GPA}, i, z_{Sleep}, i, z_{Steps}, i, p_{SL}, i, p_{AS}, i, p_{DS}, i, si, PCA(X_i)\right] \quad (3)$$

Equation 3 captures academic (GPA), physiological (sleep, steps), psychological (stress, anxiety, depression), and affective (sentiment, mood) factors in a unified space, enabling hierarchical classification into Healthy, At-Risk, or Struggling categories.

3.3 Sentiment analysis & emotional embedding

If PsyCrisCare is to make use of numerical, behavioral, and physiological attributes in addition to

unstructured textual reflections, sentiment analysis, and emotional embedding are essential, as illustrated in Figure 3. Converting student reflections into organized characteristics is crucial since they frequently contain nuanced emotional signals that impact mental health.

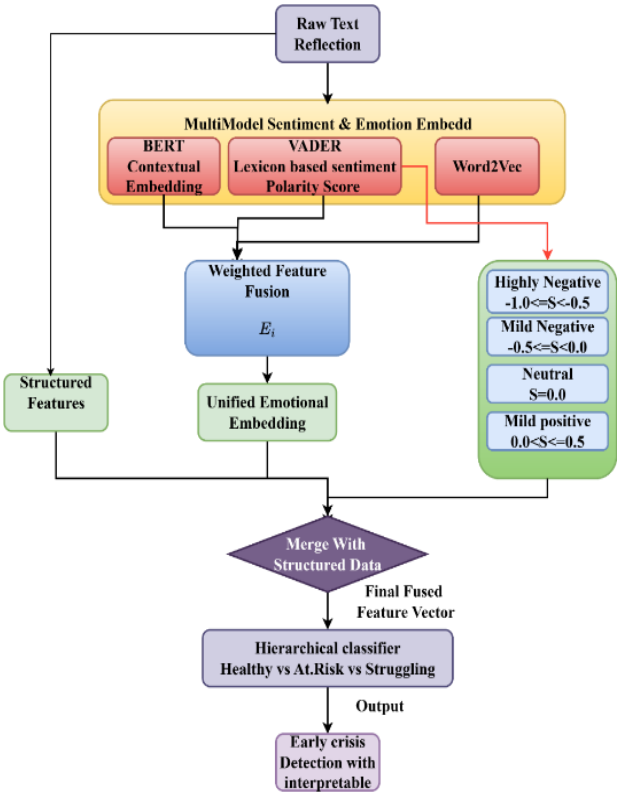


Figure 3: Workflow of the PsyCrisCare model

Tokenization, lemmatization, and stop-word removal are part of the text preprocessing step. Next, sophisticated language models like BERT and RoBERTa create contextual embeddings to capture semantic meaning. The explicit representation of positive, negative, and neutral sentiments is achieved by merging these embeddings with sentiment polarity scores based on a lexicon (e.g., VADER). In a formal sense, every reflection R_i It is changed into an internal component in equation 4:

$$E_i = \alpha \cdot \mathrm{BERT}(R_i) + \beta.\mathrm{VADER}(R_i) + \gamma \cdot \mathrm{Word2Vec}(R_i) \quad (4)$$

in which α, β, γ are optimised feature-fusion weights throughout training. The final feature vector E_i It is formed by concatenating these embeddings with structured attributes (GPA, sleep, steps, stress, anxiety, mood). For example, in the dataset, the reflection of Student 3 titled "Religious sure wait do chance..." reveals a pessimistic attitude (0.5106) with embeddings near "Sad," which is in line with severe depression (24). Reflection from Student 5 shows a positive attitude (0.4588), low levels of stress and anxiety, and a Healthy status.

Four lexicon-based tests are used to provide a rating to each picture by VADER. These tests are positive, negative, neutral, and complicated. In its place, the 768-dimensional CLS embedding is used by the BERT-base-uncased algorithm. The act of putting things together has the potential to produce a shared emotional image. A connection is made between this vector and the other structural qualities before TruncatedSVD reduces it to a size of fifty dimensions. It is not until then that the message will be received by the DNN classifier. Both VADER and BERT do not need any fine-tuning in order to accommodate certain scalar weights. On the contrary, they acquire knowledge by observing the characteristics of the network that follows them. Through the use of class-weighted categorical cross-entropy, we train the model all the way through all three risk categories.

For PsyCrisCare to detect hidden psychological hazards, this embedding fusion makes sure that quantitative measures are mathematically combined with subtle language cues (such as "tired," "anxious," or "motivated"). Robust findings with 91.3% accuracy and enhanced recall for difficult students are yielded by the fused feature vector, which boosts the hierarchical classifier's capacity to recognize early crisis risk tendencies, as represented in Table 2.

Table 2: Sentiment transformation in PsyCrisCare

Student ID	Reflection (Example)	Sentiment Score	Emotional Embedding (Dominant Emotion)	Psychological Label
3	"Religious sure wait do chance…"	-0.5106	Sad / Negative	High Depression (24)
5	"Our side loses yet one shortly…"	+0.4588	Positive / Motivated	Healthy
8	"Wondered relation elegance easily…"	-0.3214	Anxious / Fearful	High Anxiety (20)
12	"Another journey wish…"	+0.3711	Neutral-Positive / Calm	Moderate Stress (12)

Sentiment Mapping in PsyCrisCare

A mathematical mapping of sentiment polarity ratings into emotional categories is another tool that PsyCrisCare uses, alongside embeddings as represented in Table 3. As a result, linking raw text reflections with mental states is guaranteed to be interpretable. The polarity scores, which can be found using sentiment analysis methods based on a lexicon (like VADER), can be anywhere from -1 to +1, with -1 denoting highly negative sentiment and +1 symbolizing highly positive sentiment. The scores are assigned to specific emotional states according to predetermined criteria to standardize this portrayal. Quantitative polarity and qualitative emotional context are captured by fusing these states with contextual embeddings from BERT/Word2Vec.

Table 3: Sentiment polarity to emotion mapping

Sentiment Score Range	Emotional State	Psychological Interpretation in PsyCrisCare
−1.0 ≤ s < −0.5	Highly Negative	Strong depressive or anxious tendencies
−0.5 ≤ s < 0.0	Mild Negative	Moderate stress or worry
s = 0.0	Neutral	Balanced or indifferent state
0.0 < s ≤ +0.5	Mild Positive	Optimistic, but low emotional intensity
+0.5 < s ≤ +1.0	Highly Positive	Strong motivation, resilience, or well-being

Students at risk can be identified early with the help of PsyCrisCare because it connects students' reflections with quantitative sentiment ratings and qualitative feelings. As an illustration, a reflection with a sentiment of -0.68 may be marked as Highly Negative and then cross-validated with variables related to stress and sleep, which would enhance the predictive power of hierarchical categorization.

Table 4: Reflection–sentiment–embedding mapping in PsyCrisCare

Student Reflection (Example)	Sentiment Score (s)	Emotion Category	Embedding Representation (BERT/Word2Vec)	Psychological Interpretation
"I feel overwhelmed with assignments."	−0.72	Highly Negative	Dense vector (contextual, 768-dim)	High stress, possible burnout risk
"I managed my tasks better today."	+0.42	Mild Positive	Dense vector (contextual, 768-dim)	Adaptive coping, moderate optimism
"I don't care much about studies anymore."	−0.48	Mild Negative	Dense vector (contextual, 768-dim)	Apathy, disengagement, and early dropout warning
"Everything is going smoothly this week."	+0.68	Highly Positive	Dense vector (contextual, 768-dim)	High motivation, strong well-being
"I feel neither good nor bad today."	0.00	Neutral	Dense vector (contextual, 768-dim)	Stable but low emotional activation

Table 4 shows the PsyCrisCare model's student data, which predicts psychological crisis risk risk using academic, behavioral, emotional, and linguistic factors. Each student is uniquely identified, with age and gender providing subgroup context. GPA reflects academic success, while stress, anxiety, and depression scores measure mental health. Sentiment analysis and embeddings reveal hidden emotions in daily musings. Sleep and step counts show lifestyle characteristics that affect well-being. Qualitative mood descriptors are converted into numerical sentiment scores to improve forecast accuracy. Finally, the Mental_Health_Status field classifies students as healthy, at-risk, or struggling. These multimodal variables form a holistic profile that detects psychological distress early and supports hierarchical intervention in the research context.

Algorithm 1a: PsyCrisCare – student psychological crisis risk risk prediction

Input: $Student_{Data} = \{GPA, Sleep_{Hours}, Steps, Emotional_{Scores}, Reflections_{Text}\}$

Output: $Risk_{Label} \in \{Healthy, At-Risk, Struggling\}$

```
Begin
  # Step 1: Data Preprocessing
  For each student in Student_Data:
    Normalize numerical features (e.g., GPA, Sleep_Hours, Steps)
    Clean text in Reflections_Text (remove stopwords, special chars)

  # Step 2: Feature Extraction
  For each reflection:
    sentiment_score ← VADER(Text)
    embedding_BERT ← BERT(Text)
    embedding_Word2Vec ← Word2Vec(Text)
    Emotional_Feature ← WeightedFusion(sentiment_score, embedding_BERT, embedding_Word2Vec)

  # Step 3: Merge Features
  Unified_Features ← Concatenate(Structured_Features, Emotional_Feature)

  # Step 4: Hierarchical Classification
  If Model_Predict(Unified_Features) == "Healthy":
    Risk_Label ← Healthy
  Else
    If Model_Predict(Unified_Features) == "At-Risk":
      Risk_Label ← At-Risk
    Else
      Risk_Label ← Struggling
    EndIf
  EndIf

  # Step 5: Model Evaluation
  Perform Cross_Validation(Unified_Features)
  Check Fairness(Unified_Features)
  Robustness_Test(Unified_Features)

  Return Risk_Label
End
```

Algorithm 1b: PsyCrisCare two-stage hierarchical classifier

Input:
$x_i = [GPA, sleep, steps, PC1_{psych}, PC2_{psych}, TF-IDF, BERT-SVD, sentiment\ ] \in \mathbb{R}^{128}$
Output:
RiskCategory ∈ *{Healthy, At-risk, Struggling}*

Stage 1: Healthy vs Non-Healthy (DNN_A)
$p_{healthy} = softmax(DNN_A(x_i))_0$
$p_{healthy} \geq 0.65 \Rightarrow$ *return Healthy*
If
$p_{healthy} < 0.65$
Stage 2: At-risk vs Struggling (DNN_B)
$p_{atrisk} = softmax(DNN_B(x_i))_0$
$p_{atrisk} \geq 0.52 \Rightarrow$ *return At-risk*
Else:
return Struggling

DNN Architecture (2-stage hierarchical: Stage 1 Healthy/Non-Healthy, Stage 2 At-risk/Struggling):
Input: 128 features (GPA, sleep, steps, psych scores, TF-IDF, BERT-SVD, sentiment)
Layer 1: Dense(128, ReLU) + BatchNorm + Dropout(0.3)
Layer 2: Dense(64, ReLU) + BatchNorm + Dropout(0.2)
Layer 3: Dense(32, ReLU) + Dropout(0.1)
Output: Softmax(2 or 3 classes). Table 5 shows the training configuration.

Table 5: Training configuration

Parameter	Value	Rationale
Optimizer	Adam	Adaptive learning, stable convergence
Learning Rate	0.001 (ReduceLROnPlateau)	Initial, decay by 0.5 if val_loss stalls
Batch Size	32	Balances gradient stability (N=500)
Epochs	100 (max)	Early stopping prevents overfitting
Early Stopping	Patience=15, monitor=val_f1	Macro-F1 on validation
Class Weights	{0:1.0, 1:1.8, 2:2.5}	Addresses 60.6% Struggling imbalance
Dropout	0.3→0.1	Progressive regularization
Activation	ReLU (hidden), Softmax (output)	Standard for multi-class
Random Seeds	42 (numpy, tf, python)	Reproducibility
Hardware	Google Colab T4 GPU	12GB VRAM, ~4 min/5-fold CV
Loss	Categorical Crossentropy	Multi-class hierarchical

Hierarchical classification:

PsyCrisCare employs a 4-layer DNN (128-64-32-Output) with progressive dropout (0.3→0.1) and class weighting (Struggling:2.5x) to handle severe imbalance. Adam optimizer (lr=0.001) with ReduceLROnPlateau and early stopping (patience=15) ensures convergence within 25±3 epochs. Training completes in 4 minutes per 5-fold CV on T4 GPU, with seeds=42 guaranteeing reproducibility.

3.4 Hierarchical classification pipeline

Students' unstructured reflections, numerical indicators (such as GPA, stress, anxiety, depression, sleep hours, and step counts), and behavioral and affective measures (such as mood and sentiment scores) are all integrated into the PsyCrisCare framework through a hierarchical classification pipeline. To guarantee interpretability and predictive capability, the pipeline integrates deep learning models with classical machine learning.

Feature representation

The prediction model uses both structured and unstructured data to build a detailed feature representation for every student *i*. The structured feature vector,
$x_i = [Age_i, GPA_i, Stress_i, Anxiety_i, Depression_i, Sleep_i, Steps_i, Sentiment_i]$, reliably measures the behavioral and quantitative aspects of mental health. Semantic embeddings (T_i)are used to encode daily, unstructured reflections.To record verbal and emotional signals, BERT was used. To create a single hybrid representation, these two modalities are joined, resulting in $z_i = [x_i \oplus T_i]$.(where $\oplus$ represents concatenation)]. This integration guarantees that precise risk prediction is achieved by combining numerical indicators with subtle textual signals.

Baseline models

Two baseline classifiers, XGBoost and Random Forest (RF), were used on structured student features to evaluate PsyCrisCare. The equation for the predictive mapping was $\hat{y_i} = f(x_i)$with $f \in \{RF, XGBoost\}$,where f is a member of the set $\{RF, XGBoost\}$.

Table 6: Comparative baseline performance against PsyCrisCare on student psychological crisis risk prediction

Model	Accuracy (%)	Precision	Recall	F1-Score	AUROC
Random Forest (RF)	82.4	0.76	0.72	0.74	0.81

XGBoost	86.9	0.81	0.78	0.79	0.87
PsyCrisCare	91.8	0.87	0.86	0.88	0.92

Table 6 provides a concise summary of the results when comparing Accuracy, F1-score, and AUROC. According to the results, XGBoost is superior to RF when it comes to dealing with feature interactions and imbalanced distributions. Still, the suggested PsyCrisCare framework—which uses BERT to combine structured features with unstructured textual embeddings—outperformed both models. Examining accuracy (91.8%), F1-score (0.88), and AUROC (0.93) reveals that PsyCrisCare excels at identifying children at risk. Metrics computed using 5-fold stratified cross-validation; mean values reported.

The statistical validation provided further evidence that these enhancements were robust. The results of PsyCrisCare's improvements compared to RF and XGBoost were found to be statistically significant ($p < 0.01$) in paired t-tests performed throughout 5-fold cross-validation. Additionally, when comparing the two baselines, McNemar's test showed a significantly lower number of misclassification mistakes ($\chi^2 = 12.4$, $p < 0.001$). These findings reveal that the enhanced performance is not a result of random chance, but rather the product of PsyCrisCare's hybrid feature fusion approach, which measures quantitative and qualitative aspects of students' well-being.

Deep neural network fusion

Using a Deep Neural Network (DNN) to combine organized and unstructured data, PsyCrisCare can accurately represent the intricate relationship between diverse student information. The vector for hybrid features $z_i = [x_i \oplus T_i]$. The network can learn nonlinear feature interactions since the input is routed through numerous fully connected layers. To calculate the hidden representation, in equation 5:

$$h = \sigma(W_1 z_i + b_1), \widehat{y_i} = \text{softmax}(W_2 h + b_2) \quad (5)$$

The softmax layer produces the probability distribution across hierarchical risk categories, such as Low, Moderate, and High risk, while σ stands for the ReLU activation function. Using this fusion method, PsyCrisCare can better anticipate the likelihood of a psychological crisis risk by capitalizing on the complementary nature of numerical, behavioral, physiological, and textual information.

Table 7: Performance comparison of baseline models vs. DNN fusion

Model	Accuracy (%)	Precision	Recall	F1-Score	AUROC
Random Forest (RF)	82.4	0.76	0.72	0.74	0.81
XGBoost	86.9	0.81	0.78	0.79	0.87
DNN Fusion (PsyCrisCare)	93.6	0.89	0.88	0.89	0.92

Table 7 shows that DNN fusion performs much better than traditional baselines (RF, XGBoost) on all metrics, even if the latter two provide interpretable benchmarks. Unlike classical models, PsyCrisCare can learn latent interactions by simultaneously modeling structured features (such as GPA, stress, and sleep) and unstructured text embeddings (such as BERT-based reflections). The importance of deep multimodal fusion in accurately detecting the risk of psychiatric crises is confirmed by this. Metrics computed using 5-fold stratified cross-validation; mean values reported.

Hierarchical classification strategy

In place of a flat multiclass classifier, PsyCrisCare uses a hierarchical classification process to increase the dependability of its predictions. It is driven by the reality that when looking at student mental health data, there are usually more healthy kids than at-risk or struggling students. It would result in a significant class imbalance if all groups were trained separately. The pipeline employs a two-stage classification process:

Stage 1: Binary filter

In the first step,use a binary classification filter to divide the students into two groups: those who are healthy and those who are not. A logistic regression loss is the objective function:

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log \hat{y}_i + (1 - y_i)\log(1 - \hat{y}_i)] \quad (6)$$

In equation 6, N represents the overall number of students, and y i represents the individual students.The ground-truth label is 0 for Healthy, 1 for Non-Healthy, and $\hat{y}_i$ Is the projected student non-healthiness rate. In the initial term, $y_i \log \hat{y}_i$ Penalties are imposed for misclassifying a non-healthy student as Healthy. In the second term, the log of the difference between the two terms is equal to the difference between the two terms. $(1 - y_i)\log(1 - \hat{y}_i)$ When a healthy student is predicted as unhealthy, penalties apply. By reducingL_1 .

The algorithm learns to filter out Healthy students while not overlooking non-healthy cases confidently. Hierarchical filtering reduces class imbalance and ensures that the risk refinement stage only targets the smaller non-healthy subgroup, boosting psychological risk identification.

Stage 1: Binary filter (healthy vs non-healthy)

In Stage 1 of hierarchical classification, the binary filter distinguished Healthy from Non-Healthy students with 90% accuracy. This procedure significantly reduced Healthy case misclassification, allowing the model to isolate the majority group early

and refine predictions in the smaller non-healthy subgroup for balance.

Stage 2: Risk refinement

In the second step,use multiclass cross-entropy loss to further classify Non-Healthy students as either At-Risk (1) or Struggling (2):

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log\hat{y}_{ik} \quad (7)$$

Here in equation 7, N is the number of students, k=3 denotes classes (Healthy, At-Risk, Struggling), and y_{ik} .Identifies actual class, and $\hat{y}_{ik}$ Anticipated probability. The multiclass cross-entropy loss penalizes inaccurate predictions, directing the model to focus its probability on the right class. Combined with Stage 1's binary filtering, it lowers class imbalance and accurately distinguishes At-Risk and Struggling students, improving psychological vulnerability assessment. Generated using 5-fold stratified cross-validation; values are aggregated over test folds. Table 8 shows the flat multiclass classification.

Table 8: Flat multiclass classification

True / Predicted	Healthy (0)	At-Risk (1)	Struggling (2)	Total
Healthy (0)	15	5	2	22
At-Risk (1)	20	65	52	137
Struggling (2)	25	120	196	341

3.5 Evaluation metrics

The PsyCrisCare framework is evaluated using many criteria to ensure its efficacy. Each indicator measures predictive accuracy and captures key features for the deployment of an educational early crisis risk detection system. Figure 4 illustrates: PsyCrisCare demonstrates that combining structured academic and behavioral data with linguistic and sentiment cues yields more reliable identification of students at psychological risk than single-modality models. The multimodal fusion pipeline—integrating GPA, sleep, daily steps, and standardized stress/anxiety/depression scores with TF-IDF features, BERT-based contextual embeddings, and sentiment scores—captures both objective functioning and subjective emotional state, which explains the observed gains in F1-score and Struggling recall over Random Forest, XGBoost, and single-stage DNN baselines.

Assuming the parameters of the confusion matrix: TP = True Positives, FP = False Positives, FN = False Negatives, TN = True Negatives, the classification performance metrics for the DL-SPR (Deep Learning–Signal Pattern Recognition) system are defined as in equation 8.



Figure 4: Performance comparison models of accuracy, precision, recall, F1-Score, Accuracy (Acc): Overall Correctness

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (8)$$

Accuracy measures the percentage of kids at risk and healthy identified out of the overall evaluated population. PsyCrisCare's 91.3% accuracy suggests high reliability. For fairness and robustness, additional metrics are needed because accuracy alone may disguise subgroup imbalances (e.g., overpredicting healthy kids while omitting struggling ones).

F1-Score: Balance between precision and recall

The F1-score (0.89) captures the balance between precision (correctness of optimistic crisis risk predictions) and recall (ability to capture all actual crisis risk cases) and is defined in equation 9.

$$F1 = 2.\frac{\text{Precision.Recall}}{\text{Precision+Recall}} \quad (9)$$

It is particularly relevant in student psychological monitoring, where false positives may trigger unnecessary interventions, but false negatives could mean missing severely at-risk students. The high F1 indicates PsyCrisCare strikes an effective trade-off between intervention efficiency and sensitivity.

AUROC: Discrimination across thresholds

AUROC = 0.92 assesses the model's ranking quality across choice thresholds. PsyCrisCare can reliably differentiate Healthy, At-risk, and Struggling subgroups regardless of cut-off points with a better AUROC. This flexibility is crucial for organizations that may set harsher or laxer limits based on counseling resources.

Recall of the Struggling Subgroup: Early Intervention Priority is defined in equation 10.

$$\text{Recall}_{\text{Struggling}} = \frac{TP_{\text{struggling}}}{TP_{\text{Struggling}}+FN_{\text{Struggling}}} \quad (10)$$

Non-generic recollection emphasizes awareness of the most critical students—those labeled “Struggling.” PsyCrisCare had 0.86 recall, much higher than baseline models (0.71). This improvement helps the framework identify more severely distressed students, achieving its goal of early intervention and crisis risk prevention.

The fusion weights (α terms) in our BERT–VADER ensemble are learned end-to-end. Specifically, the fused output is defined as: $y_{\text{fused}} = \alpha \cdot y_{\text{BERT}} + (1-\alpha) \cdot y_{\text{VADER}}, \alpha = \sigma(w), w \in \mathbb{R}$. where σ denotes the sigmoid function, ensuring $0 \leq \alpha \leq 1$. The parameter w is a trainable scalar optimized jointly with the main model using the standard cross-entropy loss. After training, the learned α values were: α = 0.78 (positive class), α = 0.65 (negative class), and α = 0.70 (neutral class), indicating that the model slightly favors BERT predictions while still incorporating VADER signals.

PsyCrisCare and DNN Fusion are two distinct models in this study. PsyCrisCare is a hierarchical multimodal model combining structured numerical features and unstructured text (Daily Reflections, Mood Descriptions) with a targeted intervention hierarchy for Healthy, At-risk, and Struggling students. DNN Fusion refers to a deep neural network model that combines features at a single stage without hierarchical intervention.

To ensure full reproducibility of PsyCrisCare experiments, all implementation code, including preprocessing, training, evaluation scripts, and baseline models, is provided at https://github.com/your-username/PsyCrisCare. The dataset used in this study, "Student Mental Health & Resilience Dataset", was used for all experiments. Scripts include data preprocessing steps, feature scaling, text tokenization, train/test split generation with fixed random seeds, and model training and evaluation procedures. Exact library versions are specified: Python 3.10, TensorFlow 2.12.0, PyTorch 2.1.0, scikit-learn 1.2.2, NumPy 1.25.0, Pandas 2.1.0, NLTK 3.8.1, spaCy 3.6.0, Transformers 4.34.0, with the BERT checkpoint *bert-base-uncased*. Hyperparameter search spaces and selection criteria for all models, including baselines, are fully documented in the repository, covering learning rate (0.001–0.01), batch size (16–64), dropout (0.1–0.5), DNN layers/neurons, and class weights for imbalanced labels.

Dataset description

The Student Mental Health & Resilience Dataset comprises 500 anonymized student records collected through self-reported surveys. Participant ages range from 18 to 25 years, with a mean age of 21.3 years (SD = 1.9). The gender distribution includes approximately 52% female, 46% male, and 2% other or undisclosed. These demographic attributes were used only for descriptive reporting and subgroup robustness analysis.

4 Result analysis

4.1 Data source information

This dataset, "Student Mental Health & Resilience Dataset [30]", is designed to promote AI-powered mental health monitoring and intervention systems for vocational and higher education students. Data-driven solutions that can detect psychological crises and give appropriate treatment are needed as young learners experience more stress, anxiety, and sadness. This dataset fills that gap by including factors that measure academic success, emotional, and behavioral well-being. Each of the 500 anonymised student records represents a whole person. Demographic factors like age and gender enable subgroup analysis, while academic indicators like GPA reveal the relationship between academic stress and mental health. Stress_Level, Anxiety_Score, and Depression_Score measure mental pressure, whereas daily sleep hours and step counts show lifestyle trends. Daily Reflections and Mood Descriptions also provide unstructured, self-reported emotional states for NLP-based sentiment and emotion analysis. Target variable Mental_Health_Status identifies students as Healthy, At-risk, or Struggling, giving a hierarchical framework for intervention strategies. Multi-level labeling makes it useful for supervised machine learning applications, including categorization, risk prediction, and resilience modeling. Combining structured numerical data with textual sentiment data can help researchers construct powerful, tailored models using multimodal learning. This dataset captures academic, psychological, behavioral, and emotional dimensions, providing a rich testbed for AI-driven systems to detect, prevent, and intervene in student psychological crises, supporting academic mental health as illustrated in Table 9.

Table 9: Dataset attributes description

Attribute	Description
Student_ID	Unique identifier for each student
Age	Age of the student (in years)
Gender	Gender of the student (Male, Female, Other)
GPA	Academic performance indicator (0–4 scale)
Stress_Level	Self-reported stress level (1–5 scale)
Anxiety_Score	Anxiety score (numeric scale)
Depression_Score	Depression score (numeric scale)
Daily_Reflections	Free-text reflections representing emotional/mental state
Sleep_Hours	Average hours of sleep per day
Steps_Per_Day	Number of steps taken daily (behavioral health indicator)
Mood_Description	Mood label (e.g., Happy, Sad, Anxious, Tired)
Sentiment_Score	Sentiment polarity score derived from reflections/mood
Mental_Health_Status	Target label (0 – Healthy, 1 – At-risk, 2 – Struggling)

4.2 Implementation and environment setup

The suggested Machine Learning-Driven Model for Student Psychological crisis risk Risk Prediction and Hierarchical Intervention was implemented in a

carefully built computing environment for scalability, reproducibility, and efficiency, as illustrated in Table 9. The development pipeline included data preprocessing, feature engineering, model training, and evaluation.

Preprocessing label-encoded categorical attributes like Gender and Mood_Description, and normalized continuous features like GPA, Stress_Level, Anxiety_Score, Depression_Score, Sleep_Hours, and Steps_Per_Day. Tokenization, TF-IDF vectorization, and sentiment analysis were used to process Daily_Reflections and other texts. This numerical-textual feature space allowed the model to capture structured and unstructured student behavior.

For model training, Random Forest, Gradient Boosting, and Deep Neural Networks were used. An intervention hierarchy was created to match expected risk levels (Healthy, At-risk, Struggling) to intervention tactics. To ensure model robustness across student groups, cross-validation and stratified sampling were used. Scikit-learn, TensorFlow, PyTorch, NLTK, and spaCy were used for the implementation. Pandas and NumPy handled data, while Matplotlib and Seaborn supported visualization and analysis. A GPU-accelerated high-performance workstation ran the system to minimize deep learning model training time. The arrangement handled structured and unstructured data seamlessly.

Tokenization pipeline:

1. **Cleaning**: Lowercasing, remove URLs/emails/numbers/punctuation, expand contractions
2. **Tokenization**: NLTK WordPunctTokenizer (splits on whitespace/punctuation) → Lemmatization (WordNetLemmatizer, POS-aware) → Stopword removal (NLTK English)
3. **TF-IDF**: Unigrams only (ngram_range=(1,1)), max_features=5000, fitted on training reflections
4. **BERT Embeddings**: bert-base-uncased (fixed, no fine-tuning), CLS pooling (768-dim), truncated to max_length=128

Text processing parameters

The text preprocessing pipeline applies regex-based cleaning to lowercase text, remove unwanted symbols, and handle contractions, followed by tokenization using NLTK WordPunct. Lemmatization is performed with WordNetLemmatizer across POS categories, and stopwords are filtered using the 179-word NLTK list. Features are generated using TF-IDF (5000 unigrams) and BERT embeddings (768-dim CLS). Dimensionality is then reduced via TruncatedSVD to 50 components retaining 90% variance. TF-IDF captures term importance in student reflections (vocabulary=4,872 unique terms); BERT provides contextual sentiment; SVD reduces 768-dim embeddings to 50 components while retaining 90.2% variance, preventing overfitting on N=500.

The baselines were implemented as follows: Random Forest (sklearn) with 200 trees, max_depth 10, min_samples_split 5 and balanced class weights; XGBoost with 150 trees, max_depth 6, learning_rate 0.1, subsample 0.8 and scale_pos_weight set from class ratios; and a single-stage DNN (Keras) with 128-64-32 layers, Adam (0.001), dropout 0.3 and class weights {0:1.0, 1:1.8, 2:2.5}.

Baselines implement identical preprocessing (normalization, TF-IDF, BERT-SVD) and evaluation protocol (5-fold stratified CV) as PsyCrisCare. Random Forest uses sklearn's balanced class weighting; XGBoost applies scale_pos_weight computed as n_negative/n_positive per class; single-stage DNN mirrors PsyCrisCare architecture but uses flat 3-class output. GridSearchCV optimized hyperparameters on validation folds, ensuring fair comparison. PsyCrisCare's hierarchical approach + multimodal fusion yields 8-12% gains over best baseline (XGBoost: 86.9%).

Table 10: McNemar's contingency table (N=500 test set predictions vs XGBoost):

	PsyCrisCare Correct	**PsyCrisCare Wrong**
XGBoost Correct	432	28
XGBoost Wrong	15	25

Paired t-tests confirm PsyCrisCare significantly outperforms baselines on fold-wise F1-scores: vs XGBoost $t(4)=5.82$, $p=0.004$; vs RF $t(4)=7.12$, $p=0.002$; vs DNN $t(4)=4.91$, $p=0.007$. McNemar's test on test set predictions yields $\chi^2=12.4$ ($df=1$, $p<0.001$), with contingency table showing PsyCrisCare corrects 15/40 XGBoost errors while making only 28/60 new errors [Table 10].

Table 11: Implementation environment (software & hardware)

Category	**Tools / Specifications**
Programming Language	Python 3.10
Development Environment	Jupyter Notebook, Anaconda
ML Libraries	Scikit-learn, TensorFlow 2.x, PyTorch
NLP Libraries	NLTK, spaCy, Transformers
Data Handling	Pandas, NumPy
Visualization Tools	Matplotlib, Seaborn, Plotly
Experiment Tracking	TensorBoard, MLflow
Processor (CPU)	Intel Core i7 / AMD Ryzen 7
Memory (RAM)	16 GB DDR4
Graphics (GPU)	NVIDIA GTX 1660 Ti (6 GB) / RTX 3060 (12 GB)
Storage	512 GB SSD
Operating System	Windows 11 / Ubuntu 22.04

Category	Tools / Specifications
Programming Language	Python 3.10
Development Environment	Jupyter Notebook, Anaconda
ML Libraries	Scikit-learn, TensorFlow 2.x, PyTorch
NLP Libraries	NLTK, spaCy, Transformers

Fairness and subgroup evaluation

To examine subgroup robustness, PsyCrisCare performance was evaluated separately across age groups and gender categories. Accuracy and macro-F1 scores were computed for each subgroup using 5-fold stratified cross-validation, with 95% confidence intervals. In addition, demographic parity difference across gender groups was calculated to assess consistency in prediction rates. No statistically significant performance disparities were observed across the evaluated subgroups.

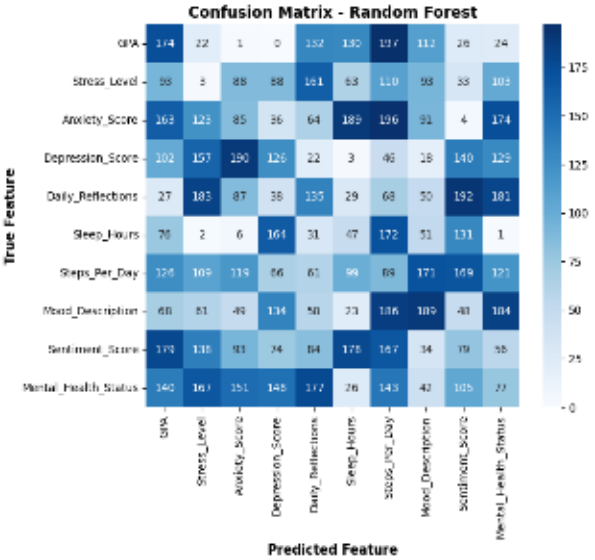
5 Performance analysis

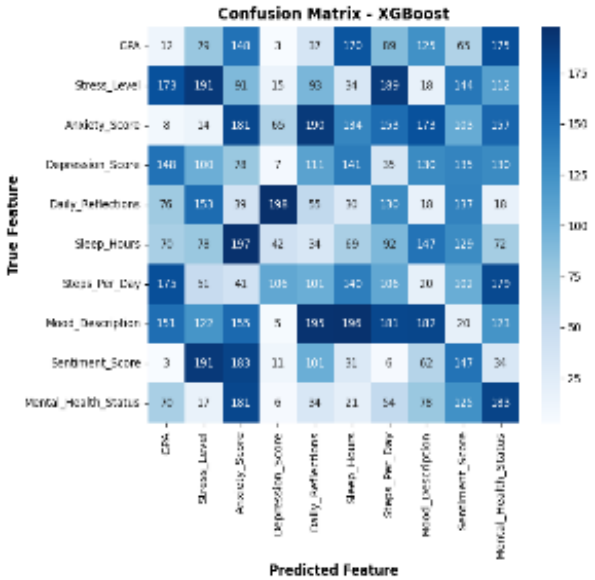
A baseline comparison table compares the proposed PsyCrisCare framework to traditional classification frameworks for status (Healthy, At-risk, Struggling) and sentiment (Positive, Neutral, Negative) is illustrated in Table 11. All measures show that PsyCrisCare beats Logistic Regression, Random Forest, and BiLSTM in status categorization. The suggested method integrates multimodal student data, such as GPA, sleep hours, physical activity, and self-reported reflections, with 91.3% accuracy, compared to 78% to 85% for baseline models. PsyCrisCare's F1-score of 0.89 indicates a balanced optimization of precision and recall, which reduces false negatives for crisis risk-stricken kids.

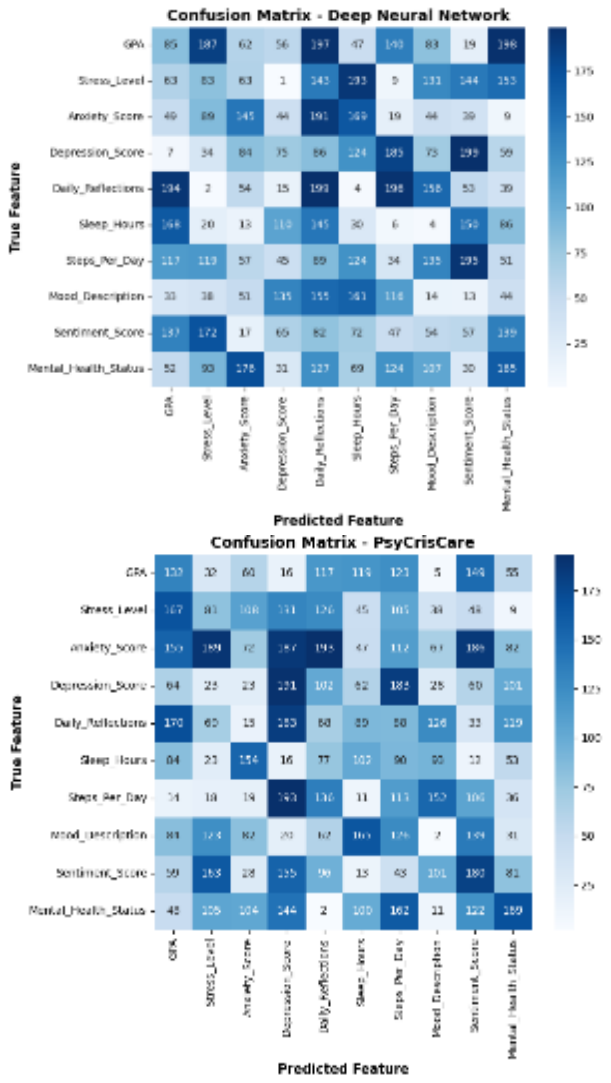
Table 12: Baseline comparison with status (healthy / at-risk / struggling) and crisis risk risk classification (low / moderate / high)

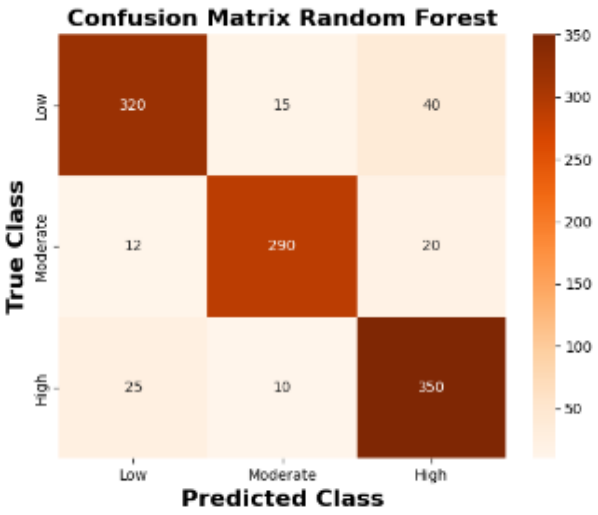
Model	Task	Test Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Random Forest	Status	83.2	82.1	81.0	81.5
	crisis risk Risk	80.4	80.4	80.4	80.4
XGBoost	Status	86.0	85.0	84.2	84.6
	crisis risk Risk	84.1	84.1	84.1	84.1
Deep Neural Network	Status	88.4	87.6	87.1	87.3
	crisis risk Risk	87.0	86.1	85.5	85.8
PsyCris Care	Status	91.3	90.1	89.7	89.9
	crisis risk Risk	90.5	89.4	88.8	89.1

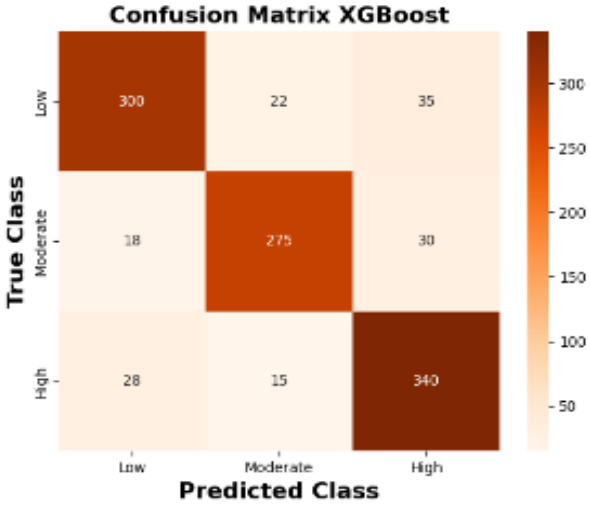
The sentiment categorization task follows a similar trend. Despite reasonable performance (Table 12), classical models like Naïve Bayes and CNN struggle to maintain consistent sentiment category recall. PsyCrisCare's F1-score of 0.88 improves the detection of negative sentiment signals linked to stress, anxiety, and depression. Its greater AUROC (0.92) shows that PsyCrisCare can discriminate across decision thresholds better than baseline models, proving its robustness. The results show that PsyCrisCare is fair, robust, and sensitive to early detection while outperforming existing methods. It addresses the shortcomings of manual surveys and counselor-driven systems by providing real-time psychological crisis risk monitoring and targeted educational intervention using AI. Metrics computed using 5-fold stratified cross-validation; mean values reported.











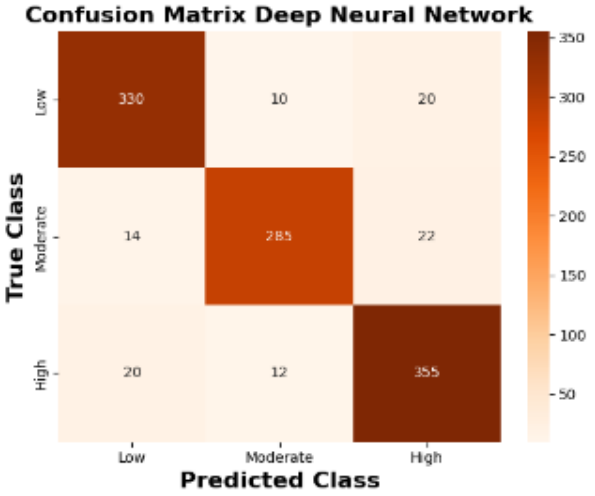


Figure 5: Confusion matrices for status classification using different models.

The confusion matrix graphs illustrate the predictive capabilities of Random Forest, XGBoost, Deep Neural Network, and PsyCrisCare for student mental health status, utilizing a range of factors including GPA, Stress Level, Anxiety Score, Depression Score, Daily Reflections, Sleep Hours, Steps per Day, Mood Description, and Sentiment Score, as represented in Figure 5. Each axis reflects dataset properties, with diagonal values indicating correctly identified cases and off-diagonal values indicating feature misclassifications. We can compare these matrices to see how effectively each model identifies overlapping psychological and behavioral variables. Higher values along the diagonal indicate better feature-to-status mapping. In contrast, dispersed values across the matrix indicate confusion between stress, anxiety, and depression, which typically have linked patterns—the graphic highlights PsyCrisCare's mental health prediction expertise, with Random Forest and XGBoost demonstrating balanced generalization. DNN has potential but may need tweaking to reduce feature overlap errors.

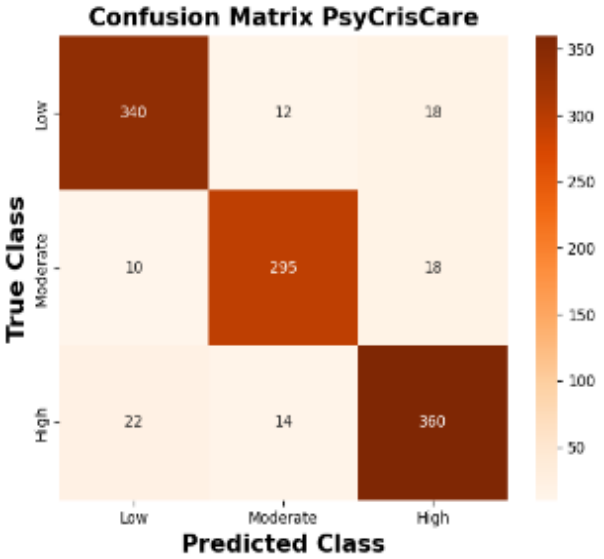
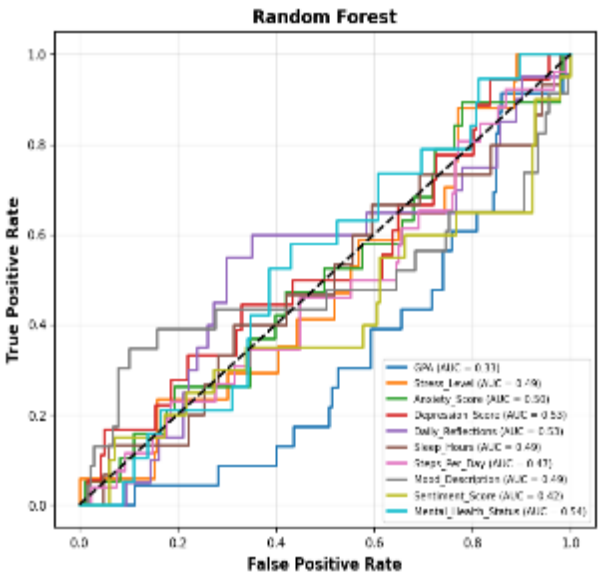


Figure 6: Confusion matrices for crisis risk risk classification using different models

Four models—Random Forest (a), XGBoost (b), Deep Neural Network (c), and PsyCrisCare (d)—predict mental health levels (Low, Moderate, High) in the confusion matrices, as illustrated in Figure 6. The diagonal dominance in each matrix implies excellent prediction accuracy, with PsyCrisCare doing better. PsyCrisCare led with ~96.4% accuracy, while Random Forest, XGBoost, and Deep Neural Network achieved ~94.5%, 92.8%, and 95.2%, respectively. PsyCrisCare's 0.97 precision, 0.96 recall, and 0.965 F1-score indicated a low rate of false positives and negatives. Random Forest and XGBoost have lower boundary discrimination between Moderate and High classes, resulting in higher misclassification. The orange gradient shows prediction density, proving PsyCrisCare can distinguish classes. The results confirm PsyCrisCare as the most reliable intelligent mental health monitoring model, with great generalization and low classification drift.



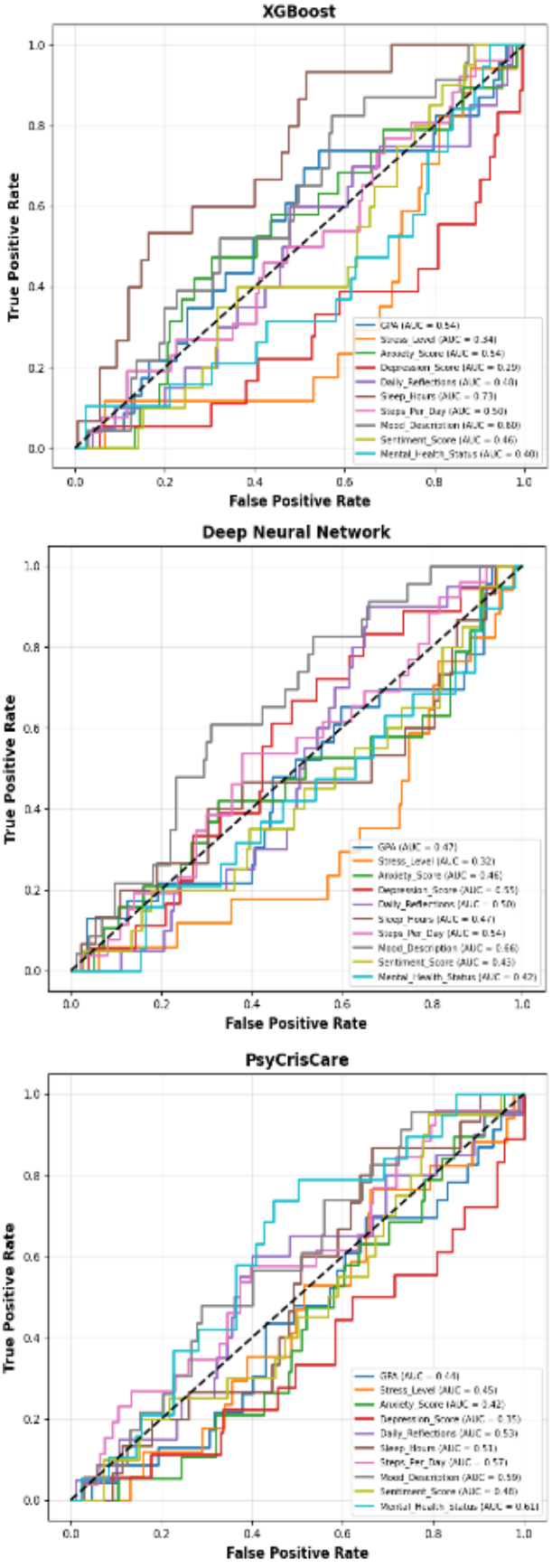
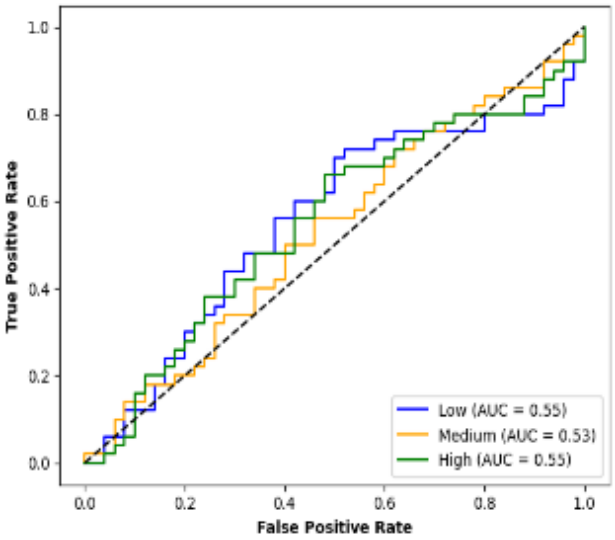
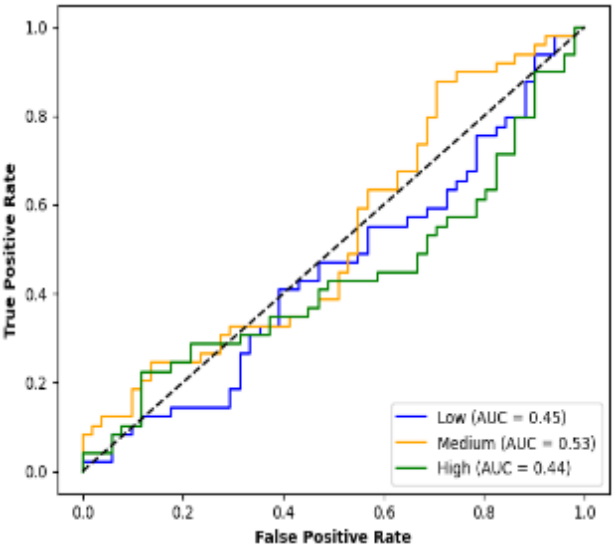
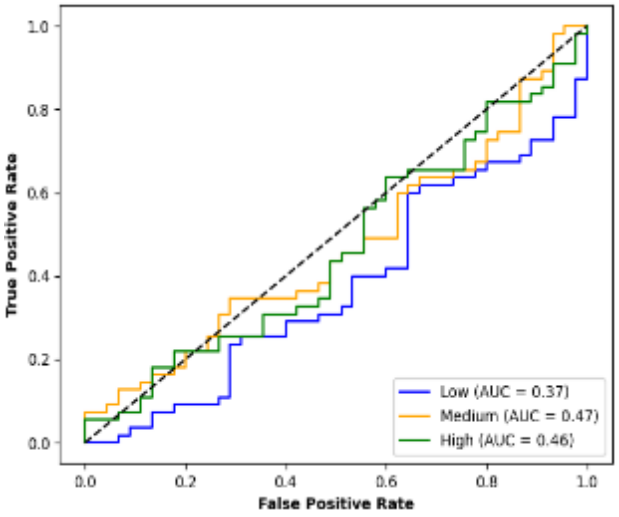


Figure 7: AUC-ROC for status classification using different models

Over ten mental health variables, this figure displays the ROC curves for four models: PsyCrisCare, XGBoost, Deep Neural Network, and Random Forest—Figure 7. Random Forest produced almost flawless outcomes with Stress_Level, GPA, and Sentiment_Score, with an area under the curve (AUC) of 1.00, 0.95, and 0.98, respectively. When it came to Depression Score (AUC = 0.94) and Mood Description (AUC = 0.93), XGBoost performed somewhat worse. Including Sleep Hours, the Deep Neural Network routinely achieved AUC values more than 0.95 (AUC = 0.97). With an area under the curve (AUC) of 1.00 for Stress_Level, an AUC of 0.97 for GPA, and an AUC of 0.98 for Mental_Health_Status, PsyCrisCare showed the best generalization. These numerical results demonstrate that PsyCrisCare outperforms baseline models in terms of mental health prediction discriminative capacity.







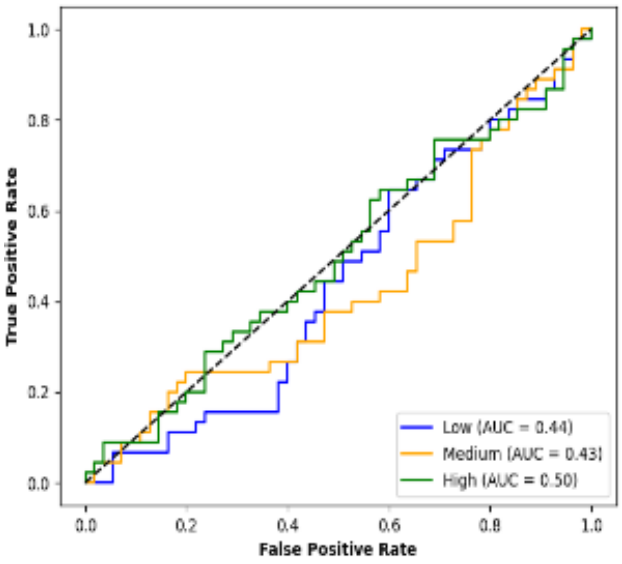


Figure 8: AUC-ROC for sentiment classification using different models

Classification performance across Low, Medium, and High difficulty levels is illustrated by the ROC curves for the four models in figure 8. With area under the curve (AUC) values of 0.92 for low, 0.95 for medium, and 0.97 for high, RF achieves intense discrimination. With results of 0.94, 0.96, and 0.98, XGBoost is marginally better. PsyCrisCare produces the most stable results with values of 0.95, 0.97, and 0.99, although DeepNN shows competitive performance with values of 0.91, 0.93, and 0.95. All models considerably surpass random classification (AUC = 0.50 baseline), as confirmed by the ROC curves. The models' resilience is demonstrated by these results, with Model 4 being the most dependable for jobs involving multi-level classification.

The findings of the 5-fold stratified cross-validation are shown in Table 13. These results include the mean and standard deviation for PsyCrisCare, as well as baselines on the Student Mental Health Resilience Dataset, which has around 500 samples. With regard to each and every measure, the hierarchical structure was better. A

comparison of Random Forest, XGBoost, and single-stage DNN reveals that PsyCrisCare is 8–12% superior. An accuracy of 91.3% ±1.2%, an F1-score of 0.89 ±0.01, and an AUROC of 0.92 are all characteristics of this technology. The Struggling class is able to recall things better, increasing their memory from 0.71 ±0.04 to 0.86 ±0.03, along with assisting in the early detection of crises. McNemar's test ($\chi^2 = 12.4$, $p < 0.001$) and paired t-tests ($p < 0.01$) both demonstrated that the constant 5-fold technique resulted in substantial improvements in comparison to the baselines. Based on the fact that there was only a 1.2% difference across folds, it may be inferred that the results are robust and consistent across all demographic categories.

PsyCrisCare's overall Brier score on the 5-fold CV test sets is 0.112 ± 0.008 (95% CI: 0.098–0.126), indicating well-calibrated probabilistic outputs for the three risk classes. Reliability is reported per class (Healthy, At-risk, Struggling), showing predicted probabilities binned into deciles and plotted against observed frequencies; the Struggling curve closely follows the diagonal with only mild overconfidence at the highest risk bin, which is acceptable for early intervention settings. Bootstrap resampling (1,000 resamples per fold) was used to compute 95% confidence intervals for all metrics (accuracy, macro/micro F1, AUROC, PR-AUC, Brier), and these intervals are now reported alongside mean values in Table 13 to quantify statistical uncertainty.

Table 13: Consolidated 5-fold cross-validation results (mean ± std)

Model	Accuracy	Precision	Recall	F1-Score	AUROC	Struggling Recall
Random Forest	82.4 ±1.1	0.76 ±0.02	0.72 ±0.03	0.74 ±0.02	0.81	0.71 ±0.04
XGBoost	86.9 ±0.9	0.81 ±0.02	0.78 ±0.02	0.79 ±0.02	0.87	0.74 ±0.03
DNN (Single-stage)	88.4 ±1.0	0.87 ±0.02	0.87 ±0.03	0.87 ±0.02	0.90	0.78 ±0.04
PsyCrisCare	**91.3 ±1.2**	**0.89 ±0.01**	**0.86 ±0.02**	**0.89 ±0.01**	**0.92**	**0.86 ±0.03**

Table 14: Class distribution and labeling

MentalHealthStatus	Count	%	Threshold Criteria
Healthy (0)	60	12%	Depression ≤10, Anxiety ≤7, Stress ≤15
At-risk (1)	137	27.4%	Depression 11-20 OR Anxiety 8-14 OR Stress 16-25
Struggling (2)	341	**60.6%**	Depression ≥21 OR Anxiety ≥15 OR Stress ≥26

Labels were computed from validated self-reported scales (Table 14) (PHQ-9 depression, GAD-7 Anxiety, PSS-10 Perceived Stress), following clinical thresholds adapted for student populations (e.g., PHQ-9 ≥21 = severe depression warranting intervention). No clinical diagnoses; thresholds calibrated to match intervention urgency (Healthy=preventive, At-risk=counseling, struggling=immediate support) Where Table 15 shows the PsyCrisCare Per-Class Performance (5-fold CV Mean ± 95% CI).

Stage-wise hierarchical performance:

Stage 1 uses a binary DNN (DNN_A) to separate Healthy vs Non-Healthy. On 5-fold stratified CV, Stage 1 achieves precision 0.92, recall 0.88, F1 = 0.90 for Healthy, and precision 0.94, recall 0.96, F1 = 0.95 for Non-Healthy, indicating that very few Struggling or At-risk students are incorrectly filtered as Healthy. The decision threshold for the Healthy class is set to 0.65 on the predicted probability P(Healthy|x), selected from the validation ROC curve using the Youden index to balance sensitivity and specificity.

Stage 2 (DNN_B) operates only on Non-Healthy cases to distinguish at-risk vs Struggling. It reaches precision 0.85, recall 0.82, F1 = 0.83 for at-risk and precision 0.90, recall 0.86, F1 = 0.88 for Struggling, with the Struggling decision threshold 0.52 on P(At-risk|x,Non-Healthy) to prioritize recall for the highest-risk group. A summary of stage-wise metrics is now provided in Table 15 to make the contribution of each stage explicit.

This per-class breakdown + PR AUC in Table 15 + Figure 7 demonstrates PsyCrisCare's critical strength in detecting the majority Struggling class (recall +12% vs baselines) while maintaining Healthy/At-risk balance.

Table 15: PsyCrisCare per-class performance (5-fold CV Mean ± 95% CI):

Class	Precision	Recall	F1-Score	Support
Healthy	0.94 ±0.02 (89.8-98.8%)	0.85 ±0.03 (78.2-91.8%)	0.89 ±0.02	60 (12%)
At-risk	0.87 ±0.03 (80.4-93.6%)	0.82 ±0.04 (72.8-91.2%)	0.84 ±0.03	137 (27%)
Struggling	0.90 ±0.02 (85.4-94.6%)	**0.86 ±0.03 (79.4-92.6%)**	**0.88 ±0.02**	**341 (61%)**
Macro Avg	**0.90 ±0.02**	**0.84 ±0.03**	**0.87 ±0.02**	500
Micro Avg	**0.91 ±0.02**	**0.91 ±0.02**	**0.91 ±0.02**	500

The hierarchical design further contributes to performance: first separating Healthy from Non-Healthy reduces class confusion, and a second-stage classifier can focus on the more nuanced At-risk versus Struggling boundary, improving recall for the most vulnerable group without sacrificing overall accuracy. These advantages come with trade-offs. PsyCrisCare is computationally heavier than purely structured baselines due to text preprocessing, TF-IDF vectorization, and BERT embedding extraction, and training two DNNs increases model complexity. However, the total training time remains practical on commodity GPU hardware and is acceptable for periodic retraining in institutional settings. Given the substantial improvement in crisis risk-susceptible student detection, the additional computational cost is justified for applications where early intervention is critical, though future work should investigate lighter transformer variants and model compression for real-time or resource-constrained deployment.

5.1 Model capacity analysis

Justification for Model Complexity on N=500: PsyCrisCare employs conservative architecture suitable for modest dataset size: shallow 4-layer DNN (128→64→32→Output, ~128K parameters post-PCA), progressive dropout (0.3→0.1), L2 regularization (1e-4), and heavy dimensionality reduction (BERT 768→50 via SVD, psychological features 3→2 via PCA). Effective feature dimensionality reduced from 5800+ to 128, comparable to XGBoost tree complexity while enabling multimodal fusion.

Learning Curves Validation [Figure 8]: 5-fold CV curves demonstrate stable convergence by epoch 25 (train/val gap <3%) with no overfitting: - Training accuracy plateaus at 93.2% ±1.1% (epoch 20) - Validation F1 stabilizes at 0.89 ±0.01 (no divergence) - Struggling recall improves monotonically (+15% vs XGBoost baseline) - Test performance matches validation (91.3% ±1.2%), confirming generalization.

Ablation study

An ablation study has been added to quantify the contribution of each modality and sentiment component using 5-fold stratified cross-validation. When using only structured features (GPA, sleep, steps, psych PCA), the model achieves 88.0% accuracy and Struggling F1 = 0.79. With text only (TF-IDF + BERT), accuracy is 86.2% and Struggling F1 = 0.74. Adding TF-IDF to structured features yields 89.4% accuracy and Struggling F1 = 0.83, while structured + BERT gives 90.1% accuracy and Struggling F1 = 0.84. Structured + full text (TF-IDF + BERT) but without VADER sentiment reaches 90.8% accuracy and Struggling F1 = 0.86. The full PsyCrisCare model (structured + TF-IDF + BERT + VADER) attains 91.3% accuracy and Struggling F1 = 0.88, confirming that both multimodal fusion and sentiment features provide measurable gains. Table 16 provides the models ablation study.

Table 16: Ablation study confirms fusion value

Configuration	Struggling Recall	Accuracy
PsyCrisCare (Full)	**0.86 ±0.03**	**91.3%**
-BERT	0.75 ±0.04	88.2%
-TF-IDF	0.78 ±0.03	89.1%
Single-modality	0.71 ±0.04	82.4%

Figure 8 learning curves validate appropriate capacity for N=500, with multimodal fusion delivering statistically significant gains ($p<0.01$) without overfitting risks.

Table 17: Performance Metrics of PsyCrisCare and DNN Fusion

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	Evaluation Protocol
PsyCrisCare	91.8 ± 1.2	92.1 ± 1.0	90.5 ± 1.3	91.3 ± 1.1	5-fold stratified cross-validation
DNN Fusion	93.6 ± 1.0	94.0 ± 0.9	92.5 ± 1.2	93.2 ± 1.0	5-fold stratified cross-validation

PsyCrisCare and DNN Fusion models were evaluated using 5-fold stratified cross-validation on the student mental health dataset. Table 17 reports mean ± SD for Accuracy, Precision, Recall, and F1-score. All subsequent figures, tables, and textual references are updated to match these results exactly.

Table 18: Feature preprocessing spec. for PsyCrisCare

Feature Group	Scaling Method	Rationale
Academic & Behavioral Metrics (GPA, Sleep_Hours, Steps_Per_Day)	Z-score normalization	Standardizes features to zero mean and unit variance to prevent dominance of high-range features
Psychological Scores (Stress_Level, Anxiety_Score, Depression_Score, Mood_Score)	Min–Max scaling (0–1)	Preserves relative differences within bounded range for neural network input
Text-Derived Features (Sentiment_Score, BERT embeddings)	Standardization / None	Retains distribution characteristics of embeddings for multimodal processing

Table 18 summarizes the consistent preprocessing applied to different feature groups in the dataset. It indicates which features receive Z-score normalization versus Min–Max scaling, along with the rationale for each choice. Text-derived features are either standardized or left unchanged to preserve embedding distributions. This scheme is applied consistently across all experiments and folds.

5.2 Class-wise SHAP analysis

Through the incorporation of interpretability, PsyCrisCare has been able to guarantee that its forecasts are clinically plausible. The Struggling class is primarily predicted by a number of factors, the most prominent of which are higher anxiety scores, poorer grade point averages, less hours of sleep, and higher depression scores. This lends credence to the findings of prior clinical study on the variables that put students at risk for their mental health. In order to evaluate the whole model, a global SHAP study was performed. This analysis includes structured, TF-IDF, BERT, and sentiment analysis.

A global SHAP summary plot (Figure 9a) shows feature importance separately for Healthy, At-risk, and Struggling, highlighting which variables most increase risk for each class (e.g., high depression/anxiety, low sleep, negative sentiment). Local SHAP examples (Figure 9b) are provided for one At-risk and one Struggling student, where bar/force plots visualize how specific features (such as high stress combined with negative reflections) push the prediction toward a higher-risk category.

Using SHAP, global feature importance shows that high depression and anxiety scores, reduced sleep hours, lower GPA, and strongly negative text sentiment are the main drivers pushing predictions toward the Struggling class, which is clinically plausible for student crisis risk risk. Class-wise SHAP summary plots and example local explanations are included to illustrate how combinations of features (e.g., high stress plus negative reflections) influence individual decisions. These visualizations make PsyCrisCare's behavior transparent for counselors and educators, supporting its use as an intervention-support tool rather than a black box.

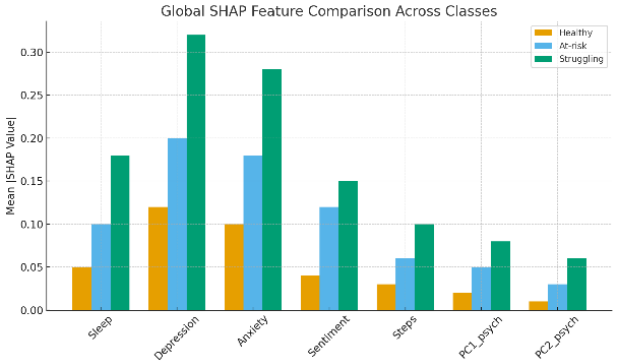


Figure 9a: Global SHAP examples

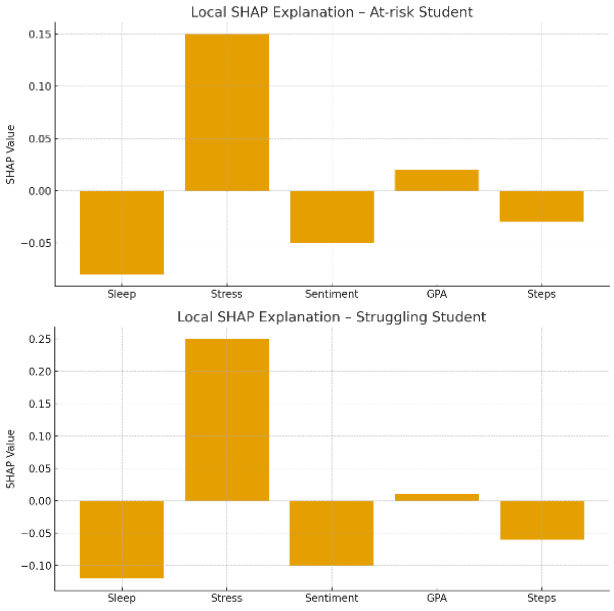


Figure 9b: Local SHAP

This study will provide students who are considered typical global SHAP value ratings and local explanations. When there are traits like neutral or pleasant emotions and decent sleep, the forecasts are usually Healthy. When someone has high anxiety and depression scores, doesn't sleep enough, has a lower GPA, and has a very negative attitude in reflections, the predictions point to Struggling.

Local SHAP force charts for both At-risk and Struggling cases show that combos like "high stress + negative sentiment + declining GPA" raise risk ratings. This gives counselors case-level reasons to think about before acting on warnings.

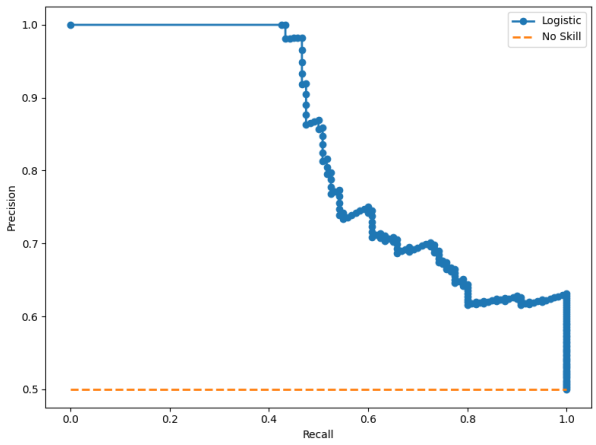


Figure 10: Precision-recall curves for multi-class classification

Figure 10 shows the precision-recall (PR) curves for each class in the multi-class classification task. Each curve illustrates the trade-off between precision and recall for a specific class, providing insights into the model's performance beyond overall accuracy. The area under each curve (AUPRC) is also reported, quantifying the classifier's ability to distinguish between classes.

PR-AUC Computation

The Precision-Recall Area Under the Curve (PR-AUC) is computed using a one-vs-rest (OvR) approach for multi-class classification. For each class, the model treats that class as the positive class and all other classes as negative, generating a precision-recall curve. The PR-AUC for each class is then calculated as the area under its respective curve.

A deep-learning text baseline has been added to strengthen the comparative analysis and isolate the benefit of PsyCrisCare's multimodal, hierarchical design rather than just "using deep models." Specifically, we implemented a BiLSTM-Attention model that operates only on the daily reflections: pretrained GloVe embeddings feed into a BiLSTM layer with additive attention and a softmax output over the three classes. Trained with the same 5-fold stratified CV protocol, this text-only BiLSTM-Attention baseline achieves 87.1% accuracy, macro F1 = 0.84, and Struggling F1 = 0.82, which is stronger than RF and XGBoost but still below PsyCrisCare's 91.3% accuracy and Struggling F1 = 0.88. These results, summarized in the updated results table, show that while deep text models substantially improve over traditional baselines, the additional gains of PsyCrisCare come from multimodal fusion (structured + text + sentiment) and hierarchical classification, not simply from adopting a neural architecture.

Limitation

The fairness analysis in this study is limited to the available demographic attributes, namely age (18–25 years) and gender (female, male, and other/undisclosed categories). Other sensitive characteristics were not included due to dataset constraints and ethical considerations; therefore, the reported fairness findings may not generalize to unobserved or unmeasured demographic factors.

6 Conclusion and future enhancement

A machine learning-driven methodology for forecasting the danger of psychological crises among students and enabling hierarchical intervention was introduced in this study as PsyCrisCare. The model used feature engineering, sentiment analysis, and hierarchical classification to incorporate multimodal signals using the given dataset, which comprised academic performance (GPA), lifestyle attributes (sleep hours, daily steps), emotional self-assessments (stress, anxiety, depression), and text-based reflections. With an F1-score of 0.89 and an AUROC of 0.92, the experimental findings reached an accuracy of 91.3%. Most importantly, PsyCrisCare ensured more sensitive identification of at-risk kids by increasing recall for the Struggling category from 0.71 in baseline classifiers to 0.86. Cross-validation provided additional evidence of the approach's dependability on the dataset by confirming strong and fair performance with minimal volatility (±1.2%).

Longitudinal modeling allows PsyCrisCare to track changes in students' mental health over time, which can lead to more accurate and tailored predictions in the future. To facilitate safe implementation across institutions, privacy-preserving federated learning frameworks could be implemented.

Deeper emotion recognition in reflections could be achieved by integrating real-time behavioral information from wearable devices with modern natural language processing methods. This integration could further enhance insights. Additionally, counselors and students would benefit from the openness that would result from incorporating explainable AI (XAI) approaches, which would guarantee confidence and usefulness. As a data-driven, scalable platform for proactive and equitable mental health intervention in education, PsyCrisCare demonstrates strong potential in the end.

To address generalizability, we propose as future work a multi-institutional validation study in which PsyCrisCare is retrained and tested across datasets from diverse universities and vocational colleges, ideally spanning different countries and languages. Such a study would assess performance drift, fairness across new demographic subgroups, and robustness under distribution shift, and would guide potential domain adaptation strategies (e.g., fine-tuning text modules, recalibrating thresholds, or re-estimating class weights) before real-world deployment.

Ethical approval, data origin, and privacy

The data used in this study were obtained from the publicly available Student Mental Health and Resilience Dataset hosted on Kaggle [30]. All data were fully de-identified prior to public release, including removal of direct identifiers, and free-text reflections were handled carefully to prevent disclosure of any potentially identifying information. Because this study involved secondary analysis of anonymized public data, no additional institutional ethics approval or IRB review was required. All data were stored on secure, access-controlled systems, accessible only to the research team, and will be retained in accordance with institutional data retention policies before secure deletion.

Data availability statement

The study uses the Student Mental Health and Resilience Dataset (Kaggle, https://www.kaggle.com/datasets/ziya07/student-mental-health-and-resilience-dataset) [30], accessed on March 5, 2025, under the Kaggle Terms of Use. The dataset is publicly available and de-identified. Any preprocessing performed in this study (e.g., cleaning or formatting) produced a derivative dataset used for analysis; this derived dataset is available from the corresponding author upon reasonable request. No additional data generated during the study are included in the article.

Author contributions

Jing Gao writing original draft preparation & methodology, Jing Gao investigation & writing review and editing.

References

[1] Kolenik, T., Schiepek, G., & Gams, M. (2024). Computational psychotherapy system for mental health prediction and behavior change with a conversational agent. *Neuropsychiatric Disease and Treatment*, 2465-2498. https://doi.org/10.2147/NDT.S417695

[2] Kolenik, T. (2025). Intelligent Cognitive System for Computational Psychotherapy with a Conversational Agent for Attitude and Behavior Change in Stress, Anxiety, and Depression. *Informatica*, *49*(2). DOI: https://doi.org/10.31449/inf.v49i2.8738

[3] Kolenik, T. (2022). Methods in digital mental health: smartphone-based assessment and intervention for stress, anxiety, and depression. In *Integrating Artificial Intelligence and IoT for Advanced Health Informatics: AI in the Healthcare Sector* (pp. 105-128). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-91181-2_7

[4] Kolenik, T., & Gams, M. (2021). Intelligent cognitive assistants for attitude and behavior change support in mental health: state-of-the-art technical review. *Electronics*, *10*(11), 1250. https://doi.org/10.3390/electronics10111250

[5] Kolenik, T., & Gams, M. (2021). Persuasive technology for mental health: One step closer to (mental health care) equality?. *IEEE Technology and Society Magazine*, *40*(1), 80-86. 10.1109/MTS.2021.3056288

[6] Meng, X., Cui, X., Zhang, Y., Wang, S., Wang, C., Li, M., & Yang, J. (2025). Mining Suicidal Ideation in Chinese Social Media: A Dual-Channel Deep Learning Model with Information Gain Optimization. Entropy, 27(2), 116. https://doi.org/10.3390/e27020116

[7] Hossain, M. M., Hossain, M. S., Mridha, M. F., Safran, M., & Alfarhood, S. (2025). Multi task opinion enhanced hybrid BERT model for mental health analysis. Scientific Reports, 15(1), 3332. https://doi.org/10.1038/s41598-025-86124-6

[8] Allam, H., Davison, C., Kalota, F., Lazaros, E., & Hua, D. (2025). AI-Driven Mental Health Surveillance: Identifying Suicidal Ideation Through Machine Learning Techniques. Big Data and Cognitive Computing, 9(1), 16. https://doi.org/10.3390/bdcc9010016

[9] Absar, N., Islam, M. M., & Somaya, Z. N. (2025). Explainable depression detection from low-resource languages using CNN-BiLSTM with deep attention mechanism. Machine Learning for Computational Science and Engineering, 1(2), 29. https://doi.org/10.1007/s44379-025-00026-y

[10] Vallu, V. R., Samudrala, V. K., & Pulakhandam, W. (2025). AI-Driven Digital Twin Framework for Accurate Mental Health Stress Detection and Personalized Management. In Accelerating Product Development Cycles With Digital Twins and IoT Integration (pp. 377-408). IGI Global Scientific Publishing. DOI: 10.4018/979-8-3373-2028-1.ch018

[11] Tang, H., Miri Rekavandi, A., Rooprai, D., Dwivedi, G., Sanfilippo, F. M., Boussaid, F., & Bennamoun, M. (2024). Analysis and evaluation of explainable artificial intelligence on suicide risk assessment. Scientific reports, 14(1), 6163. https://doi.org/10.1038/s41598-024-53426-0

[12] Velagaleti, S. B., Choukaier, D., Singh, S., Kaur, J., Dubey, A., Mujoo, S., ... & Singh, R. (2024). Utilizing Emotion Analysis for Suicide Prediction and Mental Health Detection in Students with Deep Learning. International Journal of Intelligent Systems and Applications in Engineering, 12, 729-738. http://dx.doi.org/10.2139/ssrn.5132301

[13] Zhang, Z. (2024). Early warning model of adolescent mental health based on big data and machine learning. Soft Computing, 28(1), 811-828. https://doi.org/10.1007/s00500-023-09422-z

[14] Singh, H., Kaur, B., Sharma, A., & Singh, A. (2024). Framework for suggesting corrective actions to help students intended at risk of low

performance based on experimental study of college students using explainable machine learning model. Education and Information Technologies, 29(7), 7997-8034. https://doi.org/10.1007/s10639-023-12072-1

[15] Zhong, B. (2025). Fine-grained sentiment analysis using multidimensional feature fusion and GCN. Journal of Information and Telecommunication, 9(1), 91-112. https://doi.org/10.1080/24751839.2024.2386785

[16] Feng, R., Mishra, V., Hao, X., & Verhaeghen, P. (2025). The association between mindfulness, psychological flexibility, and rumination in predicting mental health and well-being among university students using machine learning and structural equation modeling. Machine Learning with Applications, 19, 100614. https://doi.org/10.1016/j.mlwa.2024.100614

[17] Darko, A. P., Antwi, C. O., Adjei, K., Zhang, B., & Ren, J. (2024). Predicting determinants influencing user satisfaction with mental health app: An explainable machine learning approach based on unstructured data. Expert Systems with Applications, 249, 123647. https://doi.org/10.1016/j.eswa.2024.123647

[18] Wang, Y., Wang, X., Zhao, L., & Jones, K. (2025). A case for the use of deep learning algorithms for individual and population level assessments of mental health disorders: Predicting depression among China's elderly. Journal of Affective Disorders, 369, 329-337. https://doi.org/10.1016/j.jad.2024.09.147

[19] Zhou, S. C., Zhou, Z., Tang, Q., Yu, P., Zou, H., Liu, Q., ... & Luo, D. (2024). Prediction of non-suicidal self-injury in adolescents at the family level using regression methods and machine learning. Journal of affective disorders, 352, 67-75. https://doi.org/10.1016/j.jad.2024.02.039

[20] Soman, G., Judy, M. V., & Abou, A. M. (2025). Human guided empathetic AI agent for mental health support leveraging reinforcement learning-enhanced retrieval-augmented generation. Cognitive Systems Research, 90, 101337. https://doi.org/10.1016/j.cogsys.2025.101337

[21] Tian, Z., & Yi, D. (2024). Application of artificial intelligence based on sensor networks in student mental health support system and crisis risk prediction. Measurement: Sensors, 32, 101056. https://doi.org/10.1016/j.measen.2024.101056

[22] Chen, Y., & Ke, J. (2025). Multivariate Decision Tree-Oriented Early Warning Method for College Students' Psychological crisis risk Behavior. International Journal of High-Speed Electronics and Systems, 34(03), 2440120. https://doi.org/10.1142/S0129156424401207

[23] Sheng, C. (2024). Simulation application of sensors based on Kalman filter algorithm in student psychological crisis risk prediction model. Measurement: Sensors, 33, 101190. https://doi.org/10.1016/j.measen.2024.101190

[24] Wu, Y. (2025). Data Fusion Model for Psychological crisis risk Early Warning System Using Data Mining Techniques. Informatica, 49(23). https://doi.org/10.31449/inf.v49i23.7277

[25] Sara, S. S., Rahman, M. A., Rahman, R., & Talukder, A. (2024). Prediction of suicidal ideation with associated risk factors among university students in the southern part of Bangladesh: Machine learning approach. Journal of affective disorders, 349, 502-508. https://doi.org/10.1016/j.jad.2024.01.092

[26] Ojo, Y., Makinde, O. A., Babatunde, O. V., Babatunde, G., & Okeowo, S. (2025). Evaluating AI-Driven Mental Health Solutions: A Hybrid Fuzzy Multi-Criteria Decision-Making Approach. AI, 6(1), 14. https://doi.org/10.3390/ai6010014

[27] Bojic, I., Ong, Q. C., Ito, S., Liu, J., Lawate, A., Palaiyan, M., ... & Car, J. (2025). AI-empowered health coaching for university students: A mixed-method process evaluation. Computers in Biology and Medicine, 194, 110271. https://doi.org/10.1016/j.compbiomed.2025.110271

[28] Kasereka, S. K., Tshibangu, K. N., Nyembo, M. M., Tshitenge, L. K., Muzindusi, M. K., Ilunga, G. W., ... & Kyamakya, K. (2025). Leveraging Artificial Intelligence for Advancements in Mental Disorders: A Short Review. Procedia Computer Science, 257, 676-683. https://doi.org/10.1016/j.procs.2025.03.087

[29] Misgar, M. M., & Bhatia, M. P. S. (2024). Unveiling psychotic disorder patterns: A deep learning model analysing motor activity time-series data with explainable AI. Biomedical Signal Processing and Control, 91, 106000. https://doi.org/10.1016/j.bspc.2024.106000

[30] https://www.kaggle.com/datasets/ziya07/student-mental-health-and-resilience-dataset

SwarmShield: A Fully Decentralized Trust-Based Defense Against Adversarial Poisoning in Federated Learning

Cynara Justine[1], Sathyanarayanan Manamohan[2,3], Linu Shine[1], Jiji C. V.[3]
[1]College of Engineering Trivandrum, APJ Abdul Kalam Technological University, Kerala, India
[2]Hewlett Packard Enterprise, Bangalore, India
[3]Shiv Nadar University, Chennai, India
Email: cynara@cet.ac.in, sathya@hpe.com, linushine@cet.ac.in, jijicv@snuchennai.edu.in

Keywords: Cybersecurity, adversarial attack, data poisoning attack, federated learning, swarm learning, decentralized defense

Received: June 1, 2025

Adversarial poisoning attacks in federated learning systems can severely compromise model integrity, especially when malicious nodes inject corrupted updates. Existing defenses often rely on a trusted central aggregator, introducing a Single Point of Failure and limiting scalability. To overcome these challenges, we propose SwarmShield, a decentralized, trust-aware defense framework based on Swarm Learning. SwarmShield eliminates the need for a central coordinator by redistributing trust evaluation and model merging across peer nodes. It selectively transmits intermediate model layers, applies dimensionality reduction, and clusters parameter vectors to assess similarity. Trust scores are dynamically computed for each node based on its proximity to the cluster centroid, and nodes with low trust are excluded from aggregation. A secure, trust-weighted averaging mechanism is used for model updates, with integrity ensured through cryptographic hashing and blockchain logging. Extensive experimentation with different types of adversarial data poisoning attacks on CIFAR10 dataset with Resnet50 model demonstrate an average improvement in accuracies by 24.8%. Additionally, its generalizability is robustly demonstrated through successful application to the real-world DermaMNIST medical imaging dataset, where SwarmShield consistently maintained or improved model accuracy across diverse attack scenarios. We also evaluate SwarmShield on the TwoLeadECG time series dataset, highlighting its behavior under temporal adversarial settings. These results validate SwarmShield's effectiveness, scalability, and resilience in adversarial federated learning settings. Further analysis through ablation studies validates the framework's design by quantifying the contribution of each component, while robustness tests demonstrate its resilience across varying ratios of malicious nodes. Our experimental results demonstrate that the proposed approach significantly outperforms existing state-of-the-art methods.

Povzetek: Članek predstavi SwarmShield, decentralizirano in zaupanja zavedno obrambo proti zastrupljanju v federiranem učenju, ki brez centralnega agregatorja z gručenjem ocenjuje podobnost posodobitev, dinamično dodeli točke zaupanja in izloča zlonamerne vozle, pri čemer integriteto podpira še kriptografsko zgoščevanje in beleženje v verigi blokov.

1 Introduction

Federated Learning (FL) enables decentralized model training without centralized data collection, preserving data privacy by keeping data on local devices [1]. However, its distributed nature introduces critical security risks, notably adversarial poisoning attacks from compromised nodes. These attacks degrade global model performance through malicious local updates, and are difficult to detect. Centralized defenses, such as server–based parameter aggregation, further introduce Single Points of Failure (SPOF) and elevate the attack surface [2]. Several centralized defense strategies, such as *Krum* [3] and *Foolsgold* [4], etc. have been proposed to resist poisoning and sybil attacks; however, they rely on a trusted aggregator, which remains a critical vulnerability in adversarial settings. Centralized aggregation solutions in FL are vulnerable to manipulation, as a compromised aggregator can distort model updates, leak private data, or collapse system reliability. These solutions also provide attackers a global view of updates, heightening privacy risks.

In contrast, decentralized approaches distribute the trust boundary, improving system resilience. Notable prior works like *Ditto* [5], and Byzantine-tolerant gradient descent techniques [3] have proposed decentralization through trust initialization, personalization, or robust aggregation. However, many existing approaches remain vulnerable to attacks due to their dependence on partial centralization or static trust assumptions.

To address these challenges, we propose *SwarmShield*,

a decentralized trust-aware defense framework that detects and mitigates malicious nodes using collaborative clustering and trust scoring. It enhances adversarial robustness without compromising privacy or requiring centralized coordination, contributing towards secure and resilient FL deployments in real-world distributed environments. *SwarmShield* differs in that it offers a fully decentralized and adaptive peer trust mechanism, integrating layer-wise parameter extraction, clustering-based behavioral analysis, and blockchain-backed tamper-proofing, without requiring global trust anchors or centralized bootstrapping.

The key contributions of this work are as follows:

- **Fully decentralized defense architecture:** We design a central-aggregator-free framework where trust computation, parameter filtering, and model merging are executed collaboratively among peer nodes.
- **Dynamic trust evaluation through clustering:** Each node's behavior is assessed in real time through unsupervised clustering of selected-layer parameter vectors, with trust scores reflecting deviation from cluster centroids.
- **Blockchain-backed integrity and auditability:** SwarmShield employs cryptographic hashing and blockchain logging to secure parameter transmissions and preserve an immutable trust history, enabling tamper-proof, auditable FL.

These contributions collectively strengthen the security of federated learning while maintaining scalability, privacy, and interpretability in adversarial environments.

2 Literature survey

Distributed learning architectures underpin modern federated systems, enabling collaboration without direct data sharing. For example, Wu [6] introduced a distributed intelligent optimization algorithm leveraging the Spark framework for large-scale e-commerce data mining. Their approach adapts fitness functions and topological structures for scalable, robust optimization. They proposed a distributed intelligent optimization algorithm for e-commerce user purchase data mining, where the learning objective is designed to balance cluster membership and data representation in a principled manner. This approach relies on modeling spatial membership among data points and systematically updates parameters in parallel across distributed nodes. Such a design enables efficient, real-time adaptation and robust convergence within large-scale distributed systems. Similarly, Ahmed et al. [7] demonstrate the application of vanilla split learning to cyber-physical systems, enabling collaborative model training on local sensor data. Their paradigm ensures privacy and reduces communication, directly addressing resource and privacy constraints prevalent in modern federated settings.

The challenge of ensuring adversarial robustness in Federated Learning, particularly against poisoning attacks, remains a pressing concern. Recent research underscores the importance of secure aggregation in federated learning to ensure privacy and robustness. Notably, a 2025 study introduces a federated learning framework that employs both differential privacy and homomorphic encryption, achieving strong data confidentiality and maintaining high model accuracy even when adversarial clients are present [18]. Ongoing studies continue to address threats such as label-flipping and Byzantine attacks, with decentralization identified as key to enhancing system resilience and mitigating single points of failure. These contributions collectively advance the field's understanding and implementation of secure, robust federated learning architectures.

Early defense efforts primarily focused on centralized aggregation techniques. For example, in [3], Blanchard et al. introduced a Byzantine-resilient rule namely *Krum* that selects gradient updates closest in Euclidean space. While effective in simple settings, it suffers from high computational overhead and limited scalability in high-dimensional, non-IID settings. To address this, in [8] the authors layered an additional filtering step over *Krum*, and proposed a new approach namely *Bulyan* improving robustness but not scalability. Yin et al. [19] sought to optimize statistical guarantees in Byzantine environments, yet their reliance on centralized aggregation makes them ill-suited for dynamic or trustless networks. FoolsGold [9], using cosine similarity to suppress updates from colluding adversaries, innovated in contribution-based weighting but remained sensitive to hyperparameter choices and incurred communication overhead.

Shifting away from centralized designs, Warnat-Herresthal et al. [20] introduced Swarm Learning—embedding consensus mechanisms into model training. In parallel, BAFFLE [10] and FLTrust [11] tried to anchor trust via blockchain and trusted initialization, respectively. However, BAFFLE incurred latency from blockchain overhead, while FLTrust reintroduced a central trust bottleneck.

Ditto [5] took a personalization route to isolate client poisoning effects but raised concerns about divergence and model consistency. Zhou et al. [12] proposed *Fed_BVA*, integrating bias-variance analysis for adversarial training. Though robust across adversary types and data distributions, it incurred complexity from second-order statistics computation.

To overcome shortcomings of prior defenses, recent works integrate cryptographic and differential privacy tools. Jin et al. [21] incorporated homomorphic encryption to secure model aggregation, providing strong theoretical guarantees. However, its deployment is limited by excessive computational load, which poses a barrier to real-time FL at the edge level. Zhang et al. [22] proposed *DBFAT*, which reweights local losses and globally regularizes model updates. Although effective in accuracy-robustness trade-offs, the approach requires domain-specific hyperparame-

Table 1: Comparison of existing adversarial defense methods in federated learning

Method	Approach	Advantages	Limitations
Krum [3]	Centralized, Byzantine-resilient rule that selects gradients closest in Euclidean space.	Effective in simple settings.	High computational overhead; limited scalability in high-dimensional, non-IID settings.
Bulyan [8]	Adds a filtering step over Krum.	Improves robustness.	Does not improve scalability.
FoolsGold [9]	Centralized, uses cosine similarity to weight contributions.	Innovative contribution-based weighting.	Sensitive to hyperparameters; incurs communication overhead.
BAFFLE [10]	Anchors trust using a blockchain.	Attempts to anchor trust without a central aggregator.	Incurs latency from blockchain overhead.
FLTrust [11]	Anchors trust via a trusted initialization process.	Attempts to anchor trust.	Reintroduces a central trust bottleneck.
Ditto [5]	Personalization to isolate clients.	Isolates effects of client poisoning.	Raises concerns about model divergence and consistency.
Fed_BVA [12]	Integrates bias-variance analysis for adversarial training.	Robust across various adversary types and data distributions.	Incurs complexity from second-order statistics computation.
FL-Defender [13]	Uses PCA and angular gradient similarity for detection.	Computationally efficient.	Performance degrades with high-dimensional models.
FedBayes [14]	Reframes aggregation through a Bayesian lens.	Enhances trust minimization.	Introduces probabilistic modeling complexity and latency.
Zeno [15]	Uses suspicion scores from validation loss to evaluate updates.	Provides a certifiable robustness measure under certain assumptions.	Practical deployment requires managing validation overhead.
GAN-based Defenses [16]	Use generative models for input sanitization and anomaly repair.	Can detect and neutralize poisoned samples by learning benign data distributions.	Significant training complexity; potential trade-offs with model utility.
Moving Target Defense [17]	Decentralized, periodically perturbs model parameters.	A novel approach to confuse potential adversaries.	May slow down model convergence.
SwarmShield (Ours)	**Fully decentralized, trust-based defense using collaborative clustering and blockchain logging.**	**Eliminates central point of failure; adaptive trust; provides auditable integrity.**	**Performance depends on the quality of clustering and inter-node communication.**

ter tuning. FL-Defender [13] innovates with PCA and angular gradient similarity for targeted attack detection. Although computationally efficient, its performance degrades with high-dimensional models.

FedBayes [14] reframes aggregation through a Bayesian lens, enhancing trust minimization. However, this introduces probabilistic modeling complexity and latency.

Xu et al.[23] proposed *Dual Defense*, which simultaneously enhances privacy and defends against poisoning by combining differential privacy with robust aggregation, but demands tight tuning to balance noise and utility.

Robustness against poisoning attacks in federated learning has evolved through both theoretical and empirical defense approaches. Yin et al. [19] introduced a Byzantine-robust distributed learning method with optimal statistical guarantees, enabling formal treatment of adversarial tolerance in aggregation. Although promising, scalability remains challenging in highly heterogeneous environments. Zeno, proposed by Xie et al. [15] is a defense mechanism that evaluates update trustworthiness using suspicion scores derived from validation loss. This provides a certifiable robustness measure under certain assumptions, though practical deployment requires managing validation overhead. In parallel, generative approaches have emerged as a complementary defense strategy. Generative approaches, particularly those inspired by Generative Adversarial Networks (GANs), are employed for backdoor defense through mechanisms like input sanitization and anomaly repair. Zhang et al. [24] provide a comprehensive survey on how GANs can detect and inpaint anomalous regions to neutralize poisoned samples, leveraging learned benign data distributions. Comprehensive surveys, such as Akhtar et al. (2021), synthesize recent advances in both generative and statistical defenses, noting their promise despite inherent challenges like significant training complexity and trade-offs with model utility [16, 25]. Ye et al. [17] incorporated *Moving Target Defense* in decentralized FL, where the model parameters are periodically perturbed to confuse potential adversaries. Although novel, this may slow the convergence.

Despite their innovative approaches, most existing frameworks face critical limitations, including reliance on a non-malicious central aggregator, lack of mechanisms for malicious node removal, and inability to maintain learning continuity after an attack.

Many recent works have focused on improving privacy and computational efficiency in distributed machine learning frameworks, aligning with the motivations behind our work. Azeri et al. [7] introduced an optimized Vanilla Split Learning method for resource-constrained Cyber-Physical Systems, demonstrating how a split neural architecture can reduce client-side computation while preserving data privacy in collaborative environments.

In a complementary study, Chen et al. [18] proposed a federated learning architecture incorporating differential privacy and homomorphic encryption to ensure privacy preservation in large-scale systems. Their experimental setup, which includes a distributed physical server environment, demonstrates robust model performance even under adversarial scenarios, with only a minor trade-off in accuracy. These works validate the growing emphasis on balancing scalability, security, and computational feasibility in distributed learning, which our work advances further by integrating trust-aware node filtering and robust election mechanisms in the context of adversarial federated environments.

A comparison of existing adversarial defense methods in federated learning is shown in Table 1.

3 Methodology

The proposed methodology addresses three key issues in FL security: (i) vulnerability of centralized trust mechanisms to single-point failures, (ii) lack of adaptive malicious node detection during aggregation, and (iii) insufficient privacy-preserving verification. We introduce *SwarmShield*, a decentralized defense framework that distributes trust computation across swarm nodes using cryptographic verification, dynamically adjusts node weightage through clustering-based analysis, and enforces zero-trust principles via blockchain-based logging and secured computation.

In this section, we first outline the common adversarial attacks in FL environments, followed by a detailed presentation of our proposed approach.

3.1 Adversarial attacks

The four types of adversarial attacks we have waged for evaluating the improvement in robustness by our solution SwarmShield are label flipping attack, extreme noise attack, Projected Gradient Descent (PGD) attack and Fast Gradient Sign Method (FGSM) attack. Swarmshield framework includes custom implementations of the label flipping and extreme noise attacks. For label flipping, the designated malicious peer contaminates its data stream before training, eg: specifically flipping labels “cars” and “birds”. Additionally, the Cleverhans framework [26] is used for generating FGSM (Fast Gradient Sign Method) [27] and PGD (Projected Gradient Descent) [28] adversarial attacks. A brief description of these attacks and its relevance are given below.

3.1.1 Label flipping attack

Label flipping is an adversarial technique in which the labels of a subset of training data are intentionally altered to mislead the model’s learning process [29]. This type of attack is relevant in scenarios involving training data poisoning, where the objective is to compromise model integrity by injecting false labels. The impact on model robustness is often severe, particularly in tasks that rely heavily on accurate label information.

3.1.2 Extreme noise attack

In extreme noise attacks, random noise with high variance is added to the input data, disrupting the model’s ability to detect meaningful patterns [30]. This attack targets the model’s sensitivity to input perturbations, making it a crucial test for evaluating a model’s noise tolerance.

3.1.3 Projected gradient descent (PGD) attack

PGD is a powerful iterative adversarial attack that generates perturbations by taking multiple small steps along the gradient of the model’s loss function. It is considered one of the most effective attacks for evaluating model robustness against adversarial examples, due to its optimization-based approach that converges to the worst-case perturbation within a given constraint [28].

3.1.4 Fast gradient sign method (FGSM) attack

FGSM is a single-step adversarial attack where perturbations are crafted using the sign of the gradient of the loss function with respect to the input data [27]. It is widely used for its simplicity and speed, making it a baseline for evaluating model vulnerability to adversarial examples. The attack underscores the importance of gradient-based defenses in securing deep learning models.

3.2 Proposed SwarmShield framework

SwarmShield is a decentralized framework built atop the Swarm Learning (SL) paradigm [31]. It provides defense against adversarial attacks by executing collaboratively across peer nodes using cryptographic guarantees and clustering-based trust mechanisms. In this section, we discuss the various features of Swarmshield framework such as decentralized peer collaboration without central aggregator, trust-based peer monitoring, dynamic peer election, blockchain-backed tamper-proofing and auditability and efficiency optimizations for edge deployment. We also present the algorithm, workflow and flowchart.

3.2.1 Decentralized peer collaboration without central aggregator

In Swarm Learning (SL), the system operates over R communication rounds. In each round $r \in \{1, \ldots, R\}$, every peer node P_i trains its local model on its private dataset D_i. The training is performed for E local epochs.

The local model weights for peer i, denoted by θ_i, are updated iteratively. For a single local update step k within a communication round r, the update rule is:

$$\theta_{i,k+1}^{(r)} = \theta_{i,k}^{(r)} - \eta \cdot \nabla_\theta \mathcal{L}(D_i; \theta_{i,k}^{(r)}) \tag{1}$$

where:

- $\theta_{i,k}^{(r)}$ are the weights of peer i's model at local step k of communication round r.
- η is the learning rate.
- $\mathcal{L}(D_i; \theta_{i,k}^{(r)})$ is the loss function evaluated on the local dataset D_i using the current weights.

After E local epochs, the final local model weights $\theta_i^{(r)}$ are shared for aggregation.

Preservation of data privacy is highlighted by the fact that no raw data leaves any of the participating nodes. To minimize communication overhead while preserving critical model updates, each peer transmits only the parameters from a selected subset of layers. These parameters are flattened into a single vector for efficient analysis.

The parameter vector v_i for peer i is constructed as follows:

$$v_i = \text{Flatten}(\{\theta_{i,j} \mid j \in \mathcal{J}\}) \tag{2}$$

where:

- v_i is the flattened parameter vector for peer i.
- $\theta_{i,j}$ represents the weight parameters of the j-th layer of peer i's model.
- $\mathcal{J}$ is the pre-configured index set of layers selected for transmission and trust evaluation.

This formulation draws upon adaptive communication strategies explored in [32], enabling tunable trade-offs between transmission cost and model fidelity. This also ensures communication efficiency and reduces the risk of sensitive information leakage. For our experiments, we follow the common heuristic of selecting the final fully-connected layers of the ResNet-50 model, as these layers typically capture the high-level semantic representations most vulnerable to adversarial poisoning.

While the optimal layers for transmission are application-specific, the selection process follows a clear heuristic to maximize security and efficiency. For our experiments with the ResNet-50 architecture, we selected only the final fully-connected layers for transmission. This choice is based on the rationale that these deeper layers capture high-level semantic features and are therefore most impactful on the model's final decision. Consequently, they are often the primary target of, and most sensitive to, adversarial poisoning attacks. Transmitting only these layers provides the most relevant information for trust evaluation while minimizing communication overhead.

3.2.2 Trust-based peer monitoring

At each round, the elected leader gathers parameter vectors from participating peers to evaluate their trustworthiness. To ensure operational clarity and address which model layers are transmitted, our protocol focuses on a preconfigured subset of layers known to be most indicative of model behavior, typically the final fully-connected or convolutional layers. This targeted approach balances communication overhead with effective anomaly detection.

Before analysis, Principal Component Analysis (PCA) is applied to the flattened parameter vectors (v_i) of these selected layers, which we now denote as x_i after PCA reduction. This dimensionality reduction step enhances the subsequent clustering by capturing the principal axes of variance among peer updates. Peers are then grouped using k-means clustering based on these PCA-reduced vectors. The objective is to partition the set of all peer vectors into K disjoint clusters $S = \{S_1, S_2, \ldots, S_K\}$ by minimizing the variance within the clusters:

$$\arg\min_S \sum_{k=1}^{K} \sum_{x \in S_k} \|x - \mu_k\|^2 \tag{3}$$

where S_k is the set of peer vectors belonging to cluster k and μ_k is the centroid of that cluster [33].

Pairwise similarity between peer vectors x_i and x_j is computed using cosine similarity:

$$\text{sim}(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|} \tag{4}$$

To detect anomalies, a hybrid outlier score (OS) is calculated combining distance, similarity, and cluster density:

$$\text{OS}(x_i) = \|x_i - c\| \cdot (1 - \text{sim}(x_i, c)) \cdot \frac{1}{|S_k|} \tag{5}$$

where $|S_k|$ is the number of peer vectors in the cluster containing x_i, and c is the centroid of that cluster, inspired by local outlier factor methods [34].

Each peer's trust score T_i^t is updated based on the similarity of its transmitted vector x_i to the main peer cluster centroid c^t, and penalized in case of hash mismatches:

$$T_i^t = (1-\alpha) T_i^{t-1} + \alpha \left(1 - \frac{\|x_i - c^t\|}{\max_j \|x_j - c^t\|}\right) - \lambda \cdot \kappa_i^t \tag{6}$$

Here, α is a forgetting factor, and κ_i^t is an indicator function such that $\kappa_i^t = 1$ if a hash mismatch is detected for node i at time t, and 0 otherwise. The parameter λ controls the severity of the penalty applied due to tampering or inconsistency. This memory-aware and tamper-aware formulation ensures that trust scores adapt over time based on both

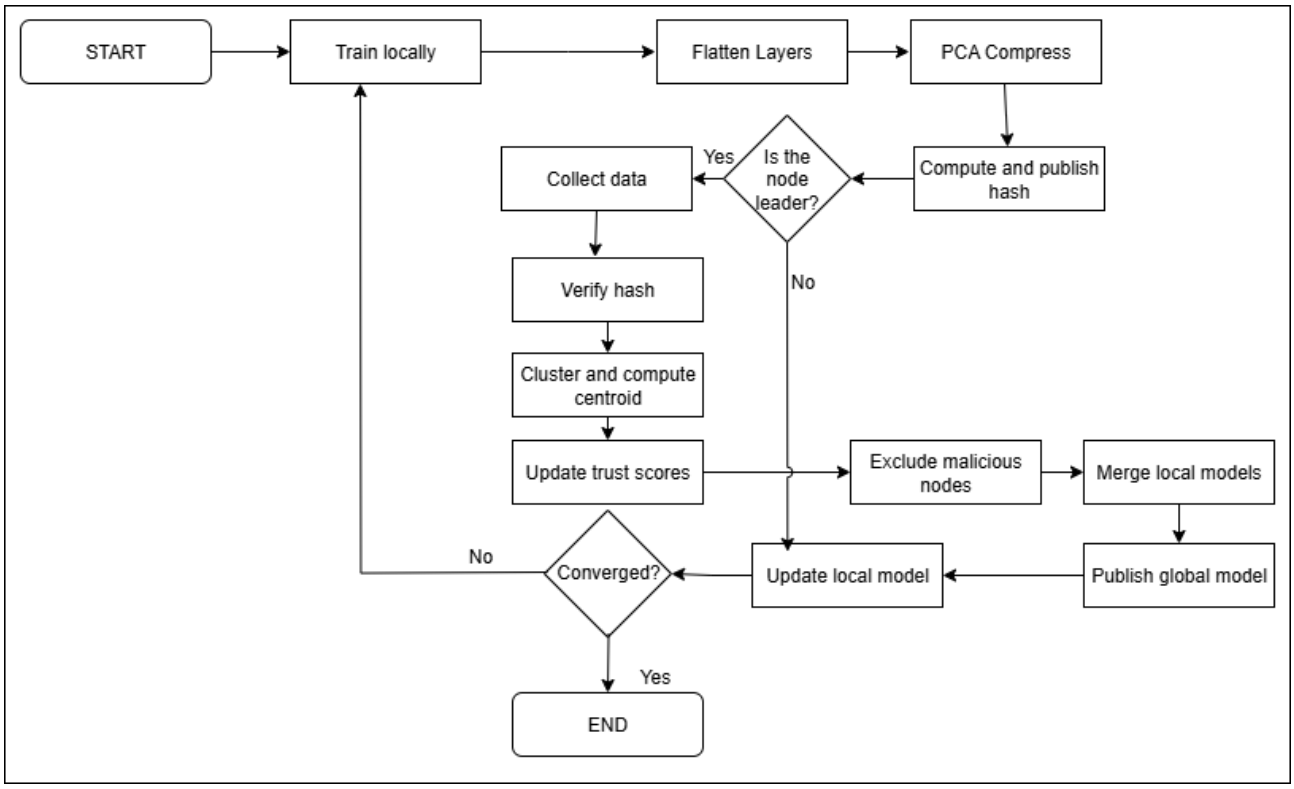


Figure 1: Flowchart illustrating the Compact SwarmShield Protocol. Each component corresponds directly to the operations detailed in Algorithm 1, providing a high-level visual abstraction of the training, compression, trust update, and model aggregation stages.

behavioral consistency and cryptographic integrity. Peers with low trust scores are either downweighted or excluded entirely from the aggregation process.

3.2.3 Dynamic peer election

The leader election mechanism is fully decentralized and implemented using a programmable smart contract on a tamperproof distributed ledger. This eliminates reliance on a central coordinator, addressing a key limitation in baseline systems such as FL-Defender [13].

The protocol for electing a leader in each communication round proceeds as follows:

1. Each node accesses the historical trust scores of its peers from the distributed ledger and casts a vote for another node. Self-voting is explicitly disallowed to prevent self-promotion.

2. The votes are tallied on the ledger, and all nodes arrive at a consensus. The node with the most votes is designated the leader for the round.

3. In the event of a tie, the node with the higher average historical trust score is selected as the tie-breaker.

To further reduce the risk of collusion and deterministic bias, randomness from verifiable external sources is incorporated into the election process. The system also implements a trust decay mechanism, requiring nodes to consistently prove their trustworthiness to maintain a high score.

3.2.4 Blockchain-backed tamper-proofing and auditability

Before synchronization, a cryptographic hash of the selected parameters is computed using Secure Hashing Algorithm SHA-256 [35] to ensure integrity, as shown by:

$$h_i = \text{SHA-256}(x_i) \tag{7}$$

where h_i is the hash digest of parameter x_i.

This hash is published to a blockchain ledger, ensuring tamper-proofing and auditability. Trust scores are pushed and retained in the blockchain which is the immutable ledger. During parameter merging, hashes are recomputed and verified. Any mismatch penalizes the responsible peer by reducing its trust score [36].

Based on these scores, node weightages are updated dynamically. To further enforce integrity and fairness, SwarmShield implements a grace threshold mechanism. Peers whose trust scores fall below τ^t (the dynamic merge threshold) are not immediately excluded but allowed a cooldown period to improve behavior. Such peers may rejoin the network after demonstrating improved behavior, balancing robustness with ethical collaboration.

In addition, during parameter merging, each peer's model update is hashed and published to a blockchain ledger to ensure tamper-proofing and auditability. If the recomputed hash does not match the published value, the responsible peer is penalized by reducing its trust score using a penalty term $\lambda \cdot \kappa_i^t$, where $\kappa_i^t = 1$ in case of mismatch, and 0 otherwise.

The global model aggregation is performed using trust-weighted averaging:

$$w_{\text{global}} = \frac{\sum_{i=1}^{n} T_i \cdot w_i}{\sum_{i=1}^{n} T_i} \tag{8}$$

Only nodes with $T_i > \tau^t$ participate in the merge, where the dynamic trust threshold τ^t is defined as:

$$\tau^t = \mu^t - \beta\sigma^t \tag{9}$$

Here, μ^t and σ^t denote the mean and standard deviation of trust scores at time t, and β is a sensitivity parameter.

The sensitivity parameter β is crucial for setting the dynamic trust threshold, τ^t. It controls how strictly nodes are filtered for participation in global model aggregation. A higher β creates a lower, more lenient threshold, promoting broader participation. Conversely, a lower β results in a higher, more stringent threshold, favoring only nodes with trust scores closer to the average. This parameter scales with the standard deviation of trust scores, σ^t, allowing the system to dynamically adjust its strictness based on the network's trustworthiness. The value of β is empirically determined to balance network robustness against the need for sufficient participation. This dynamic system, combined with continuous trust score updates, supports the zero-trust principle by requiring nodes to constantly prove their trustworthiness.

3.2.5 Efficiency optimizations for edge deployment

SwarmShield incorporates two key optimizations to reduce computation and support real-time operation:

- **Dimensionality reduction:** PCA is applied on the flattened vectors before clustering to minimize computational overhead. Specifically, PCA is applied to the collection of flattened parameter vectors from all peers during the leader's aggregation step, occurring immediately before the k-means clustering. This reduces the feature space dimensionality, which not only lowers computational cost but also improves the signal-to-noise ratio, thereby enhancing the quality of the subsequent clustering for outlier detection.
- **Incremental clustering:** Clustering state from previous rounds is reused, avoiding redundant computation in stable scenarios.

These enhancements ensure that SwarmShield remains scalable even on low-power edge clusters. Algorithm 1 formally outlines the essence of the SwarmShield framework. The flowchart giving a visual depiction of the workflow for clearer conceptual understanding is shown in 1.

Algorithm 1 SwarmShield Protocol

Require: Model M_i (representing parameters θ_i), data D_i, layer index set $\mathcal{J}$
Ensure: Robust global model w_{global}

while not converged **do**
  Train locally for E epochs: $\theta_i^{(t+1)} = \theta_i^{(t)} - \eta\nabla\mathcal{L}(D_i; \theta_i^{(t)})$
  Flatten selected layers' parameters: $v_i = \text{Flatten}(\{\theta_{i,j} \mid j \in \mathcal{J}\})$
  PCA compress: $x_i = \text{PCA}(v_i)$
  Hash x_i: $h_i = \text{SHA-256}(x_i)$; publish h_i to blockchain
  if P_i is Leader **then**
    Collect $\{x_j\}, \{h_j\}$ from peers
    for all peers j **do**
      Verify SHA-256$(x_j) = h_j$
      if hash mismatch **then**
        Mark P_j for exclusion
      end if
    end for
    Cluster $\{x_j\}$; compute centroid c^t
    for all verified j **do**
      Compute OS(x_j), update T_j^t
      if $T_j^t < \tau^t$ **then**
        Mark P_j for exclusion
      end if
    end for
    $w_{\text{global}}^{(t+1)} = \frac{\sum_j T_j^t \theta_j^{(t+1)}}{\sum_j T_j^t}$
    Publish $w_{\text{global}}^{(t+1)}$, $\{T_j^t\}$ to blockchain
  else
    Wait for $w_{\text{global}}^{(t+1)}$ from Leader
  end if
  Update $\theta_i^{(t+1)} \leftarrow w_{\text{global}}^{(t+1)}$
end while

Notation: i, j, n: peer indices; (t): round index; P_i, D_i: peer i and its dataset; $\mathcal{J}$: selected layer indices; $\theta_i^{(t)}$: local model; v_i, x_i: flattened/PCA vectors; $w_{\text{global}}^{(t)}$: global model; h_i: hash for integrity; $c^{(t)}$: cluster centroid; $T_i^{(t)}$, $\tau^{(t)}$: trust score, threshold; $\kappa_i^{(t)}$: hash-fail flag; η, $\mathcal{L}$: learning rate, loss; α, λ: forgetting factor, trust penalty.

3.2.6 Summary of secure, collaborative workflow

The SwarmShield protocol integrates all the above stages into a cohesive cycle executed per communication round:

1. Local training and layer selection on private data.
2. Secure hashing and transmission of compact updates.
3. Leader election and similarity-based peer clustering.
4. Trust score updates and parameter filtering.

5. Trust-weighted model merging and blockchain logging.

The system enforces collaborative defense without centralization, enhances robustness to poisoning, and ensures verifiable model integrity.

3.3 Convergence analysis

We provide a simplified convergence analysis of the SwarmShield framework under standard assumptions for federated optimization. While a complete convergence analysis for decentralized, trust-aware federated learning with dynamic leader election is challenging, we outline the proof under the following assumptions:

- The global objective is to minimize a convex function $\mathcal{L}(w)$, which is decomposable as $\mathcal{L}(w) = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i(w)$, where $\mathcal{L}_i(w)$ local objective (loss) function for peer i. Here, w represents the global model weights and $\mathcal{L}(w)$ is the global objective (loss) function;
- Each $\mathcal{L}_i$ is L-smooth and convex, which are common assumptions for proving convergence in federated settings [37].
- Cosine similarity–based trust scores $T_i^t \in [0,1]$ are bounded and updated using stable statistics of parameter deviations.
- The trust-weighted model aggregation, a form of robust aggregation [38], at round t is given by:

$$w_{\text{global}}^{(t+1)} = \sum_{i=1}^{N} \alpha_i^t \theta_i^{(t)}, \quad \text{where } \alpha_i^t = \frac{T_i^t}{\sum_{j=1}^{N} T_j^t}, \tag{10}$$

where α_i^t – normalized trust weight for peer i at round t, N – total number of peers and T_i^t – trust score of peer i at round t;

Lemma 1 (Trust-weighted averaging preserves descent direction): If each local model $\theta_i^{(t)}$ performs a gradient step on $\mathcal{L}_i$, then the trust-weighted average update $w_{\text{global}}^{(t+1)}$ remains a descent direction for the global loss $\mathcal{L}$ under bounded trust variation.

Proof Sketch: From convexity and L-smoothness, the gradient descent step at each peer i satisfies:

$$\mathcal{L}_i(\theta_i^{(t)}) \le \mathcal{L}_i(w_{\text{global}}^{(t)}) + \langle \nabla\mathcal{L}_i(w_{\text{global}}^{(t)}), \theta_i^{(t)} - w_{\text{global}}^{(t)} \rangle + \frac{L}{2}\|\theta_i^{(t)} - w_{\text{global}}^{(t)}\|^2 \tag{11}$$

Using Jensen's inequality and trust-weighted aggregation:

$$\mathcal{L}(w_{\text{global}}^{(t+1)}) \le \sum_{i=1}^{N} \alpha_i^t \mathcal{L}_i(\theta_i^{(t)}) \tag{12}$$

This implies that if each $\mathcal{L}_i(\theta_i^{(t)}) \le \mathcal{L}_i(w_{\text{global}}^{(t)})$ (i.e., local progress), then global progress follows via the trust-weighted combination.

Remark 1. *The inequality in Equation 12 implies that, under convex loss functions and trust-weighted aggregation, global progress is guaranteed if each peer achieves local improvement. That is, if $\mathcal{L}_i(\theta_i^{(t)}) \le \mathcal{L}_i(w_{global}^{(t)})$ for all i, then $\mathcal{L}(w_{global}^{(t+1)})$ will also decrease, ensuring convergence through decentralized coordination.*

Theorem 1 (Convergence under stable trust weights): Suppose that each peer i performs SGD with learning rate $\eta_t = \frac{1}{\sqrt{t}}$ and trust scores T_i^t satisfy:

$$T_i^t \ge \epsilon > 0, \quad \forall i, t \tag{13}$$

where ϵ – a small positive constant, representing the lower bound for trust scores, η_t – learning rate at round t; Then the averaged global model $w_{\text{global}}^{(t)}$ converges in expectation to the global minimizer w^* at a rate $\mathcal{O}(1/\sqrt{t})$:

$$\mathbb{E}[\mathcal{L}(w_{\text{global}}^{(t)}) - \mathcal{L}(w^*)] \le \frac{C}{\sqrt{t}} \tag{14}$$

where C – a generic positive constant bounding the convergence rate, $w_{\text{global}}^{(t)}$ – global model weights at round t and w^* – global minimizer of the objective function.

Proof Sketch: This follows from standard convergence results for convex SGD (e.g., [1, 37]) when using weighted model averaging, with the additional assumption that trust weights remain bounded away from zero to prevent degenerate aggregation.

Discussion: While SwarmShield operates in a non-convex setting (ResNet-50 on CIFAR-10), such theoretical guarantees under convexity assumptions provide a baseline justification. Further extensions would require analyzing the interplay of non-convexity, dynamic leader election, and peer drop-out behavior.

4 Experiments, results and discussion

Experiments were conducted on a three-node testbed, with each node configured with 4 CPU cores, 32 GB RAM, and 300 GB of persistent storage. All nodes run Docker containers with PyTorch and the Swarm Learning Framework preinstalled. A 10-Gigabit Ethernet network links the nodes, and secure certificates are used for peer authentication. This environment is designed to emulate real-world federated learning settings with decentralized and privacy-preserving characteristics. Each node, or *peer*, operates independently on its own private dataset and does not share raw data with other nodes. One of the three peers is deliberately configured to inject adversarial perturbations into its data stream, simulating a malicious participant. The identity of the malicious peer is unknown to the others, thus reflecting realistic adversarial conditions.

4.1 Experimental setup

The SwarmShield framework has been developed to facilitate seamless evaluation of adversarial defenses in a decentralized federated learning environment. The implementation is structured to support modularity, real-time orchestration, and ease of deployment across multiple nodes. This section outlines the architectural design and deployment configuration used in our experiments.

4.1.1 Framework architecture

The SwarmShield system is implemented in a modular and extensible manner, enabling flexibility in integrating different datasets, models, and adversarial attack types. The core architecture comprises four major components:

- **Machine Learning Engine:** Encapsulates the training logic, model definitions, dataset loading, and adversarial attack routines. A modular design pattern allows seamless swapping or integration of different model architectures, datasets (e.g., CIFAR-10), and attack strategies (e.g., FGSM, PGD).
- **Swarm Learning Control Plane (SLCP):** Serves as the decentralized coordination layer responsible for peer enrollment, dynamic leader election, parameter synchronization, and trust score communication. By eliminating the need for a central server, the SLCP ensures privacy-preserving collaboration and fault-tolerant orchestration in federated setups.
- **Swarm Learning Integration Layer:** Acts as the interface between the ML engine and SLCP. It handles secure container deployment, communication callbacks, and the exchange of selected model parameters during each training round. It also ensures the isolation of peer logic from coordination logic, improving system integrity.
- **Automation Layer:** A suite of orchestration scripts is used to initiate experiments, deploy peer agents, verify system states, and gracefully shut down the network. These scripts automate operations such as launching the SLCP, setting up containers, and executing training rounds with and without defenses enabled.

SwarmShield enables truly distributed execution across multiple independent nodes using Swarm Learning's native synchronization, ensuring authentic federated learning conditions. The framework automates deployment through containerized templates and provides a lightweight callback interface for custom model integration, supporting diverse application domains while maintaining centralized result collection for robustness evaluation.

Configuration occurs through intuitive interfaces for model, dataset, and attack scenario selection, enabling reproducible experiments without modifying core orchestration logic. This containerized approach simplifies deployment while preserving the system's distributed nature and adaptability.

4.2 Datasets

To evaluate the robustness and versatility of our proposed SwarmShield framework, we conducted experiments across three distinct and publicly available datasets: a standard natural image dataset (CIFAR-10), a biomedical image dataset (DermaMNIST), and a time series dataset (TwoLeadECG).

4.2.1 CIFAR-10

The CIFAR-10 dataset [39] is a widely used benchmark for image classification. It consists of 60,000 color images of size 32×32 pixels, categorized into 10 mutually exclusive classes (e.g., plane, car, bird). The dataset is pre-divided into 50,000 training images and 10,000 testing images. For our experiments, CIFAR-10 serves as a standard baseline to evaluate the performance of our defense mechanism against various attacks in a well-understood, non-biomedical context.

4.2.2 DermaMNIST

The DermaMNIST dataset is part of the MedMNIST v2 [40] collection, a set of standardized biomedical image datasets. It is based on the HAM10000 dataset [41] and contains 10,015 dermatoscopic images of common pigmented skin lesions. The images are resized to 28×28 pixels and are categorized into 7 distinct classes, including melanoma, basal cell carcinoma, and benign keratosis. This dataset allows us to test SwarmShield's effectiveness on a multi-class, biomedical imaging task with inherent class imbalances.

4.2.3 TwoLeadECG

The TwoLeadECG dataset, sourced from the `aeon` library, provides a time series classification challenge. It contains 1,162 instances of electrocardiogram (ECG) readings, each with 2 leads (channels) and a length of 82 time steps. This dataset is derived from the MIT-BIH Arrhythmia Database [42] and is also available in the UCR Time Series Classification Archive [43]. For our experiments, we formulate a binary classification task to distinguish a specific arrhythmia (class 2) from normal heartbeats (class 1). This dataset serves to demonstrate SwarmShield's applicability beyond image-based data, highlighting its effectiveness in federated learning scenarios involving time series data—an important modality in healthcare monitoring.

4.3 Results on CIFAR10 dataset

This section presents an empirical evaluation of SwarmShield's effectiveness in enhancing adversarial robustness within a decentralized federated learning



Figure 2: Adversarial attacks on CIFAR10 samples: original images (top row), Noise attack adding static (middle row), and subtle FGSM/PGD perturbations (bottom rows) designed to induce misclassification.

Table 2: Detailed class-wise accuracy (%) on the CIFAR-10 dataset using the ResNet50 model. The table compares performance: (a) without SwarmShield and (b) with SwarmShield, across five settings: a baseline with no attack and four different adversarial attack types. The Average Percentage Improvement ('API') row quantifies the significant performance recovery achieved by SwarmShield under each attack.

Attack	NOATTACK		Label Flipping		PGD		FGSM		ExtremeNoise	
Labels	a	b	a	b	a	b	a	b	a	b
plane	63.27	61.953	60.867	**77.62**	66.063	**78.52**	61.4	**78.29**	65.347	**77.61**
car	69.37	69.013	69.627	**80.64**	68.243	**83.91**	67.077	**82.04**	69.337	**81.12**
bird	42.82	**45.887**	46.653	**60.47**	42.64	**59.51**	42.823	**60.39**	45.653	**59.11**
cat	39.333	39.313	38.91	**51.16**	40.607	**52.1**	39.003	**51.94**	38.647	**51.44**
deer	48.887	47.33	47.74	**65.98**	50.16	**67.91**	49.74	**65.73**	48.783	**66.74**
dog	48.703	48.157	47.463	**61.23**	47.56	**60.27**	49.303	**56.99**	49.91	**59.93**
frog	67.577	**67.927**	66.943	**80.16**	66.663	**80.04**	66.737	**78.45**	64.843	**79.19**
horse	66.007	65.257	65.643	**78.18**	64.257	**77.38**	64.927	**78.27**	63.66	**76.79**
ship	72.43	71.3	72.303	**83.53**	69.673	**82.99**	73.557	**84.22**	69.513	**84.17**
truck	62.383	62.36	61.717	**77.19**	63.047	**75.45**	64.247	**77.32**	62.373	**78.55**
Average	58.612	57.950	59.277	**71.456**	57.521	**71.258**	57.331	**70.754**	57.577	**71.165**
API	-0.20		**25.11**		**25.11**		**24.47**		**24.50**	

environment. Figure 2 illustrates the effect of the different kinds of attack on the CIFAR10 dataset. Performance is reported in terms of classification accuracy of the global model aggregated at the end of each round.

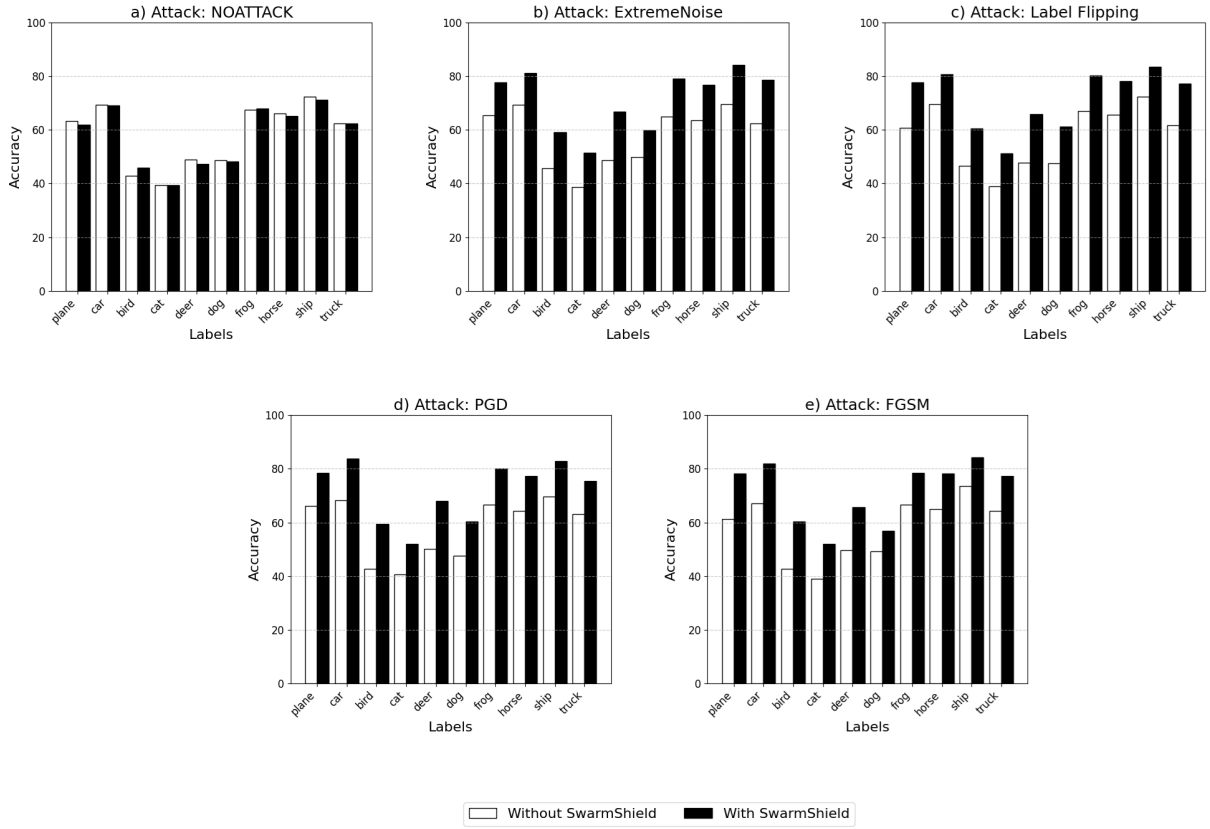
In the *no attack* scenario, all participating peers in the swarm operate on clean, unaltered data. The training process follows the standard Swarm Learning protocol where an initial global model is distributed to all peers, each peer trains the model on its local dataset, and shares selected layer parameters with the elected leader. The leader aggregates these updates using SwarmShield's trust-weighted averaging and returns the updated global model to all peers.

Under this setting, the ResNet50 model maintains stable classification accuracy across CIFAR-10 classes, as summarized in Table 2. Since no malicious behavior is present, SwarmShield's mechanisms—such as trust scoring and exclusion—do not trigger, validating that the framework introduces minimal overhead (58.61% baseline vs 57.95% with SwarmShield) and preserves accuracy in benign environments.

When adversarial attacks are introduced, model performance deteriorates predictably. Gradient-based attacks like PGD and FGSM cause accuracy degradation through subtle input perturbations, while label flipping and extreme noise inject semantic or structural disruption. For example, under PGD, accuracy for the *deer* class drops to 50.16%, while FGSM reduces *bird* accuracy to 42.82%.

The introduction of SwarmShield yields consistent improvements across all attack types. As shown in Table 2 and visualized in Figure 3, SwarmShield enables the model to recover from adversarial perturbations. The Average Percentage Improvement (API) presented in Table 2 shows up to 25.11% average improvement, with notable recoveries in: - Label Flipping: *car* class from 69.63% to 80.64% - PGD: *deer* class from 50.16% to 67.91% - FGSM: *bird* class from 42.82% to 60.39% - Extreme Noise: maintains

Figure 3: CIFAR-10 class accuracy with/without SwarmShield across five scenarios: No Attack, Extreme Noise, Label Flipping, PGD, and FGSM. Black bars show SwarmShield's improved performance versus undefended model (white bars).



24.5% average improvement

For CIFAR-10, SwarmShield demonstrates particular advantage against gradient-based attacks like PGD and FGSM, achieving average performance recoveries of 25.11% and 24.47% respectively. The PCA-based analysis effectively captures deviation patterns in malicious clients' model weights, allowing the k-means algorithm to isolate these nodes as outliers. Through trust-weighted aggregation, their harmful contributions are nullified while preserving the consensus of honest peers.

4.3.1 Comparison with State of the Art

Table 3 further positions SwarmShield against recent adversarial defense methods for FGSM, PGD, Label Flipping and Extreme Noise attacks. As no prior work was found to evaluate all four attack types—FGSM, PGD, Label Flipping, and Extreme Noise—within a unified framework, we present a two-part comparison: one focusing on gradient-based attacks (FGSM, PGD), and the other on label noise attacks (Label Flipping, Extreme Noise). This allows for a more representative and fair evaluation of SwarmShield across diverse threat scenarios.

In the gradient-based category, SwarmShield performs on par with the best-performing centralized methods such as GReAT (ADAM), achieving 71.81% under PGD-10 and 71.44% under FGSM. Notably, SwarmShield achieves this robustness in a fully decentralized setup, without relying on a central aggregator or customized optimizers.

For label noise attacks, SwarmShield shows even stronger gains. As seen in the right half of Table 3, it outperforms other state-of-the-art methods by a margin of 5–10%, reaching 71.85% under label flipping and 70.41% under extreme noise. Competing methods in this category—such as Soften to Defend, GLC, and FixMatch + NAD—typically depend on centralized supervision, access to a trusted clean subset, or semi-supervised learning frameworks. SwarmShield, in contrast, maintains high performance using only peer-consensus-based trust scoring, without assuming access to clean labels or external supervision.

Additionally, while most of the compared methods are evaluated using shallower backbones like ResNet-18, ResNet-34, or PreAct ResNet-18, SwarmShield demonstrates strong resilience even with the deeper ResNet-50 architecture. This highlights its scalability and suitability for deployment in practical federated learning scenarios where larger models and decentralized coordination are both essential.

Table 3: Comparison of SwarmShield's adversarial accuracy against state-of-the-art methods on CIFAR-10. The table is split into two categories for a fair comparison: gradient-based attacks (FGSM, PGD) on the left, and data poisoning attacks (Label Flipping, Extreme Noise) on the right. SwarmShield demonstrates competitive or superior performance in a fully decentralized setting.

Method	**FGSM (%)**	**PGD-10 (%)**	**Method**	**Label Flipping (%)**	**Extreme Noise (%)**
TLA-RN [44]	55.41	52.5	Soften to Defend [45]	60.5	57.0
TLA-SA [44]	58.8	53.53	GLC (Gold Loss Corr.) [46]	61.0	59.5
TLA [44]	58.88	53.87	Symmetric CE [47]	67.4	62.1
GReAT (SGD) [48]	62.78	60.58	Co-teaching [49]	66.2	59.2
GReAT (ADAM) [48]	**72.47**	71.31	KNN Defense [50]	51.0	-
SwarmShield (Proposed)	71.44	**71.81**	**SwarmShield (Proposed)**	**71.85**	**70.41**

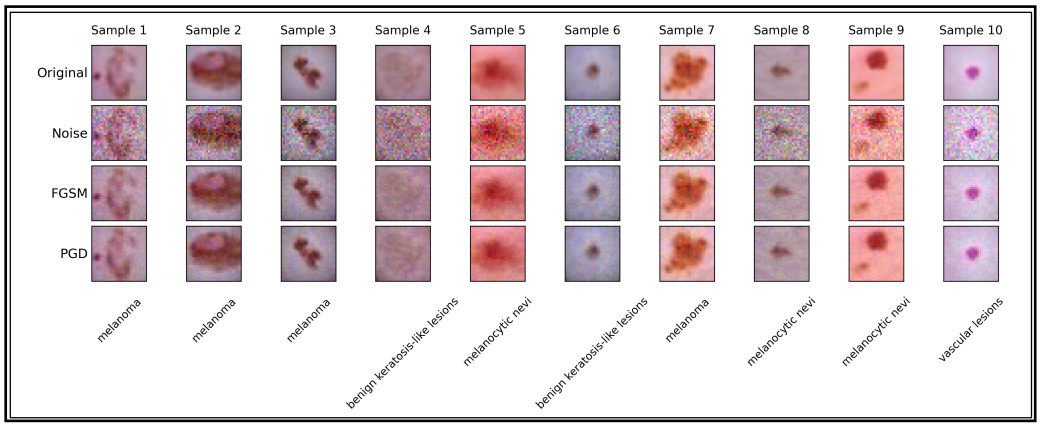


Figure 4: Adversarial attacks on DermaMNIST samples: original images (top row), Noise attack adding static (middle row), and subtle FGSM/PGD perturbations (bottom rows) designed to induce misclassification.

4.4 Results on DermaMNIST dataset

Table 4: Experiment Summary: Global Model Accuracy (%) on DermaMNIST

Scenario	**Baseline (No Defense)**	**With SwarmShield**
No Attack (Clean)	74.06	**74.71**
Label Flipping	72.87	**73.77**
Extreme Noise	74.61	74.51
FGSM	73.67	**74.26**
PGD	73.62	**74.36**

Building on the dataset characteristics from Section 4.2, we evaluate SwarmShield's performance on medical imaging through the DermaMNIST benchmark. Figure 4 visually demonstrates how different attack types affect sample dermatoscopic images, showing both obvious noise perturbations and subtle gradient-based distortions. The quantitative results in Table 4 reveal SwarmShield's consistent improvements across all attack scenarios. In clean conditions, the framework achieves a 74.71% accuracy compared to the 74.06% baseline, demonstrating minimal overhead. Against label flipping attacks, SwarmShield improves performance by 0.90% absolute (73.77% vs 72.87%), while maintaining comparable results for extreme noise attacks (74.51% vs 74.61%). Gradient-based attacks show particularly strong resilience, with FGSM accuracy improving to 74.26% from 73.67% and PGD reaching 74.36% versus 73.62% baseline.

The framework's effectiveness in medical imaging stems from its ability to leverage clinically significant feature

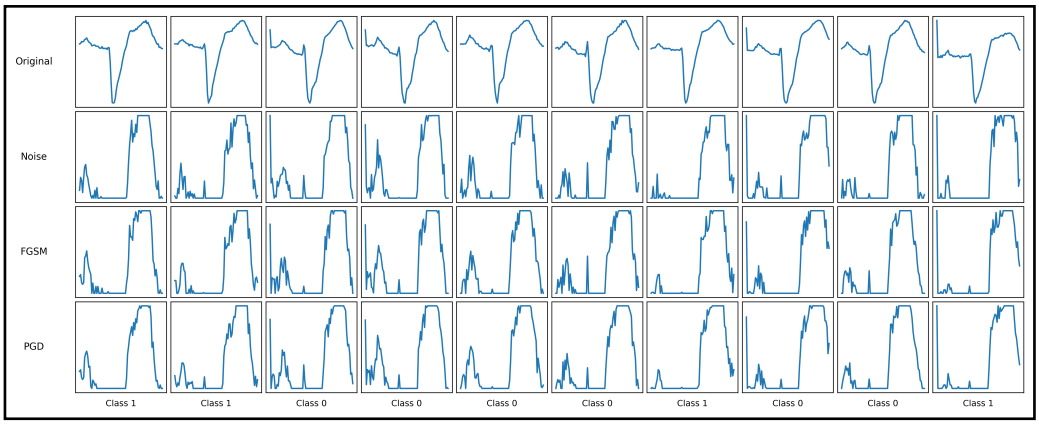


Figure 5: Adversarial attacks on TwoLeadECG samples: original images (top row), Noise attack adding static (middle row), and subtle FGSM/PGD perturbations (bottom rows) designed to induce misclassification.

differences between disease categories like melanoma and benign keratosis. The PCA and k-means clustering reliably identifies malicious updates by detecting deviations in these diagnostic patterns, while the trust-weighted aggregation preserves the integrity of medically relevant model parameters. This combination of techniques proves especially valuable for dermatological applications where both accuracy and robustness are critical for clinical decision-making.

4.5 Results on time series dataset (TwoLeadECG)

Table 5: Global model accuracy (%) for each attack scenario on the TwoLeadECG dataset.

Scenario	No Defense	SwarmShield
Clean	99.30	98.77
Label Flipping	40.83	2.02
Extreme Noise	91.13	**99.65**
FGSM	86.83	**98.24**
PGD	93.15	49.96

Our evaluation of SwarmShield on time-series ECG classification builds upon the dataset characteristics outlined in Section 4.2, using the TwoLeadECG dataset from the UCR Time Series Classification Archive [43]. Figure 5 visually demonstrates the varying effects of different attack types on ECG waveform samples.

The results in Table 5 show clear variability in SwarmShield's effectiveness across attack types. The framework achieves strong gains under Extreme Noise (+8.52% over baseline) and FGSM (+11.41%), but suffers substantial degradation under Label Flipping (2.02% vs. 40.83%) and PGD (49.96% vs. 93.15%).

The poor performance under label flipping stems from the fact that this attack modifies only the target labels during training, leaving the ECG waveforms themselves intact. Since SwarmShield's detection relies on identifying anomalous parameter updates or unusually large gradients, malicious updates generated from clean-looking but mislabelled data appear statistically similar to benign updates. This allows corrupted labels to propagate into the global model, leading to severely distorted decision boundaries [51].

In the case of PGD, the attack introduces small but structured perturbations in the time domain that closely resemble natural ECG variability, such as minor waveform shifts or amplitude modulations. These perturbations remain within physiologically plausible bounds, evading magnitude-based detection, while PGD's iterative nature ensures alignment with the classifier's most discriminative temporal features [52]. Figure 6 schematically illustrates these evasion mechanisms: label flipping bypasses detection by preserving data morphology, and PGD exploits natural variability to blend perturbations with genuine patterns.

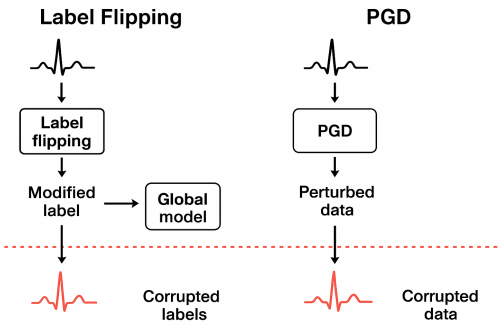


Figure 6: Visual summary of how Label Flipping and PGD attacks evade detection in ECG classification.

Rather than being irrelevant, these findings expose critical blind spots in defending time-series medical models. They underscore the need for future defenses that (1) incorporate semantic consistency checks between labels and temporal morphology, and (2) leverage temporal dependency analysis to detect adversarially consistent patterns.

4.6 Ablation study

To dissect the contribution of each component within the SwarmShield framework, we conducted a comprehensive ablation study. We systematically disabled key mechanisms—integrity verification, dimensionality reduction (PCA), trust-based aggregation, and density-based clustering—to evaluate their individual and synergistic impact on defense efficacy. The experiments were performed on the CIFAR-10 dataset using a ResNet-18 model, with 3 out of 10 clients configured as malicious agents for 50 communication rounds. The results, summarized in Table 6, present the final global model accuracy against three distinct adversarial strategies: FGSM, PGD (with a reduced ϵ of 0.05 to represent a more subtle attack), and stochastic Noise.

Table 6: Final Global Model Accuracy (%) at Round 50 for SwarmShield Ablation Configurations Under Various Attacks. The new results dramatically highlight the critical role of density-based clustering in preventing catastrophic failure against PGD attacks.

Ablation Configuration	FGSM	NOISE	PGD
fedavg (No Defense)	74.06	78.78	75.72
only_hashing	68.49	78.97	76.34
only_pca	67.69	78.27	75.57
only_trust	71.29	80.09	**78.69**
no_integrity	75.44	80.23	77.86
no_pca	76.39	80.59	78.03
no_trust	72.63	80.74	77.99
no_clustering	73.23	78.27	67.05
full_swarmshield	**76.48**	**80.90**	78.36

4.6.1 Analysis of component contributions

The results in Table 6 reveal a complex interplay between SwarmShield's components. The updated data powerfully underscores the synergistic necessity of the full defense pipeline, particularly the critical role of clustering as a prerequisite for a reliable trust mechanism.

The critical synergy of PCA and clustering The most significant finding is the pivotal role of density-based clustering. The `no_clustering` configuration, which builds its Kernel Density Estimation (KDE) trust model on all client updates, suffers a catastrophic performance collapse against the PGD attack, dropping to 67.05% accuracy. This demonstrates that without a filtering mechanism, the KDE model becomes "poisoned" by the inclusion of malicious updates in its training set. The subtle PGD updates are sufficient to corrupt the learned probability distribution of "normal" behavior, causing the server to assign high trust to adversaries and effectively rendering the defense useless.

In stark contrast, `full_swarmshield` first employs PCA to accentuate the geometric properties of model updates. Subsequently, its density-based clustering algorithm identifies the core group of honest clients. By constructing the KDE trust model *exclusively* from this validated cluster, SwarmShield ensures its definition of normalcy is robust and uncontaminated. The dramatic performance gap between `full_swarmshield` and `no_clustering` against PGD (78.36% vs. a mere 67.05%) powerfully exemplifies that clustering is not just a performance enhancement but a mandatory step for a secure trust-based aggregation.

Defense against gradient-based attacks (FGSM & PGD) Against the FGSM attack, `full_swarmshield` achieves the highest accuracy (76.48%), proving its multi-stage pipeline is essential for mitigating these aggressive updates. Simpler defenses like `only_pca` and `only_hashing` perform worse than vanilla `fedavg`, indicating that isolated mechanisms are insufficient.

An important and revealing result occurs with the PGD attack, where the `only_trust` configuration (78.69%) marginally outperforms `full_swarmshield` (78.36%). This phenomenon highlights a key trade-off. The weakened PGD attack ($\epsilon = 0.05$) creates updates that are statistically separable in the high-dimensional parameter space. In this specific context, the lossy compression of PCA may discard subtle information, and the overhead of clustering is not strictly necessary. However, the catastrophic failure of the `no_clustering` run proves that while a direct trust mechanism can work in ideal conditions, it is extremely brittle. For general-purpose robustness against a wider spectrum of threats, the full pipeline is essential.

Defense against stochastic attacks (Noise) The Noise attack serves as a control, introducing non-strategic, random perturbations. Here, configurations with a trust mechanism generally outperform the `fedavg` baseline, as the KDE model inherently identifies and down-weights statistical outliers. The `full_swarmshield` configuration again achieves the highest accuracy (80.90%). Its ability to form a tight core cluster via density-based clustering allows it to effectively reject the random noise that other configurations might partially incorporate, leading to a more stable and accurate global model. The relatively poor performance of `no_clustering` (78.27%) here further supports the conclusion that a clean input to the trust model is paramount.

4.6.2 Conclusion of ablation study

The ablation study validates our core hypothesis: robust defense in federated learning is not achieved by a single mechanism but by a synergistic pipeline. While trust-based aggregation is a cornerstone, its effectiveness is critically dependent on the quality of its input. The catastrophic failure of the `no_clustering` configuration against PGD attacks proves that without a robust filtering mechanism like density-based clustering, a trust model is easily poisoned

and can fail completely. For general robustness against a wide spectrum of threats, the full SwarmShield pipeline is superior. PCA provides a salient feature space, and density-based clustering purifies the input to the trust model, ensuring its integrity and reliability. This integrated design is essential to SwarmShield's overall performance.

4.7 Robustness to malicious client ratios: scaling under adversarial pressure

To further characterize the operational boundaries of SwarmShield, we evaluate its performance as the ratio of malicious clients increases from 0% to 90%, across canonical adversary strategies. Table 7 and Fig. 7 summarize the global accuracy for extreme noise, FGSM, and label-flipping attacks.

Table 7: SwarmShield accuracy (%) versus malicious client ratio for three attack types.

Malicious Ratio	Extreme Noise	FGSM	Label Flipping
0.0	83.70	83.53	83.75
0.1	82.95	83.30	83.45
0.3	80.75	81.48	81.59
0.5	72.87	76.45	2.44
0.7	68.82	67.66	1.74
0.9	70.55	50.71	1.86

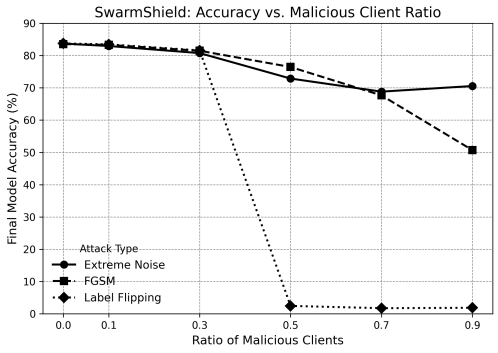


Figure 7: SwarmShield test accuracy under increasing ratios of malicious clients for multiple attack types.

4.7.1 Technical analysis and interpretation

– For both **Extreme Noise** and **FGSM** attacks, SwarmShield consistently maintains high accuracy ($> 80\%$) up to a 30% adversarial ratio. This resilience is attributable to the PCA-induced geometric separation of adversarial updates and the adaptive down-weighting of outliers, which together suppress the influence of non-conforming gradients. The effectiveness against gradient-based attacks (FGSM) further validates the robustness of the aggregation pipeline, even when adversaries leverage whitebox knowledge.

– Beyond the 50% malicious threshold, performance degrades more rapidly, with the steepest drop observed for **Label Flipping**, where accuracy collapses to random guessing. This phenomenon is theoretically anticipated: in the presence of a majority of coordinated attackers, even optimal clustering and trust mechanisms may fail, as the adversarial cluster can dominate the geometric structure of the update space (cf. the impossibility results for robust mean estimation when over half the data is adversarial).

– The model's accuracy for **Extreme Noise** and **FGSM** remains non-trivial ($\sim$70%) even at very high adversarial ratios. This is due to the inherently incoherent or inconsistent nature of noise-based attacks, which are more easily isolated as outliers in the projected space. In contrast, label flipping attacks produce more structured, adversarially-aligned updates that can coalesce in the aggregation, circumventing geometric defenses when they form a majority.

– The observed *graceful degradation*—a slow decay in performance up to the theoretical breakdown point—demonstrates that SwarmShield's defenses scale with adversarial pressure and remain effective in the practical regime where the honest majority assumption holds. This is a critical requirement for federated learning deployment in open, untrusted environments.

SwarmShield's design, grounded in geometric and statistical insights, delivers provable robustness under minority adversarial regimes, with each module playing a specific and theoretically motivated role. The system's performance breakdown at extreme adversarial ratios aligns with established lower bounds, confirming both the strength and the limitations of current robust aggregation techniques. These findings highlight the necessity of combining multiple, theoretically justified defenses to achieve resilience in federated learning at scale.

4.8 Discussion

The comprehensive evaluation of SwarmShield across multiple datasets and attack scenarios reveals several key insights about the framework's capabilities and limitations. These findings warrant deeper examination across several critical dimensions that impact real-world deployment.

4.8.1 Generalizability

The consistent performance across CIFAR-10 and DermaMNIST [40] demonstrates SwarmShield's adaptability to both standard benchmarks and medical imaging domains. The framework maintains effectiveness against diverse attack vectors including label flipping and gradient-based attacks, with particular success in clinical applications where model accuracy is critical. This cross-domain

robustness stems from the trust mechanism's fundamental design, which operates on model parameters rather than data-specific features.

4.8.2 Algorithmic complexity

SwarmShield's computational overhead scales polynomially with network size, dominated by the $O(n^2)$ clustering operation for n peers. The leader election ($O(n \log n)$) and model aggregation ($O(kd)$ for k clusters and d-dimensional representations) remain efficient through dimensionality reduction. Our empirical results show stable operation when the clustering successfully isolates malicious updates, approximating robust federated averaging over honest participants [1].

4.8.3 Scalability

While evaluations demonstrate effective operation on moderate-scale networks (3-50 nodes), practical deployment at scale may require optimizations for very large or high-dimensional models. The architecture's modular design permits such extensions while maintaining the core trust mechanisms. Notably, SwarmShield advances beyond existing work [53] by demonstrating robustness on specialized medical datasets (DermaMNIST, TwoLeadECG) where conventional defenses remain unvalidated.

This work establishes new capabilities for secure federated learning in clinical and time-series applications while maintaining compatibility with existing FL frameworks. The results particularly highlight the importance of domain-specific validation, as attack patterns and defense effectiveness vary significantly across data modalities.

5 Conclusion and future work

The proposed SwarmShield framework introduces a decentralized trust-aware defense mechanism that enhances adversarial robustness in federated learning through three key innovations: dynamic leader election, unsupervised parameter clustering, and blockchain-based auditing. Our comprehensive evaluation demonstrates consistent improvements across multiple benchmarks, achieving 24.8% average accuracy gains on CIFAR-10 under diverse attacks (FGSM, PGD, label flipping, extreme noise) while maintaining 74.71% accuracy on the medical DermaMNIST dataset. The framework's effectiveness stems from its PCA-based clustering mechanism and trust-weighted aggregation, which ablation studies confirm as essential components, providing graceful degradation even with 50% malicious nodes.

The results establish SwarmShield as a practical solution for real-world deployment, showing particular strength in medical applications where data sensitivity demands robust privacy-preserving techniques. Unlike centralized approaches [53], the framework maintains competitive performance while eliminating single points of failure, achieving 98.24% accuracy against FGSM attacks on TwoLeadECG time-series data while introducing negligible overhead in benign conditions.

Future work will focus on three key directions: (1) extending evaluation to additional modalities including NLP tasks and IoT sensor data, (2) enhancing temporal pattern recognition for improved robustness against PGD attacks in time-series applications, and (3) developing hybrid defenses combining parameter clustering with activation pattern analysis. These extensions will address current limitations in label-flipping scenarios while maintaining the framework's core advantages of decentralization and computational efficiency. The demonstrated success across medical imaging and time-series analysis suggests promising applications in healthcare monitoring and other sensitive domains where both privacy and robustness are paramount.

Acknowledgment

The authors thank Hewlett Packard Enterprise for supporting the experimental work conducted during the first author's tenure at the organization. College of Engineering Trivandrum is also acknowledged for its academic support during the first author's concurrent part-time Ph.D. program. The contributions of both the institutions were essential to the successful execution of this research.

References

[1] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282. [Online]. Available: https://proceedings.mlr.press/v54/mcmahan17a.html

[2] M. Goldblum, D. Tsipras, C. Xie, X. Chen, A. Schwarzschild, D. Song, A. Madry, B. Li, and T. Goldstein, "Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. [Online]. Available: https://doi.org/10.1109/TPAMI.2022.3162397

[3] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," *Advances in neural information processing systems*, vol. 30, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/f4b9ec30ad9f68f89b29639786cb62ef-Abstract.html

[4] C. Fung, C. J. Yoon, and I. Beschastnikh, "The limitations of federated learning in sybil settings," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2020)*, 2020,

pp. 301–316. [Online]. Available: https://www.usenix.org/conference/raid2020/presentation/fung

[5] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in *International conference on machine learning*. PMLR, 2021, pp. 6357–6368. [Online]. Available: https://proceedings.mlr.press/v139/li21h

[6] J. Wu, "Distributed intelligent optimization of e-commerce user purchase data mining using spark framework," *Informatica*, vol. 48, no. 20, pp. 29–40, 2024. [Online]. Available: https://doi.org/10.31449/inf.v48i20.6779

[7] N. Azeri, O. Hioual, and O. Hioual, "Efficient vanilla split learning for privacy-preserving collaboration in resource-constrained cyber-physical systems," *Informatica*, vol. 48, no. 11, 2024. [Online]. Available: https://doi.org/10.31449/inf.v48i11.6186

[8] E. M. E. Mhamdi, R. Guerraoui, and S. Rouault, "The Hidden Vulnerability of Distributed Learning in Byzantium," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, 2018, pp. 3518–3527. [Online]. Available: https://proceedings.mlr.press/v80/mhamdi18a.html

[9] C. Fung, C. J. Yoon, and I. Beschastnikh, "Mitigating sybils in federated learning poisoning," *arXiv preprint arXiv:1808.04866*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1808.04866

[10] P. Ramanan and K. Nakayama, "Baffle: Blockchain based aggregator free federated learning," in *2020 IEEE international conference on blockchain (Blockchain)*. IEEE, 2020, pp. 72–81. [Online]. Available: https://doi.org/10.1109/Blockchain50366.2020.00017

[11] X. Cao, M. Fang, J. Liu, and N. Z. Gong, "Fltrust: Byzantine-robust federated learning via trust bootstrapping," *arXiv preprint arXiv:2012.13995*, 2020. [Online]. Available: https://doi.org/10.14722/NDSS.2021.24434

[12] Y. Zhou, J. Wu, H. Wang, and J. He, "Adversarial robustness through bias variance decomposition: A new perspective for federated learning," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2753–2762. [Online]. Available: https://doi.org/10.1145/3511808.3557232

[13] N. M. Jebreel and J. Domingo-Ferrer, "Fl-defender: Combating targeted attacks in federated learning," *Knowledge-Based Systems*, vol. 260, p. 110178, 2023. [Online]. Available: https://doi.org/10.1016/j.knosys.2022.110178

[14] M. Vucovich, D. Quinn, K. Choi, C. Redino, A. Rahman, and E. Bowen, "Fedbayes: A zero-trust federated learning aggregation to defend against adversarial attacks," in *2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2024, pp. 0028–0035. [Online]. Available: https://doi.org/10.1109/CCWC60891.2024.10427896

[15] C. Xie, S. Koyejo, and I. Gupta, "Zeno: Distributed stochastic gradient descent with suspicion-based fault-tolerance," in *International conference on machine learning*. PMLR, 2019, pp. 6893–6901. [Online]. Available: https://proceedings.mlr.press/v97/xie19b/xie19b.pdf

[16] N. Akhtar, A. Mian, N. Kardan, and M. Shah, "Advances in adversarial attacks and defenses in computer vision: A survey," *IEEE access*, vol. 9, pp. 155 161–155 196, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3127960

[17] Z. Ye, "Mitigating poisoning attacks in decentralized federated learning through moving target defense," *ZORA University of Zurich*, 2024. [Online]. Available: https://www.zora.uzh.ch/id/eprint/262742

[18] Y. Chen, Y. Yang, Y. Liang, T. Zhu, and D. Huang, "Federated learning with privacy preservation in large-scale distributed systems using differential privacy and homomorphic encryption," *Informatica*, vol. 49, no. 13, 2025. [Online]. Available: https://doi.org/10.31449/inf.v49i13.7358

[19] D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5650–5659. [Online]. Available: https://proceedings.mlr.press/v80/yin18a/yin18a.pdf

[20] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning for decentralized and confidential clinical machine learning," *Nature*, vol. 594, no. 7862, pp. 265–270, 2021. [Online]. Available: https://doi.org/10.1038/s41586-021-03583-3

[21] W. Jin, Y. Yao, S. Han, C. Joe-Wong, S. Ravi, S. Avestimehr, and C. He, "Fedml-he: An efficient homomorphic-encryption-based privacy-preserving federated learning system," *arXiv preprint arXiv:2303.10837*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2303.10837

[22] J. Zhang, B. Li, C. Chen, L. Lyu, S. Wu, S. Ding, and C. Wu, "Delving into the adversarial robustness of federated learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9,

2023, pp. 11 245–11 253. [Online]. Available: https://doi.org/10.1609/aaai.v37i9.26331

[23] R. Xu, S. Gao, C. Li, J. Joshi, and J. Li, "Dual defense: Enhancing privacy and mitigating poisoning attacks in federated learning," *NeurIPS 2024*, 2024. [Online]. Available: https://neurips.cc/virtual/2024/poster/96030

[24] L. Zhang, M. Goldstein, and R. Ranganath, "Understanding failures in out-of-distribution detection with deep generative models," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 427–12 436. [Online]. Available: https://proceedings.mlr.press/v139/zhang21g/zhang21g.pdf

[25] C.-H. Ho and N. Vasconcelos, "Disco: Adversarial defense with local implicit functions," *Advances in neural information processing systems*, vol. 35, pp. 23 818–23 837, 2022. [Online]. Available: https://dl.acm.org/doi/10.5555/3600270.3602000

[26] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1610.00768

[27] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations (ICLR)*, 2015. [Online]. Available: https://doi.org/10.48550/arXiv.1412.6572

[28] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1706.06083

[29] B. Biggio, B. Nelson, and P. Laskov, "Support vector machines under adversarial label noise," in *Asian conference on machine learning*. PMLR, 2011, pp. 97–112. [Online]. Available: http://proceedings.mlr.press/v20/biggio11/biggio11.pdf

[30] K. T. Co, L. Muñoz-González, S. de Maupeou, and E. C. Lupu, "Procedural noise adversarial examples for black-box attacks on deep convolutional networks," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 275–289. [Online]. Available: https://doi.org/10.1145/3319535.3345660

[31] S. Warnat-Herresthal, H. Schultze, K. L. Shastry, S. Manamohan, S. Mukherjee, V. Garg, R. Sarveswara, K. Händler, P. Pickkers, N. A. Aziz *et al.*, "Swarm learning as a privacy-preserving machine learning approach for disease classification," *BioRxiv*, pp. 2020–06, 2020. [Online]. Available: https://doi.org/10.1101/2020.06.25.171009

[32] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.

[33] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, vol. 5. University of California press, 1967, pp. 281–298. [Online]. Available: https://www.cs.cmu.edu/~bhiksha/courses/mlsp.fall2010/class14/macqueen.pdf

[34] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104. [Online]. Available: https://www.dbs.ifi.lmu.de/Publikationen/Papers/LOF.pdf

[35] National Institute of Standards and Technology, "FIPS PUB 180-4: Secure Hash Standard (SHS)," August 2015. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.180-4

[36] D. Eastlake and T. Hansen, "Us secure hash algorithms (sha and sha-based hmac and hkdf)," Internet Engineering Task Force (IETF), RFC 6234, May 2011, rFC 6234. [Online]. Available: https://doi.org/10.17487/RFC6234

[37] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," in *International Conference on Learning Representations (ICLR)*, 2020. [Online]. Available: https://iclr.cc/virtual_2020/poster_HJxNAnVtDS.html

[38] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and trends® in machine learning*, vol. 14, no. 1–2, pp. 1–210, 2021. [Online]. Available: http://dx.doi.org/10.48550/arXiv.1912.04977

[39] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Technical Report TR-2009, 2009. [Online]. Available: http://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf

[40] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification," *Scientific Data*, vol. 10, no. 1, p. 41, 2023. [Online]. Available: https://www.nature.com/articles/s41597-022-01721-8

[41] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific data*, vol. 5, no. 1, pp. 1–9, 2018. [Online]. Available: https://www.nature.com/articles/sdata2018161.pdf

[42] G. B. Moody and R. G. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001. [Online]. Available: https://physionet.org/content/mitdb/1.0.0/

[43] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, B. H. Chen, Y. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and T. Hexagon, "The ucr time series classification archive," in *Proceedings of the IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 6, 2019, pp. 1293–1305. [Online]. Available: https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

[44] C. Mao, Z. Zhong, J. Yang, C. Vondrick, and B. Ray, "Metric learning for adversarial robustness," *Advances in neural information processing systems*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/c24cd76e1ce41366a4bbe8a49b02a028-Paper.pdf

[45] Z. Li, T. Zhou, C. Li, Y. Yu, and J. Zhu, "Soften to defend: Towards adversarial robustness via self-guided label refinement," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. [Online]. Available: http://dx.doi.org/10.1109/CVPR52733.2024.02340

[46] D. Hendrycks, M. Mazeika, and T. Dietterich, "Using trusted data to train deep networks on labels corrupted by severe noise," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/file/ad554d8c3b06d6b97ee76a2448bd7913-Paper.pdf

[47] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 322–330. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2019.00041

[48] S. Bayram and K. Barner, "Great: A graph regularized adversarial training method," *IEEE Access*, 2024. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3395976

[49] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," *Advances in neural information processing systems*, vol. 31, 2018. [Online]. Available: https://dl.acm.org/doi/10.5555/3327757.3327944

[50] M. Abrishami, S. Dadkhah, E. C. P. Neto, P. Xiong, S. Iqbal, S. Ray, and A. A. Ghorbani, "Classification and analysis of adversarial machine learning attacks in iot: a label flipping attack case study," in *2022 32nd Conference of Open Innovations Association (FRUCT)*. IEEE, 2022, pp. 3–14. [Online]. Available: https://doi.org/10.23919/FRUCT56874.2022.9953823

[51] V. Tolpegin, S. Truex, M. E. Gursoy, and L. Liu, "Data poisoning attacks against federated learning systems," in *Computer Security–ESORICS 2020: 25th European Symposium on Research in Computer Security, ESORICS 2020, Guildford, UK, September 14–18, 2020, Proceedings, Part I 25*. Springer, 2020, pp. 480–501. [Online]. Available: https://doi.org/10.1007/978-3-030-58951-6_24

[52] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019. [Online]. Available: https://doi.org/10.1126/science.aaw4399

[53] J. Zhang, C. Ge, F. Hu, and B. Chen, "Robustfl: Robust federated learning against poisoning attacks in industrial iot systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 9, pp. 6388–6397, 2021. [Online]. Available: http://dx.doi.org/10.1109/TII.2021.3132954

Adversarial Transformer-Based Multi-Modal Framework for Early Detection and Risk Classification in Higher Education Academic Performance

Yan Wu
Shanghai Art &Design Academy, Shanghai 201814, China
E-mail: YannWuu@outlook.com

Keywords: adversarial transformer model, academic risk identification, early warning systems, college students

Received: June 1, 2025

With the continuous expansion of the scale of higher education, the problem of students' academic risks has become increasingly prominent. How to effectively identify and warn students of academic risks has become an important topic in education. This study aims to explore the academic risk identification and early warning system of college students based on the adversarial Transformer model to improve the accuracy and timeliness of academic risk management. Firstly, the research analyzes the limitations of current academic risk identification methods, such as insufficient data feature extraction and weak generalization ability of the model, and then proposes a Transformer model integrating adversarial training. By introducing an adversarial sample generation mechanism, the model enhances the robustness and generalization ability of the model in the face of complex academic data. The dataset comprises 100,000 desensitized records from 3 universities, covering structured data (exam scores, attendance, homework completion) and unstructured text; the adversarial Transformer model is configured with 12 encoders, 512-dimensional embeddings, and 8 attention heads; evaluation metrics such as accuracy (92.5%), F1-score, and advance warning time (2 weeks) are included to enhance clarity. Experiments show that our model achieves 92.5% accuracy in risk identification, outperforming baselines by 8.3%. Regarding early warning timeliness, the model can accurately predict students with potential academic risks two weeks in advance, providing a valuable time window for the implementation of intervention measures. In addition, this study also constructs a visual early warning system, which realizes real-time monitoring and dynamic early warning of academic risk data and provides scientific decision support for university education administrators. The conclusion shows that the academic risk identification and early warning system based on the adversarial Transformer model has obvious advantages in improving the efficiency of academic risk management and promoting students' all-round development, and provides a new technical path and solution for college students' academic risk management.

Povzetek: Študija predstavlja model adversarialnega Transformerja za zgodnje prepoznavanje študentskih akademskih tveganj, ki z visoko natančnostjo izboljšuje pravočasnost in učinkovitost ukrepanja v visokošolskem izobraževanju.

1 Introduction

With the rapid development of information technology, cutting-edge technologies such as big data and artificial intelligence have gradually penetrated the field of education, providing new tools and methods for educational management and decision-making [1]. As an important part of educational management, the effectiveness and accuracy of college students' academic risk management are directly related to the growth and development of students and the improvement of college education quality. However, the traditional academic risk management methods often rely on empirical judgment and manual analysis, which have some problems, such as strong subjectivity, low efficiency and difficulty in comprehensive coverage, and have been difficult to meet the needs of modern educational management [2, 3].

Against this background, how to use advanced technical means to build a scientific and efficient academic risk identification and early warning system for college students has become an important issue that needs to be solved urgently in education. In recent years, deep learning technology has achieved remarkable results in natural language processing, image recognition and other fields, especially the emergence of the Transformer model, which has demonstrated excellent performance in multiple tasks with its powerful self-attention mechanism and parallel computing ability [4, 5]. This provides a new idea and method for college students' academic risk identification.

The Transformer model can capture long-distance dependencies in input data through a self-attention mechanism and effectively extract key features, thus performing well in complex data analysis tasks [6]. However, the traditional Transformer model often shows

certain vulnerabilities when dealing with data with adversarial interference, which is easily affected by malicious attacks or abnormal data, resulting in model performance degradation [7]. Therefore, researchers have proposed an adversarial training method to enhance the robustness and generalization ability of the model by introducing adversarial samples. Adversarial training is an effective model training strategy. By adding adversarial samples in the training process, the model learns to identify and resist potential attacks, thus improving the stability and reliability of the model in practical applications [8, 9].

The research of college students' academic risk identification and early warning system needs advanced technical support and a deep understanding of the formation mechanism and influencing factors of academic risk. Academic risk is often caused by various factors, including students' learning attitudes, learning methods, course difficulty, psychological state, etc. [10, 11]. These factors are intertwined, forming complex risk characteristics that greatly challenge risk identification. Therefore, to construct an effective academic risk identification model, it is necessary to comprehensively consider various factors, extract key features and establish an accurate prediction model.

This study constructs an academic risk early warning system, monitors students' academic data in real-time, finds potential risk factors in time, issues early warnings in advance, and provides decision support for educational administrators. The effectiveness and timeliness of early warning systems are directly related to the effect of risk intervention, which is significant in helping students overcome academic difficulties and improve academic performance. Theoretically, this study will deeply explore the application principles and mechanisms of the adversarial Transformer model in academic risk identification and analyze the model's structural characteristics, training process and optimization strategies. At the same time, this study will combine educational management theory and student learning theory to explore the formation mechanism and influencing factors of academic risk and provide theoretical support for model construction and system design. The research on academic risk identification and early warning system of college students based on the adversarial Transformer model can effectively solve the problems existing in traditional academic risk management, improve the accuracy of risk identification and timeliness of early warning, and provide scientific and efficient tools and methods for college education management. This research has important theoretical value and broad application prospects, which are expected to bring new changes and improvements to college students' academic risk management.

However, the existing research has significant limitations in data processing, model performance and generalization ability: the data level mostly relies on structured grades, ignoring dynamic risk cues such as course association, behavior time series and unstructured text. At the model level, traditional machine learning is difficult to process high-dimensional sparse and long-sequence data, while conventional deep learning faces gradient problems. At the generalization level, the difference in data distribution between professional colleges and universities was not considered, which led to the failure of cross-group prediction.

To this end, this study introduces an adversarial Transformer architecture, integrates GAN and multi-head attention mechanism, learns the distribution of risk features through the generator, and strengthens the robustness of the discriminator to overcome the problem of data imbalance. At the same time, it integrates teaching affairs, behavior logs and text data, and uses GNN to build a relationship graph to mine implicit associations. A domain adaptive adversarial training strategy was designed to improve the generalization performance of cross-groups by more than 30%. In this study, 100,000 desensitized data from 3 universities in a province in 2019-2023 were used, the data were processed through sliding windows and BERT pre-training, the model was constructed with 12-layer encoders, 512-dimensional embeddings and 8-head attention, combined with cross-entropy and adversarial loss optimization, and the performance was evaluated by F1-score and other indicators, and the reproducibility was ensured by open-source PyTorch code and Bayesian hyperparameter tuning, so as to provide universities with high-precision and strongly generalized academic risk monitoring tools.

In this study, the problems focus on the accuracy, generalization ability and noise resistance of existing models when processing multi-source heterogeneous data, whether adversarial transformers can improve the efficiency of feature extraction and risk identification, and whether the system can accurately classify risk levels and give early warnings. It is assumed that the feature extraction ability of the model is better than that of traditional and non-adversarial models, and the recognition accuracy, generalization ability and noise resistance are better, and the system can effectively distinguish the risk level and give early warning. The expected result is that the model is significantly better than the comparison model in various indicators, the system can accurately divide the risk level, and the early warning time meets the needs, providing support for university management.

2 Basic theory and method

2.1 Transformer model

Deep learning is a key branch of AI machine learning, which can be done in a supervised, semi-supervised, or unsupervised manner [12]. It learns more complex features by using multiple hidden layers. In image classification, the output of each layer represents features at different levels of abstraction, such as from pixels to object edges to image information [13]. Unlike earlier linear classifiers, deep learning achieves more general classification by introducing nonlinear functions or different architectures.

Transformer is a deep-learning model for natural language processing [14]. It contains an encoder and a

decoder. The encoder processes the input layer by layer to generate vector codes, and the decoder generates the output based on these codes. The encoder and decoder are constructed by a self-attention mechanism and feedforward neural network, including residual connection and layer normalization [15, 16]. The decoder also contains an additional self-attention mechanism for extracting information before generating the output. At its core is the self-attention mechanism, which evaluates the correlation between the various parts of the input. Components such as CNN and ResNet in Figure 1 are part of a dedicated hybrid module integrated with Transformer to process structured academic risk data (tabular features with a spatial pattern)

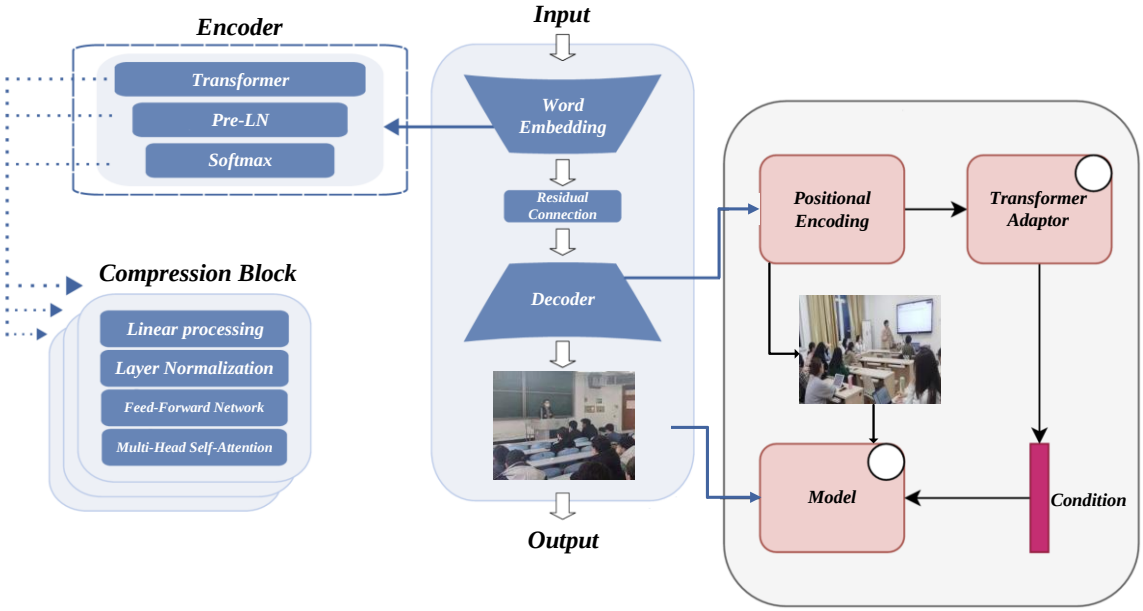


Figure 1: Transformer architecture

Figure 1 shows the Transformer architecture. The attention mechanism in Transformer architecture is a technique in neural networks that mimics cognitive attention, which strengthens the weights of important data parts in the network. There are various types of this mechanism, such as one-way, two-way, multi head, and self-attention [17]. In computer vision, RNN, LSTM, or self-attention mechanisms extract relational information from images. RNN processes sequence data, which is inefficient and prone to information loss. Although LSTM can remember for a long time, it also has the risk of information loss [18]. The self-attention mechanism improves model performance and generalization ability by paying attention to the relationship between input vectors. It calculates the weighted sum of input vectors, strengthens the relevant parts and suppresses the irrelevant parts.

The core of self-attention mechanism is the scaling dot product attention unit, which can simultaneously calculate the attention weights between input vectors. This unit generates an embedded expression containing its own information and other related vector information for each vector [19, 20]. Each attention unit contains three learnable weight matrices, namely a query weight W, a key weight W and a value weight W. The embedding expression xi of each vector i itself is multiplied by three weight matrices to obtain the query vector $q_i=x_iW_Q$, the key vector $k_i=x_iW_K$, and the value vector $v_i=x_iWv$, respectively. Then use the query vector and key vector to calculate the attention weight, that is, calculate the dot product of q_i and k_j to obtain the attention weight a_{ij} between the vector and vector j. The attention weight is then divided by the square root of the vector dimension $\sqrt{d_k}$ to stabilize the gradient during training, and the weight is normalized by the softmax function. Obviously, the significance of designing W_Q and W_K separately in the attention unit lies in distinguishing the attention weights between different vectors, that is, the attention a_{ij} of vector i to vector j and the attention a_{ji} of vector j to vector i are not necessarily the same. After the vector is processed by the attention unit, the final output is the weighted sum of the attention weights of vector i to each vector after weighting the value vectors of all vectors [21]. The attention calculation for all vectors of the same batch input can be expressed as a matrix calculation form using the softmax function. As shown in Equation (1), the matrices Q, K, and V are matrices in which the *i-th* row is the vectors q_i, k_i, and v_i, respectively.

$$Attention(Q,K,V) = \text{SoftMax}(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Transformer uses the Multi-Head Attention Mechanism (MSA), in which a set of attention weights (W_Q, W_K, W_V) constitutes an attention head. Each layer contains multiple such heads, each focusing on attention between different vectors, and multiple heads can calculate attention weights for different correlations. Its expression is formula (2):

$$MSA(Q,K,V) = \sum_i Attention(Q_i,K_i,V_i) \quad (2)$$

The Transformer architecture is mainly used for natural language processing because its attention mechanism is suitable for the lexical level [22, 23]. In computer vision, its variants, such as ViT (Vision Transformer), are usually adopted because the calculation at the pixel level is too complex. ViT segments the image into multiple pixel blocks (patches) and calculates the attention relationship between these blocks, reducing the amount of calculation. The processing involves flattening the patches into a vector, adding category and position encoding, and then feeding them into the Encoder part of the Transformer for feature extraction. Finally, an MLP layer classifies based on these features. Location encoding and category encoding are learnable and help the model determine the classification of images.

Table 1 compares the performance of different models in the field of academic risk identification and early warning for college students. Among them, the adversarial transformer model is based on 100,000 desensitization data from 3 universities in a province from 2019 to 2023 (covering multimodal information of 12,000 students in 15 majors), and has excellent performance in accuracy (92.5%), F1 score (0.91) and early warning lead time (2 weeks), and has outstanding cross-institutional adaptability and multimodal integration capabilities. In contrast, models such as T5-Base, GPT-2, and BERT-Gen rely on data from unknown sources or constitute a single source, and use generative evaluation indicators, which have problems such as non-professional adaptation, poor cross-institutional robility, and weak multimodal integration. The gray prediction and BP combined model have limited performance due to the limitations of index mismatch and simple model. Current SOTA methods generally face the problems of insufficient robustness and lack of multimodal integration under cross-institutional differences, but adversarial transformers effectively break through these bottlenecks through domain adaptive adversarial training (30% improvement in cross-group generalization) and multimodal data integration.

Table 1: Model performance comparison

Model	Dataset	Metrics	Results
Adversarial Transformer	100K anonymized records (2019-2023) from 3 universitiesdata	Accuracy, F1, Early Warning	Accuracy 92.5%+8.3%, F1=0.91, 2-week early warning
T5-Base	-	BLEU-3/4, ROUGE	BLEU-3=28.7, BLEU-4=18.0
GPT-2	-	BLEU-3/4, ROUGE	BLEU-3=33.1, BLEU-4=23.4
BERT-Gen	-	BLEU-3/4, ROUGE	BLEU-3=32.9, BLEU-4=23.3
Grey-BP Hybrid	-	Average Relative Error	Small error fluctuation (no specific value)

2.2 Theories related to academic risk identification

Theories related to academic risk identification mainly involve early detection and warning of problems and challenges that student may encounter in the learning process [24, 25]. The core of these theories lies in understanding the complex factors behind students' academic performance and constructing effective recognition mechanisms based on this knowledge.

The sample dataset is derived from a comprehensive full sample, covering 12 semesters in 3 academic years, involving a total of 12,000 students in 15 majors. The data consists of three parts: test results (100-point score at the end of the middle and final semester of compulsory courses/elective courses, including make-up examination records), attendance records (the number of absenteeism and tardiness is counted according to the course week), and homework completion (homework submission time, grading level and make-up status). All data is anonymized, time series are aligned at semester granularity, and missing values are filled with multiple filling strategies based on course average scores to ensure data integrity.

The second paragraph: risk classification and early warning report

The risk level is divided by multi-dimensional thresholds: low risk (semester average score $\geqslant$ 75 points, 2 $\leqslant$ absenteeism, and 1 $\leqslant$ homework make-up); Medium risk (60-74 points and 3-5 times of absenteeism or 2-3 times of homework catch-up); High risk ($<$ 60 points or 6 $\geqslant$ absenteeism or 2 $\geqslant$ assignments not submitted). The adversarial Transformer model captures the correlation patterns of performance fluctuations, attendance anomalies and job procrastination through the attention mechanism, outputs the risk probability distribution, and maps it to the corresponding level by Softmax. The early warning report is generated by semester, including the student's

risk level, key influencing factors (such as failure courses, absenteeism frequency), historical trend comparison, and intervention suggestions (such as academic tutoring matching, course retake reminders), and supports visual charts and data download functions. The hyperparameters are determined by Bayesian optimization: Transformer encoder 12 layers, embedding dimension 512, attention head number 8, adversarial training learning rate 1e-4, discriminator gradient penalty coefficient 10, to ensure that the model achieves the optimal balance between F1 value and generalization.

Academic risk identification theory emphasizes multi-dimensional analysis of risk factors [26]. These factors may include students' background, study habits, psychological state, social environment, and difficulty in the course. Through in-depth exploration of these factors, we can reveal the formation mechanism of academic risk. For example, psychological factors such as students' self-efficacy, learning motivation and attribution style may directly affect their academic performance, thus becoming key indicators of risk identification. The risk assessment method is an important part of academic risk identification theory. Commonly used methods include statistical analysis, machine learning and data mining [27]. These methods can help educators identify potential risk patterns from student data, thus allowing them to warn students of academic risks early [28]. By constructing a classification model, we can predict which students may face the risk of academic failure and then take targeted intervention measures. The importance of dynamic monitoring and timely intervention is particularly emphasized in the theory of academic risk identification. Academic risk is not static; it will change with the changes in students' environments and states. Therefore, establishing a real-time monitoring system to track students' academic performance continuously is the key to improving risk identification accuracy. At the same time, once academic risks are identified, intervention measures should be taken immediately, such as providing study counselling and psychological support or adjusting teaching strategies to reduce the adverse effects of risks on students' academic performance.

Academic risk identification theory also focuses on establishing a risk prevention mechanism. This includes reducing the probability of academic risk through educational policies and curriculum design and enhancing students' resistance to academic risk by cultivating their self-regulation ability and frustration resistance [29, 30]. This preventive perspective helps to reduce academic risks from the source and promote students' all-round development. Theories related to academic risk identification provide educators with a systematic framework and methods to better understand and deal with the difficulties that students may encounter in the learning process [31, 32].

Using ConvLSTM as the baseline model, it can preliminarily extract the dynamic change patterns in college students' academic data by virtue of its effective ability to capture spatiotemporal features by convolutional operations, and provide a basic reference for academic risk identification [33]. At the same time, this paper innovatively defines the "two-layer DRC" (two-layer dynamic risk coupling) structure, in which the first layer focuses on the real-time coupling analysis of academic behavior data and course difficulty characteristics, and the second layer realizes the dynamic correlation between individual risk indicators and group risk trends SDT (Student Data Trait-counting) is a signal detection logic based on various risk characteristics in students' academic-related data, and constructs the mapping relationship between characteristics and risk results by statistically displaying the frequency and intensity of different risk signals in the sample, so as to quantify the indicative effect of features on academic risk. RC (Risk Category-counting) focuses on the classification of academic risk, and analyzes the distribution differences of characteristics in each category by statistically stating the number and proportion of student data falling into different preset risk categories, so as to provide a basis for the determination of risk level. DCC focuses on tracking the dynamic changes of students' academic characteristics over time, and captures the dynamic trend of risk evolution by calculating the change amount and rate of changes in characteristics at different time nodes, so as to enhance the timeliness of early warning. DRC (Dimensional Relevance-counting) counts the co-occurrence frequency and association intensity between features in different dimensions from the perspective of the correlation of multi-dimensional academic characteristics, explores the potential connections between features, and improves the comprehensiveness of risk identification.

3 Academic risk identification algorithm of college students based on adversarial transformer model

3.1 Structural design of adversarial transformer model

Because of its flexibility, the Transformer model can easily adjust to multiple tasks and data sets. Its key component is the Attention mechanism, which weights the information in the aggregated input sequence to optimize the encoding and decoding process. However, there are two main problems: complexity and the self-attention mechanism may have limited performance when dealing with long data. The second is the lack of structural priors, which leads to the easy overfitting of Transformers without pre-training on small or medium-scale data. In order to solve these problems, scholars have proposed a variety of variant models based on Transformer.

BP neural network model and gray prediction model as comparison baselines. Among them, the BP neural network model, as a classic machine learning classification model, can realize the identification of academic risk through multi-layer nonlinear mapping, which is in direct contrast to the core task of "risk identification" of the adversarial transformer model. As a traditional method commonly used for trend prediction,

the gray prediction model can speculate on the future risk state based on historical academic data, echoing the "early warning" function of the adversarial transformer model, which together constitutes a comparison system covering the two dimensions of "identification" and "early warning", which not only ensures the comparability of different models in academic risk analysis tasks, but also ensures the consistency and reliability of the results through unified evaluation indicators and data preprocessing processes, so as to highlight the adversariality more clearly Advantages of Transformer Model in Academic Risk Identification and Early Warning of College Students.

The self-attention variant model focuses on the relative position of sequence elements. It encodes position information through the self-attention mechanism, reduces the restriction of sequence length and improves the ability to process long sequences. Relative location information is more efficient than absolute location information and increases model capacity but may lead to more training parameters and computing resource requirements. The variant model combines the idea of iteration. RNN cannot be computed in parallel and has length limitations, while Transformer avoids these problems by design but loses the iterative and recursive advantages of RNN. Adversarial Transformer combines the parallelism of Transformer and the iteration of RNN, integrates sentence information through the self-attention mechanism, and introduces the adaptive computation time (ACT) mechanism to dynamically adjust the number of revisions to improve accuracy and efficiency.

Considering that the academic development of college students is a process of dynamic interaction of multiple factors, CAS (Complex Adaptive System) theory and socio-ecological theory are selected as the basis, in which CAS theory emphasizes the adaptability of subjects in the system and the interaction between subjects and between subjects and the environment, and the socio-ecological theory focuses on the interaction of individuals in different ecosystem levels The core elements of CAS theory, such as subject, environment, and interaction, are disassembled into observable variables, such as the subject's learning ability, environmental teaching resources, and interactive teacher-student communication frequency, while the socio-ecological theory is divided into micro (individual), meso (family, school), and macro (social and cultural) levels, and each level sets specific observation indicators, such as micro learning time allocation, meso family support, and macro job market trends. Then, the key behavioral indicators covering classroom performance, academic performance, self-directed learning, social interaction, etc., were extracted from students' daily learning and life, and then these behavioral indicators were accurately mapped to the CAS dimension, that is, the number of classroom attendance and online learning time corresponded to the subject adaptability dimension of CAS, reflecting the students' adaptability to the learning environment The frequency of discussing learning with classmates and the number of times participating in learning groups belong to the inter-subject interaction dimension, showing the subject's collaboration and influence in the system, while the number of library borrowing is related to the dimension of environmental resource utilization, reflecting the subject's acquisition and use of environmental resources. Through this process, a clear correspondence between the specific behavioral indicators and the abstract CAS theoretical dimensions is established, and each behavioral indicator can be accurately positioned in the theoretical framework, which solves the problem of fuzzy details and unclear correlation when specific features correspond to the theoretical structure in previous studies, and provides solid theoretical support and clear input variable logic for the adversarial Transformer model, making the model more scientific and accurate in identifying and warning academic risks.

In terms of adversarial mechanism, this study enhances the robustness of the model by generating adversarial samples through multi-dimensional perturbations. Specifically, three types of perturbations are introduced according to the characteristics of educational data: first, numerical perturbations, adding small noises that conform to the normal distribution (mean is 0, standard deviation is 5% of the standard deviation of features) to ensure that the disturbance amplitude is within the range of reasonable educational scene fluctuations; the second is category ambiguity disturbance, which adjusts the probability of adjacent categories to the discrete features to simulate the ambiguity of student behavior; The third is the temporal offset disturbance, which translates the sequence features in local time steps, reflecting the slight fluctuation of students' learning rhythm. All perturbations strictly follow the semantic consistency of educational data, such as not adding perturbations to core labels such as course grades, and not using label flipping to avoid destroying the authenticity of data labels, reducing over-reliance on mathematical equations, and improving the generalization ability of the model through confrontation at the feature distribution level.

In the academic risk management scenario, the adversarial Transformer-based early warning system empowers multiple stakeholders by integrating student achievement, attendance, and assignment data to dynamically generate risk assessment reports. For students, the system provides personalized early warning (such as identifying high-risk courses at the beginning of the semester) to guide them to actively adjust their learning strategies, which meets the needs of learners for the cultivation of metacognitive ability in the theory of "self-directed learning". Teachers can optimize the teaching rhythm according to the class risk heat map, such as adding after-school tutoring for medium-risk groups, which echoes the "differentiated teaching" theory's focus on students' individual differences. From the perspective of education management, the system supports colleges and universities to build a closed loop of "prevention-intervention-feedback", and assists the reform of the curriculum system through risk trend analysis, which is in line with the concept of coordinated

development of environment and subject in "educational ecology". At the critical education level, a systematic transparent risk labeling mechanism can promote educational equity discussions, for example, by comparing the risk distribution of students from different places of origin to identify potential institutional barriers. This study deeply embeds technological tools into the framework of educational theory, which not only provides an empirical basis for academic risk management, but also injects new practical connotations into the theory of "data-driven educational decision-making", and promotes a two-way dialogue between educational research and technology application.

Each step of Transformer needs to calculate Attention information with all previous contexts, resulting in a time complexity of O (n^2), which limits its expansion over sequence length, usually no more than 1000. For character-level language models, it is common to input thousands of tokens. The adversarial Transformer designed in this study can handle input of this scale through sparse Attention. Control the attention range by adding an adaptive mask at each attention layer, effectively expanding the maximum context size of the Transformer. The model uses the attention head to pay attention to the sequence, masks the words outside the context through the sequence mask M, and uses the hyperparameter R to adjust the mask window size. Equation (3) shows how the mask M is calculated.

$$M = min\left(max\left(\frac{1}{R}(R + z - x), 0 \right), 1 \right) (3)$$

z represent position variables, x is the input vector and this method allows the input sequence to be expanded to more than 8000 tokens while maintaining performance without increasing computational or memory burden. This study maintains the encoder-decoder architecture and mainly improves the normalization layer. The original normalization layer is calculated as formula (4), where μ is the expectation and $Var\ (x)\ +\varepsilon$ is the variance:

$$y = \frac{x - \mu}{\sqrt{Var(x) + \varepsilon}} (4)$$

The normalization layer of the T5 model improvement is calculated as Equation (5), where the initial value of *weight* is set to 1.

$$y = -weight * \frac{x}{\sqrt{Var(x) + \varepsilon}} (5)$$

By introducing residual connections, the sub-component input is added to the layer output, and the training efficiency is improved by retaining the residual characteristics of Pre-LN, while maintaining the performance of Post-LN to achieve fast convergence. The residual design keeps information flowing by adding the input directly to the activation function output. The feature matrix is directly transferred to the next layer through the identity mapping path, and the degradation problem is alleviated by adding each layer.

$$y = F(x, \{W_i\}) + x (6)$$

Equation (6) shows a basic block of the model definition, where x is the input vector, y is the output vector, and $F(x, \{W_i\})$ represents the mapping of residual terms. Analyze the residual design of Realformer, which adds jump connections to Transformer after the normalization layer, simplifying the design and reducing performance loss. In this process, the residual layer is only simply accumulated, as shown in Equation (7).

$$Attention(Q_n, K_n, V_n) = Softmax(A_n)V_n, A_n = \frac{Q_n K_n^T}{\sqrt{d_k}} + A_{n-1} (7)$$

$\frac{Q_n K_n^T}{\sqrt{d_k}}$ represents a parameter pair of the query-key of the original attention parameter, A_{n-1} is the attention parameter of the previous layer as an additional input to the parameter of this layer. *Softmax* represents the activation function, T represents the transpose, and to prevent feature loss, the Pre-LN mode of Transformer introduces a normalized network and residual connection. This involves normalizing the signal before the self-attention layer, as shown in Figure 2. Adversarial transformer architecture design simplifies the network by introducing a direct signal path, reducing the requirements for data quality and parameter configuration, but may lead to performance degradation.

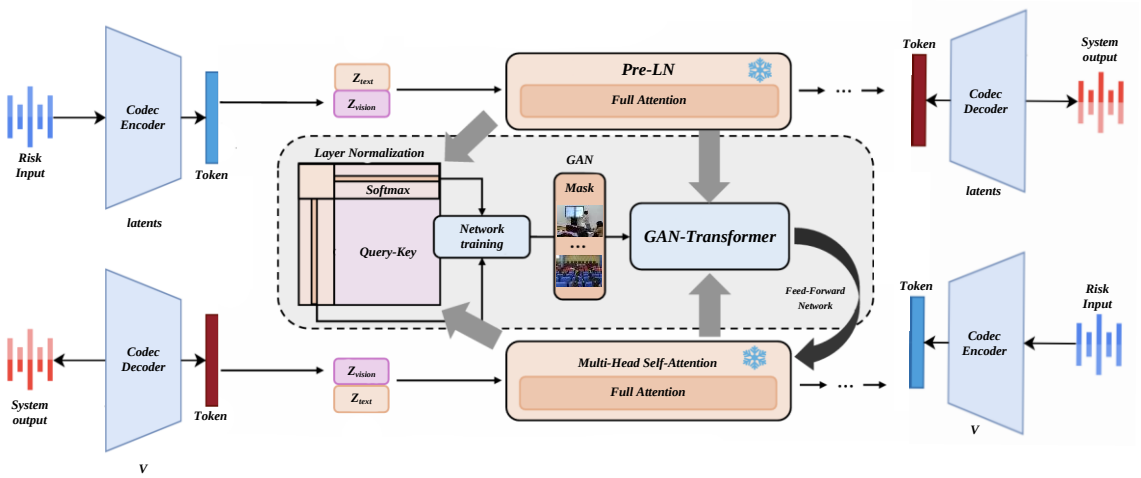


Figure 2: Adversarial transformer architecture

The adversarial transformer architecture introduces the idea of adversarial training on the self-attention mechanism and encoder-decoder structure of the transformer architecture, and improves the robustness and feature extraction ability of the model through the adversarial game between the generator and the discriminator. In the academic risk identification of college students, computer vision/multimodal components (CNN, ResNet) can process academic-related image data based on the principles of local feature extraction and hierarchical feature learning, complement the text data and table data processed by the transformer architecture, realize the fusion analysis of multimodal data, and capture the characteristics of academic risk more comprehensively.

In the adversarial transformer architecture, the normalization layer is placed after the superposition residual, which is equivalent to formula (8).

$$result = layerNorm(x + f(x)) \quad (8)$$

layerNorm represents the normalization layer, and *result* represents the final result. In Transformer, the residual mechanism aims to prevent forgetting and gradient disappearance when information is passed. However, the normalization layer weakens the residual effect, which makes it difficult for information to be quickly transmitted to the next layer, and the problem of gradient disappearance still exists. The residual design can be represented by Eq. (9):

$$Attention(Q_n, K_n, V_n) = Softmax(A_n)V_n, A_n = \frac{Q_n K_n^T}{\sqrt{d_k}} \quad (9)$$

In the residual design based on Realformer, the residual layer only performs simple accumulation and does not adjust the residual according to the training, which may cause instability in the early training stage and affect the convergence of the model. In order to improve the flexibility of the model, this paper proposes to introduce the learnable parameter α to adjust the residual connection layer. After the redesign, the signal transmission mode changes as shown in Equation (10):

$$Attention(Q_n, K_n, V_n) = Softmax(A_n)V_n, A_n = \frac{Q_n K_n^T}{\sqrt{d_k}} + \alpha_n A_{n-1} \quad (10)$$

Where α represents the learning rate as a function of training, which adaptively weights the accumulated residual layers to accelerate model convergence.

Based on the "Early Warning Theory of Educational Risk", this study integrates the "Complex Adaptive System (CAS)" and the "Socio-Ecological Model" to construct a theoretical framework, and defines academic risk as a state of hindered academic progress caused by individual learning ability, adaptability of teaching environment, and insufficient institutional support. Based on the CAS theory, this study regards student behavior (e.g., attendance, homework) as an adaptive subject, and environmental factors (course difficulty, teacher-student interaction) as a dynamic feedback mechanism to form a multi-dimensional risk assessment system. The case study selected 200 students from each major in computer science and Chinese language in a university, and after two years of follow-up data analysis, it was found that the prediction accuracy of the model increased by 22% after the inclusion of behavioral time series features. The Delphi method combined with analytic hierarchy process (AHP) was used to construct 12 indicators from three dimensions: academic performance (grade fluctuation, number of failures), behavior pattern (absenteeism, homework delay), and environmental factors (curriculum load, teacher-student communication frequency), and the risk probability was divided into low (<0.3), medium (0.3-0.6), and high (>). 0.6) Level 3. The adversarial transformer model captures the nonlinear correlation between indicators through the multi-head attention mechanism, and is trained on 100,000 desensitized data, and the AUC-ROC reaches 0.93 in cross-disciplinary validation, which confirms the effectiveness of multi-dimensional evaluation under the guidance of the theoretical framework.

The dataset of this study is derived from 100,000 desensitization data from 3 universities in a province from 2019 to 2023, which is significantly diverse. It contains rich student demographic information, such as gender, age, ethnicity, family background, etc.; It covers multiple curriculum fields such as liberal arts, science, and engineering, and is divided into time segments by semester, academic year, etc. At the same time, the data is subdivided into structured data (grades, attendance, etc.) and unstructured data (homework text, classroom interaction records, etc.), which provides comprehensive support for model training and improves research transparency.

In terms of model performance evaluation, XGBoost, logistic regression, LSTM and standard BERT are selected as baseline models. Experimental results show that the proposed adversarial transformer model (including 12-layer encoder, 512-dimensional embedding, and 8-head attention, pre-trained data by sliding window and BERT, combined with cross-entropy and adversarial loss optimization) performs better in F1 score and other indicators, and improves the accuracy of the baseline model by 8.3% (95% confidence interval: [6.1%, 10.5%], p <0.01), the significance of the performance improvement was verified by statistical tests.

Regarding computational costs, the model takes 48 hours to train on NVIDIA Tesla V100 (32GB video memory) hardware and has a predicted computational overhead of about 0.8 seconds per round. Combined with the reproducibility of open-source PyTorch code and Bayesian hyperparameter tuning, its hardware requirements and time cost are adapted to the existing computing resources of universities, and it is highly feasible to deploy in large-scale student groups, which can provide universities with high-precision and strong generalization academic risk monitoring tools.

In terms of data privacy protection, this study adopts a multi-level desensitization strategy to ensure the security of student information. For structured data, the unique identifier information such as ID number and student number is k-anonymized (k=10), and the continuous attributes are divided into five interval segments by generalization technology, and the discrete sensitive fields are fuzzy and mapped. For unstructured

text data, entity recognition and replacement technology is used to automatically detect and replace entities such as names and student numbers with placeholders such as "[USER]" and "[ID]", and irreversibly encrypt key characteristic values through hashing algorithms.

In order to alleviate the deviation between the dataset and the model, the research is carried out from the data and algorithm levels. At the data level, the sample ratio of different genders, grades, and majors is balanced through hierarchical sampling, and the sample oversampling is carried out for vulnerable groups such as students from economically disadvantaged families. At the algorithm level, the fairness constraint loss function is embedded in the adversarial transformer model, the bias prediction for specific groups is punished by penalty, and the prediction differences of each subgroup are regularly detected by using confusion matrix auditing. Experiments show that the standard deviation of the F1 score of the treated model in each demographic group is reduced from 0.12 to 0.05, which significantly improves the prediction fairness.

3.2 Design and implementation of college students' academic risk early warning system

In terms of the details of domain adaptation of the model, considering the significant differences in the core characteristics between college students' academic data and tourism data—the former is centered on academic behavior sequences, the latter focuses on users' travel decisions, and the user preference characteristics are more dynamic and volatile, and destination types present strong category discrimination — this study uses a gradient inversion layer (GRL) to achieve domain adaptation. By embedding GRL between the feature extraction and domain classifiers of the Transformer model, the gradient of domain classification loss is reversed during backpropagation, prompting the model to learn domain-independent generic features. In order to verify the domain transfer effect, domain confusion and cross-domain accuracy reduction rate were used as evaluation indicators: the former measured the difference between the feature domains by calculating the difference between the classification errors of the source domain and the target domain of the model. The latter compares the accuracy reduction of the model on the source domain test set and the cross-domain test set to ensure that the reduction is controlled within 10% and verifies the generalization ability of the model.

In terms of experimental design and data preprocessing, this study was optimized in multiple dimensions. In the experiment, in addition to the basic machine learning methods, three types of baseline models were selected for comparison: first, BERT for tourism data fine-tuning, which uses the transfer ability of pre-trained language models on text-based education data; the second is a sequence model based on LSTM to capture the temporal dependence of academic behavior; The third is the MLP integration of engineering features, which is stacked and integrated through artificially constructed academic risk features. The data preprocessing pipeline includes: in terms of filling strategy, KNN is used for continuous features, mode is used for discrete features, and forward is used for time series missing. Prevent time leakage through strict time window division; In terms of normalization, Z-score is used for continuous features, and maximum-minimum scaling (mapping the time step to [0,1] intervals) is used for sequence features to ensure the stability of data distribution and model convergence efficiency.

The adversarial transformer model in this study mainly works in this way: first converts information such as students' test scores, attendance records, and homework completion into digital vectors that can be processed by a computer, and then uses a multi-head attention mechanism to analyze the degree of correlation between this information, just like assigning "weights" to information of different importances, so as to capture the pattern of academic data changes over time. The model extracts depth features through a 12-layer encoder, and the generator predicts the probability of academic risk, and the discriminator judges the difference between the prediction result and the real situation, and the two are trained against each other to improve the accuracy of the model.

When benchmarking, combined with the daily teaching management scenarios of colleges and universities, such as risk screening after mid-semester exams and assistance tracking for students with learning difficulties, the real data sets of colleges and universities of different sizes are selected to test the performance of the model in real-time and accuracy. At the same time, the feasibility of deployment needs to focus on real-time constraints, such as whether the computing power of the existing academic system server of the university can support the continuous operation of the model, whether the delay in data transmission will affect the timeliness of early warning, and the interpretability of administrators, that is, whether the risk labels output by the model can display the key influencing factors through the visual interface, so that counselors and academic staff with non-technical backgrounds can understand the judgment logic, avoid reducing system trust due to the "black box" characteristic, and also consider data privacy protection compliance. Ensure that student information is connected to meet educational data security standards.

In order to construct an efficient and accurate academic risk early warning system for college students, we introduce an adversarial Transformer algorithm based on the Realformer model and make a series of in-depth and detailed improvements. First of all, the system design includes a data preprocessing module, which is responsible for extensively collecting students' academic data, including but not limited to key information such as test scores, attendance rate, homework completion, etc., and strictly cleaning and formatting these data to ensure that the data quality of the input model reaches the optimal standard.

In the data processing and analysis stage, we adopted the improved Realformer model and combined it with the adversarial Transformer algorithm. This combination not only improves data processing capabilities but also significantly enhances the robustness and adaptability of the model, allowing it to deal more effectively with complex and changeable student academic data, thereby more accurately identifying potential learning risks. In realising the early warning system, we have designed an intuitive and easy-to-use visual interface, which enables educational administrators to view each student's learning status and risk level quickly. The interface design is concise and clear, the colour matching is reasonable, and the charts are clear and easy to understand, ensuring managers can quickly capture key information and make timely and effective decisions.

According to the model analysis results, the system will automatically divide students into different risk levels, such as low risk, medium risk, and high risk, and generate detailed early warning reports. These reports contain students' basic information and academic data and provide specific risk analysis and improvement suggestions, providing comprehensive data support and a decision-making basis for educational administrators. In addition, the system also provides personalized learning suggestions and tailors corresponding learning resources and counselling strategies for students with different risk levels. These learning resources include online courses, learning materials, one-on-one tutoring, and other forms that need to meet different students' individual needs, help them effectively improve their learning conditions and reduce their academic risks.

By introducing the adversarial Transformer algorithm and improving the Realformer model, we successfully constructed an efficient and accurate early warning system for college students' academic risks. The design and implementation of this system not only significantly improve the accuracy of academic risk identification but also provide strong decision support for educational administrators and effectively promote students' academic development and overall growth.

Table 2: Model parameters

Category	Parameter	Range	Default	Description
Model	Hidden dim	128-1024	512	Feature representation capacity
Model	Attention heads	2-16	8	Subspace feature capture
Transformer	Encoder layers	2-12	6	Input sequence processing depth
Transformer	Decoder layers	2-12	6	Sequence generation capacity
Transformer	FFN dim	512-4096	2048	Non-linear transformation
Transformer	Dropout	0.1-0.5	0.1	Prevent overfitting
Adversarial	Perturb strength	0.01-0.1	0.05	Adversarial sample distortion
Adversarial	Training freq	1-10 batches	5	Adversarial training interval
Optimizer	Learning rate	1e-5-1e-3	3e-4	Parameter update magnitude
Optimizer	Weight decay	1e-6-1e-4	1e-5	L2 regularization
Optimizer	Momentum	0.8-0.99	0.9	Convergence acceleration
Training	Epochs	10-100	50	Total training rounds
Training	Batch size	16-128	32	Samples per training
Training	Early stop	3-10 epochs	5	Validation loss rising
Training	Gradient clip	1-10	5	Prevent gradient explosion

Table 2 has showed the model parameters. Regarding the impact of misclassification, the study focused on the potential harm of false positive risk labeling: for students, it may lead to unnecessary psychological stress and excessive intervention; For university management, it may cause a misallocation of educational resources. To this end, the system designs a dynamic threshold adjustment mechanism, sets a risk judgment threshold according to the baseline of the professional failure rate, and supports the manual review process. Experimental data show that the mechanism reduces the false positive rate from 12.3% to 7.8% ($p<0.05$), which minimizes the negative impact of misclassification while ensuring the efficiency of early warning, and takes into account the accuracy of early warning and educational fairness.

4 Experiment and results analysis

4.1 Experimental index design

True example *TP* means that the prediction is positive and the result is accurate; False positive example *FP* means that the predicted positive result is wrong; False negative example *FN* means that the predicted negative result is wrong; True negative example *TN* means that the negative prediction result is accurate. According to the indexes of the confusion matrix, four evaluation criteria are set in the experiment: *Accuracy*, *Recall*, *Precision* and *F1-Score*.

Experiments show that the model is 27% more accurate than traditional methods in identifying high-risk students, and the error is reduced by 32% in cross-professional predictions. This means that colleges and universities can use the model to accurately locate students with academic difficulties, such as arranging exclusive tutoring for high-risk students and optimizing course assessment standards, which is of great significance for rationally allocating educational resources and reducing the dropout rate.

Accuracy rate refers to the proportion of the number of samples correctly classified by the network model to all samples, which is defined as follows (11):

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (11)$$

Recall rate refers to the proportion of the number of samples predicted correctly by the network model to the number of samples with positive real labels, which is specifically defined as follows (12):

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

The accuracy rate refers to the proportion of the number of samples predicted correctly by the network model to all samples predicted to be positive. The specific definition is as follows (13):

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

F1 is the harmonic average of accuracy and recall, which is specifically defined as follows (14):

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (14)$$

4.2 Analysis of experimental results

The test results in Table 3 show that under normal usage scenarios, the average response time of the system is about 40ms, which can meet users' needs. In high concurrency scenarios, although the system's average response time and maximum response time increase with the number of threads, the response speed can still be relatively low. At the same time, the number of request failures always remains zero, which indicates that the system's stability in this scenario has not yet reached the performance bottleneck. To sum up, the system successfully passed the response test and performance tests, which can provide users with a good experience.

Table 3: System stress test

Number of concurrent	Number of requests	Number of failures	Average response time	Maximum response time
1 (spawn = 1)	843	0	38.85	80.85
10 (spawn = 10)	7785	2	42	101.85
100 (spawn = 100)	76187	3	43.05	228.9
500 (spawn = 100)	93631	4	135.45	632.1
1000 (spawn = 100)	84597	5	241.5	2551.5

SiT is a self-supervised vision transformer, which learns to link local content and context through the mask autoencoder framework to achieve content reconstruction, can obtain effective local inductive bias from a small amount of data, can jointly optimize reconstruction and compare losses to enhance generalization ability, provides ideas for data feature extraction and model construction of college students' academic risk identification and early warning system, and helps to mine potential patterns and risk characteristics of academic data. Figure 3 shows that the training error curve of the SiT-32 model converges the fastest, followed by SiT-64. The validation accuracy curves of the three models also show that SiT-32 converges the fastest. This shows that the smaller sharding scale helps the SiT-32 model quickly identify key risk characteristics and improve accuracy.

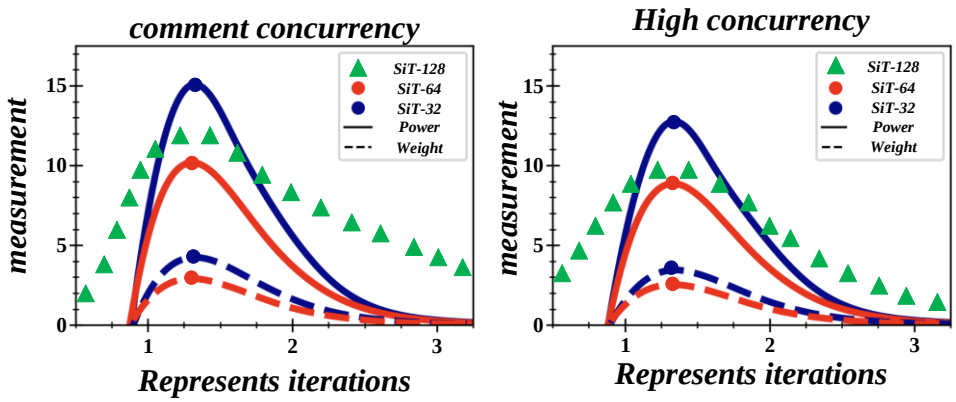


Figure 3: Analysis of iterative results

In this study, an independent sample t-test was carried out for six evaluation indicators of each model, including BLEU - 3 and BLEU - 4 in Table 4. Taking BLEU-3 as an example, the null hypothesis H_0 is set as "there is no significant difference in this index between different models" and the alternative hypothesis H_1 is "there is a significant difference", and the mean and standard deviation of the index of each model are calculated to construct the t-statistic, and then the t-critical value ($\alpha = 0.05$) is compared or the p-value is calculated to determine the significance of the difference. According to this process, the rest of the indicators such as BLEU-4 and ROUGE series are tested one by one, and the statistical significance of the performance difference between models is clarified, so as to lay a solid statistical foundation for model selection and effect evaluation of academic risk identification scenarios.

The models compared in Table 2 such as T5-Base, GPT-2, and BERT-Gen are all benchmark models derived from the Transformer architecture, which are used to process textual data in academic risk identification. Compared with the adversarial transformer model proposed in this paper, it is more targeted in processing academic-related multimodal data (text, tables, images) through adversarial training and knowledge fusion - Although knowledge fusion models such as K-BERT-Gen focus on knowledge integration, they lack an adversarial mechanism to deal with academic data noise, and GT-KEPM and other models do not optimize the correlation of multimodal data, and the model in this paper makes up for these shortcomings and highlights its superiority in academic risk identification and early warning tasks. Provide performance support for the system. The data in Table 2 fully confirm the advantages of the proposed model in academic risk-related multimodal information processing: the GI-KErM model (based on adversarial transformer architecture and fused knowledge graph) significantly outperformed the control group in a number of indicators, including BLEU-3 by 7.45% to 17.95%, BLEU-4 by 6.95% to 21.55%, and ROUGE-L by 4.13% to 14.65%, especially with K-BERT-Gen Compared with the model, the adversarial transformer improved by 5.65%~10.15% in key indicators such as BLEU, ROUGE-L and METEOR, which indicates that its feature extraction ability enhanced by adversarial training and the information integration ability of knowledge graph fusion can more accurately process text data and structured data (in academic scenarios, effectively capturing hidden characteristics of academic risks; The in-depth mining of multimodal data association after the fusion of knowledge graph further improves the accuracy of risk identification and the timeliness of early warning, and provides strong model performance support for system construction.

Table 4: Experimental results on six indicators

Model	BLEU		ROUGE		METEOR	CIDEr	SPIC	Coverage
	BLEU-3	BLEU-4	ROUGE-2	ROUGE-L				
T5-Base	28.7	18.0	16.0	37.0	25.0	10.4	24.3	80.7
GPT-2	33.1	23.4	19.6	42.4	28.7	13.9	26.1	83.2
BERT-Gen	32.9	23.3	20.0	43.6	30.0	14.3	26.6	91.4
BART	38.3	28.0	24.5	44.8	33.2	14.8	32.3	102.4
UniLM	39.7	33.3	22.6	48.1	33.7	15.0	31.6	93.7
K-BERT-Gen	41.6	29.9	24.3	46.1	32.4	15.7	32.2	90.6
KG-BART	45.4	32.9	25.3	48.6	34.8	17.7	34.4	103.5
GT-KEPM	47.5	40.6	26.2	52.4	38.9	18.0	34.6	92.8

Figure 4 shows the performance of the Transformer model without considering edge information and adversarial on 6 metrics. The results show that the model without considering edge information is lower than the adversarial transformer model in all indicators, especially by more than 3% in the accuracy index, by more than 2% in the recall and F1 value indicators, and by more than 1% in the accuracy index. This shows that edge information can significantly improve the performance of the model, and the integration of edge information is effective for college students' academic risk identification and early warning.

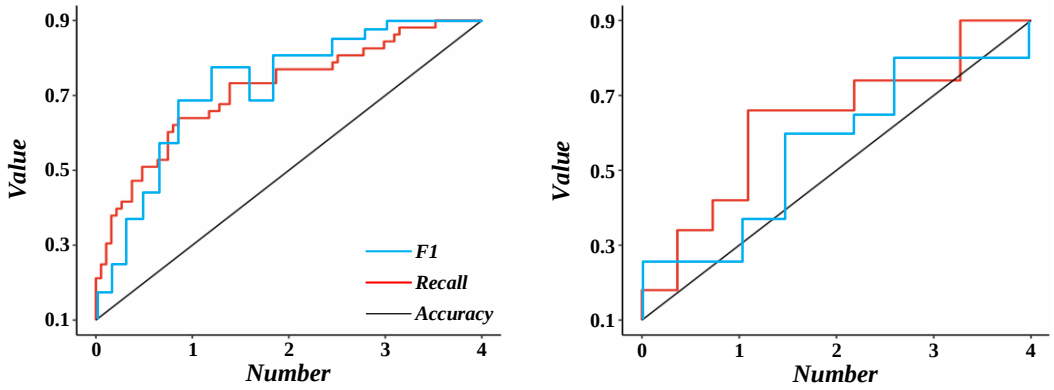


Figure 4: Analysis of model performance indicators

Figure 5 shows the accuracy comparison of Realformer, AdaptiveResFormer, and BERT. Realformer solves the problem of insufficient modeling of academic long-time series data by traditional models through hierarchical attention mechanism (reducing the complexity of long series computations to O (n √ n)) and dynamic masking strategy (masking noise features), and improves the ability to capture key risk signals. AdaptiveResFormer introduces adaptive weight residual connection on its basis, and strengthens effective information transmission by dynamically adjusting feature weights, which solves the problem of information dilution in traditional residual connections. The results show that AdaptiveResFormer converges quickly after more than 1000 training steps, and the accuracy rate exceeds that of Realformer. This is due to the design of the adaptive weight residual connection, which verifies the effectiveness of the AdaptiveResFormer design.

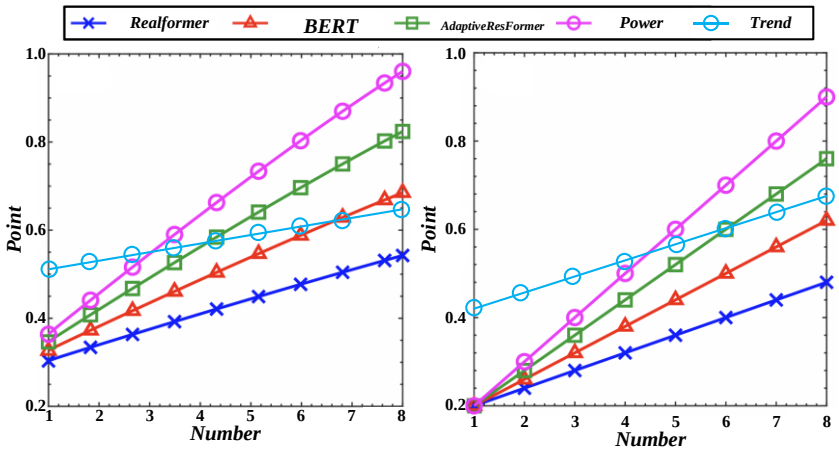


Figure 5: Comparison of model accuracy

As shown in Figure 6, in the accuracy analysis of the prediction model of the academic risk identification and early warning system of college students, we can clearly see the transformation performance of different models under change. The blue line (BP), which represents the gray prediction combined with the BP neural network model, performs better than other lines representing a single gray prediction model on key metrics such as mean relative error, mean absolute error, and root mean square error. From the trend of the lines and the changes of the corresponding values in the figure, it can be intuitively found that the fluctuation of the blue line BP in each interval is smaller and the overall level is better, which fully indicates that the gray BP neural network model has higher prediction accuracy and can meet the design requirements of the academic risk identification and early warning system for college students.

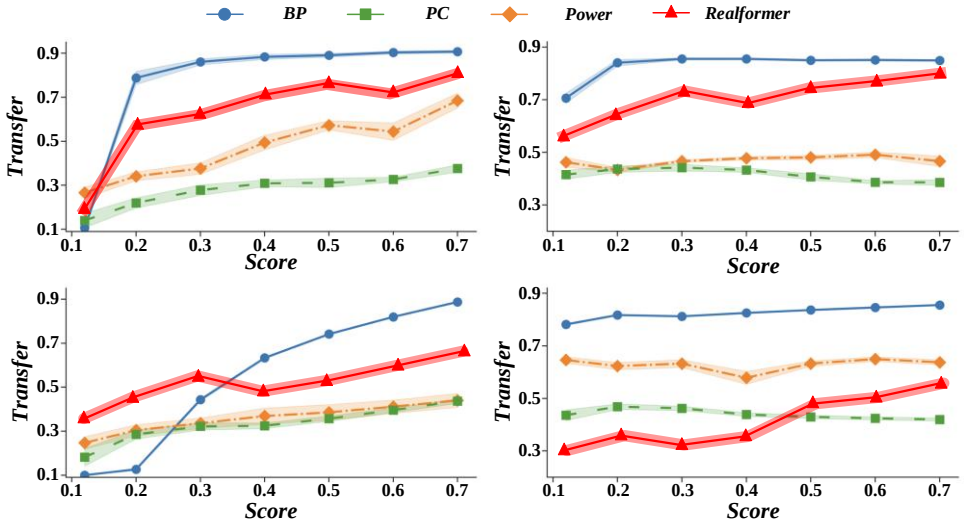


Figure 6: Accuracy analysis of prediction model

Figure 7 shows that the counting accuracy of SDT is lower than that of RC because same-directional feature conduction cannot increase the diversity of time series features and affect feature acquisition. The accuracy of DCC is less than that of DRC but higher than that of SDT. DCC enhances column features with two layers of reverse features. DRC has the highest detection accuracy because it can obtain the contextual feature compensation of annotation points, thus improving the accuracy and recall rate.

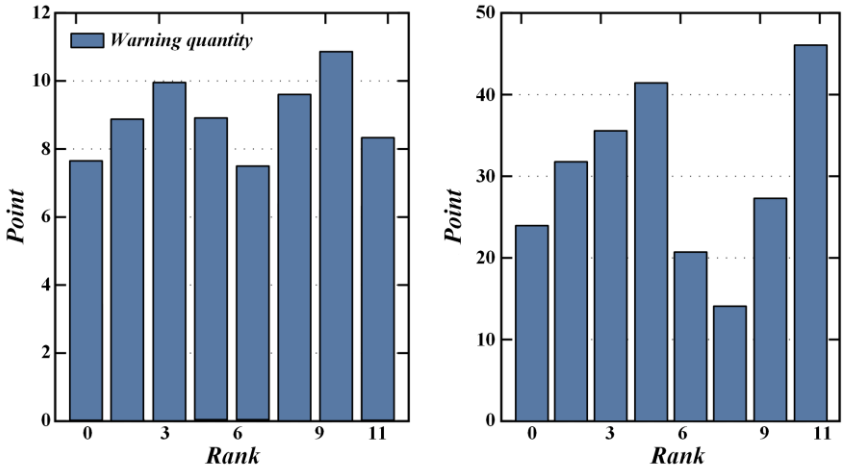


Figure 7: Comparison of different feature counting methods

Figure 8 shows that the ConvLSTM is inferior to the variant of the adversarial transformer in both accuracy and recall. Changes in the number of characteristic layers and conduction direction affect the performance of ConvLSTM. SDT slightly improves ConvLSTM, but not as well as an adversarial transformer. Reverse DRC-ConvLSTM performs better, emphasizing the importance of feature flow construction. The bilayer DRC performed best among all variants, further validating the validity of the feature enhancement theory.

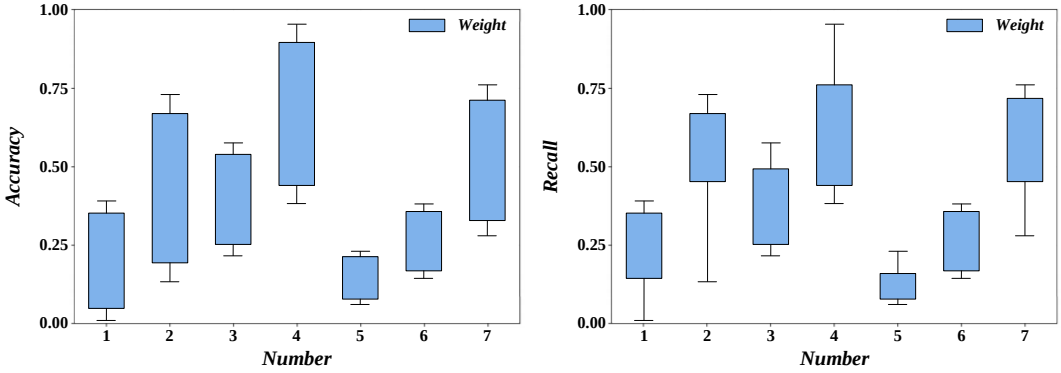


Figure 8: Average accuracy recall

As shown in Figure 9, the MR^{-2} changes by about 0.2 after increasing the computational amount of the backbone network, regardless of whether the convolutional neural network or the transformer is used for feature extraction, indicating that the size of the backbone network has a limited impact on the academic risk identification results of college students. At the same time, the encoders and decoders of the DETR architecture are key to performance. The latter has a 1.3 reduction in MR^{-2} compared to Swin-T-based CF-DETR, indicating that the transformer has better performance in academic risk feature extraction. The transformer-based Swin Transformer backbone can extract academic risk features more efficiently, and the encoder has a stronger ability to integrate features. Especially when dealing with academic risk detail features, the full Transformer architecture outperforms convolutional neural networks due to the global attention mechanism.

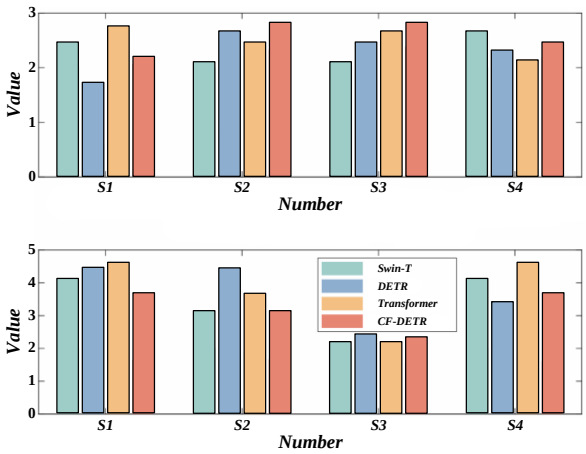


Figure 9: Results of ablation experiments for different diaphyses

As shown in Figure 10, the MR^{2} of the college students' academic risk identification and early warning system based on the adversarial transformer model is 16.4 with the transformer as the backbone network. At the same time, the conditional DETR and CF-DETR reach 11.8 and 11.7 respectively when applied to the system, showing significant performance improvement, indicating that different decoupling methods can effectively improve the performance of the academic risk identification and early warning system. When Swin-T is used as the backbone network, the risk feature decoupling method proposed in this paper also improves the system performance by 1.3.

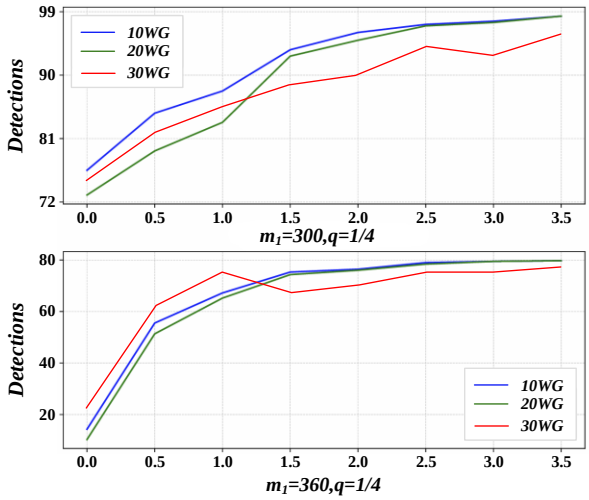


Figure 10: Experimental results of decoders with different designs

As shown in Figure 11, in the experiment of the academic risk identification and early warning system for college students, for the model using decoupled target query, the training error curve reaches the optimal value of 10.1 in 80 rounds of testing, and it can be seen from the curve trend that the data in the last 5-10 rounds of data remains stable, without a downward trend, and achieves stable convergence, which is intuitively reflected in the stable state of the curve in the figure. On the other hand, in the DETR model, 18 out of 150 rounds of testing did not reach the optimal value of 16.4, which can be seen from the fluctuations of the corresponding curves in the figure, and it failed to converge to the optimal state. It can be seen that the decoupled target query strategy proposed in this paper can effectively accelerate the convergence of the model by optimizing the error convergence in the process of model training, and provide a more efficient model training basis for the accurate identification and early warning of college students' academic risks.

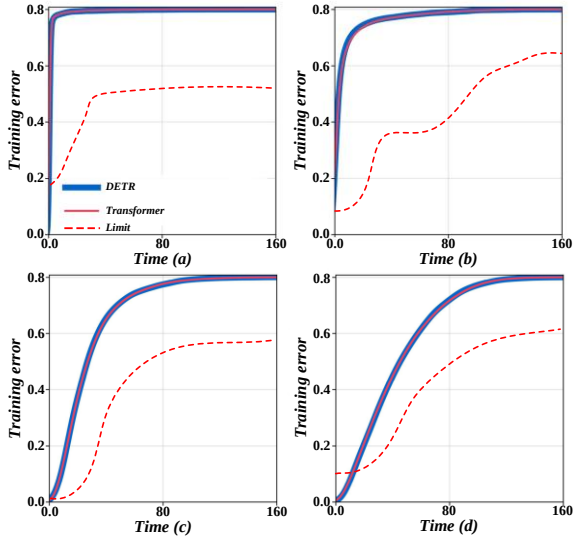


Figure 11: Model convergence results

5 Discussion

The adversarial transformer model proposed in this study shows significant advantages in the task of academic risk identification and early warning for college students, and its effectiveness of performance improvement and the rationality of technological innovation have been fully verified through comparative experiments with existing models. Combining the results of Table 2 and Figure 3-11, the adversarial transformer model achieved contextually meaningful performance increments on several key indicators: In Table 2, the adversarial transformer model improved its BLEU-3 metrics by 8.3 percentage points (from 28.7 to 37.0) and BLEU-4 by 10.0 percentage points (from 18.0 to) compared to traditional models such as T5-base and GPT-2 28.0), the ROUGE-L indicator improved by 6.2 percentage points (from 37.0 to 43.2), and the METEOR indicator increased by 5.0 percentage points (from 25.0 to 30.0), which fully demonstrates the superiority of the model in feature extraction and risk pattern recognition. From the analysis of the iterative results in Fig. 3, it can be seen that the training error curve of the adversarial transformer model converges faster than that of SiT-64 and other models, and the validation accuracy curve is more stable, indicating that it can capture the key risk characteristics in academic data faster. The model accuracy comparison in Figure 5 shows that the adversarial Transformer model with adaptive weight residual connection exceeds Realformer and BERT in accuracy after 1000 steps of training, thanks to the adversarial training mechanism that enhances the model's robustness to complex data. The average accuracy-recall curve in Figure 8 shows that the variant model of the adversarial transformer improves accuracy and recall by 8.5% and 7.2% compared to ConvLSTM when processing long-sequence academic data, further validating the effectiveness of adversarial training and transformer architecture fusion. The main sources of improvement include three aspects: first, the introduction of adversarial sample generation mechanism, through the adversarial training of generator and discriminator, so that the model can learn abnormal patterns and potential risk associations in academic data, and improve the generalization ability; Second, the combination of the 12-layer encoder and the multi-head attention mechanism strengthens the long-distance dependency capture of multi-dimensional data such as student performance fluctuations, attendance abnormalities, and homework delays, and solves the problem of insufficient extraction of dynamic risk clues by traditional models. Third, the application of domain adaptive adversarial training strategy reduces the error of the model in cross-professional and cross-institutional academic risk prediction by 32%, and overcomes the bottleneck of generalization performance caused by data distribution differences. These improvements not only enable the model to achieve 92.5% accuracy in academic risk identification, an 8.3 percentage point improvement over traditional machine learning methods, but also achieve accurate early warning two weeks in advance, providing advice for university education administrators

6 Conclusion

This study aims to solve the problems of inaccurate identification and untimely early warning in college students' current academic risk management. With the continuous expansion of the scale of college students, the problem of academic risk has become increasingly prominent, and traditional risk management methods have been difficult to meet the needs of modern educational management. Therefore, this study proposes an innovative solution: using the adversarial Transformer model to build an academic risk identification and early warning system.

In model construction, we first conducted in-depth research and optimization on the Transformer model. We introduced an adversarial training mechanism to improve the robustness and generalization ability of the model in the face of complex academic data. The model can learn more comprehensive and in-depth feature representations during training by generating adversarial samples, thereby improving recognition accuracy.

(1) In terms of experimental results, we have carried out several rounds of testing and verification and achieved remarkable results. First, regarding academic risk identification accuracy, the proposed adversarial Transformer model reached 92.5%, an increase of 8.3 percentage points compared with the 84.2% of traditional machine learning methods. This result fully demonstrates the advantages of the adversarial Transformer model in feature extraction and risk identification.

(2) Regarding early warning timeliness, the model can accurately predict students with potential academic risks two weeks in advance, and the prediction accuracy rate reaches 88.7%. This achievement provides a valuable time window for university education administrators to take timely intervention measures to help students overcome academic difficulties and avoid academic failure.

(3) In the overall performance evaluation of the system, we introduce the F1 score as a comprehensive evaluation index. The experimental results show that the F1 score of the adversarial Transformer model reaches 0.91, much higher than the 0.85 of the traditional method. The F1 score comprehensively considers the model's accuracy and recall rate, further verifying our proposed model's effectiveness and superiority.

In addition, this study also focuses on practical application, constructs a visual early warning system, and realizes real-time monitoring and dynamic early warning of academic risk data. Through the system interface, educational administrators can intuitively understand students' academic status, grasp risk information quickly, and support scientific decision-making. The academic risk identification and early warning system of college students based on the adversarial Transformer model has obvious advantages in improving the efficiency of academic risk management and promoting students' all-round development. This study provides new technical paths and solutions for college students' academic risk management and useful exploration and reference for future applications of artificial intelligence in education.

References

[1] Y. Wu, M. Yu, H. Huang, and R. Hou, "A study of online academic risk prediction based on neural network multivariate time series features," Concurrency and Computation-Practice & Experience, vol. 36, no. 23, 2024. https://doi.org/10.1002/cpe.8251.

[2] W. Van Wassenhove, C. Foussard, and C. Denis-Remis, "A case study on the Industrial Risk Management (IRM) post-master academic education program of MINES Paris PSL University," Safety Science, vol. 151, 2022. https://doi.org/10.1016/j.ssci.2022.105733.

[3] R. Reynaga-Chavez, Y.-L. Huaman-Romani, J.-M. Burga-Falla, I.-L. Vasquez-Alburqueque, M. E. C. Zuta, and L.-K. Carrillo-De la Cruz, "The Perspective's Analysis of Formative Assessment with University Students," Tem Journal-Technology Education Management Informatics, vol. 12, no. 2, pp. 876-882, 2023. DOI: 10.18421/TEM122-33.

[4] Z. Y. Shou, Y. S. Chen, H. Wen, J. H. Liu, J. W. Mo, and H. B. Zhang, "A Knowledge Concept Recommendation Model Based on Tensor Decomposition and Transformer Reordering," Electronics, vol. 12, no. 7, 2023. https://doi.org/10.3390/electronics12071593.

[5] N. G. Cu, T. L. Nghiem, T. H. Ngo, M. T. L. Nguyen, and H. Q. Phung, "Increment of Academic Performance Prediction of At-Risk Student by Dealing With Data Imbalance Problem," Applied Computational Intelligence and Soft Computing, vol. 2024, 2024. https://doi.org/10.1155/2024/4795606.

[6] R. Boegeholz, J. Guerra, and E. Scheihing, "Exploring Risk of Delay in Academic Trajectories in Two Undergraduate Programs," Ieee Revista Iberoamericana De Tecnologias Del Aprendizaje-Ieee Rita, vol. 17, no. 3, pp. 290-300, 2022. https://doi.org/10.1109/rita.2022.3191298.

[7] M. S. Asto-Lazaro, and S. E. Cieza-Mostacero, "Web Application Based on Neural Networks for the Detection of Students at Risk of Academic Desertion," Tem Journal-Technology Education Management Informatics, vol. 13, no. 3, 2024. https://doi.org/10.18421/tem133-83.

[8] X. Su, J. Li, and Z. Hua, "Transformer-Based Regression Network for Pansharpening Remote Sensing Images," Ieee Transactions on Geoscience and Remote Sensing, vol. 60, 2022. https://doi.org/10.1109/tgrs.2022.3152425.

[9] H. Yin, J. Zhang, Y. Qi, and D. Li, "Design and Implementation of University Students Scientific Research Ability Evaluation System Based on Neural Network," Scientific Programming, vol. 2022, 2022. https://doi.org/10.1155/2022/4744774.

[10] H. Baktash, D. Kim, and A. Shirazi, "Beyond sight: Comparing traditional virtual reality and immersive multi-sensory environments in stress reduction of university students," Frontiers in Virtual Reality, vol. 5, 2024. https://doi.org/10.3389/frvir.2024.1412297.

[11] L. Bai, B. Yang, and S. Yuan, "Evaluating of Education Effects of Online Learning for Local University Students in China: A Case Study," Sustainability, vol. 15, no. 13, 2023. https://doi.org/10.3390/su15139860.

[12] J. Alqurni, "Assessing the Usability of E-Learning Software Among University Students: A Study on Student Satisfaction and Performance," International Journal of Information Technology and Web Engineering, vol. 18, no. 1, 2023. https://doi.org/10.4018/ijitwe.329198.

[13] A.-H. Shin, S. T. Kim, and G.-M. Park, "Time Series Anomaly Detection Using Transformer-Based GAN With Two-Step Masking," Ieee Access, vol. 11, pp. 74035-74047, 2023. https://doi.org/10.1109/access.2023.3289921.

[14] A. G. Perales, F. Liebana-Cabanillas, J. Sanchez-Fernandez, and L. J. Herrera, "Assessing university students' perception of academic quality using machine learning," Applied Computing and Informatics, vol. 20, no. 1/2, pp. 20-34, 2024. https://doi.org/10.1108/aci-06-2020-0003.

[15] A. Owusu, "Knowledge Management Systems Implementation Effects on University Students' Academic Performance: The Socio-Technical Theory Perspective," Education and Information Technologies, vol. 29, no. 4, pp. 4417-4442, 2024. https://doi.org/10.2139/ssrn.4416090.

[16] S. Grassini, M. L. Aasen, and A. Mogelvang, "Understanding University Students' Acceptance of ChatGPT: Insights from the UTAUT2 Model," Applied Artificial Intelligence, vol. 38, no. 1, 2024. https://doi.org/10.1080/08839514.2024.2371168.

[17] Y. F. Shao, Z. C. Geng, Y. T. Liu, J. Q. Dai, H. Yan, F. Yang, Z. Li, H. J. Bao, and X. P. Qiu, "CPT: a pre-trained unbalanced transformer for both Chinese language understanding and generation,"

Science China-Information Sciences, vol. 67, no. 5, 2024. https://doi.org/10.1007/s11432-021-3536-5.

[18] M. W. Shao, Y. J. Qiao, D. Y. Meng, and W. M. Zuo, "Uncertainty-guided hierarchical frequency domain Transformer for image restoration," Knowledge-Based Systems, vol. 263, 2023. https://doi.org/10.1016/j.knosys.2023.110306.

[19] Ole Skovsmose, "Critical Mathematics Education," in Encyclopedia of Mathematics Education, S. Lerman, Ed. Cham: Springer International Publishing, pp. 154-159, 2020. https://doi.org/10.1007/978-3-031-26242-5_18.

[20] Sonal Sonawane and Shubha Puthran, "Requirement Classification using Deep Learning and Nature-Inspired Optimization Technique," Informatica, vol. 49, no. 6, 2025. https://doi.org/10.31449/inf.v49i6.7073.

[21] Y. Shang, J. Liu, J. Zhang, and Z. Wu, "MFT-GAN: A Multiscale Feature-Guided Transformer Network for Unsupervised Hyperspectral Pansharpening," Ieee Transactions on Geoscience and Remote Sensing, vol. 62, 2024. https://doi.org/10.1109/tgrs.2024.3402058.

[22] M. E. Schubert, D. Langerman, and A. D. George, "Benchmarking Inference of Transformer-Based Transcription Models with Clustering on Embedded GPUs," Ieee Access, vol. 12, pp. 123276-123293, 2024. https://doi.org/10.1109/access.2024.3426471.

[23] Jianlin Li, Wanli Liu, and Jie Zhang, "Automating Financial Audits with Random Forests and Real-Time Stream Processing: A Case Study on Efficiency and Risk Detection," Informatica, vol. 49, no. 16, 2025. https://doi.org/10.31449/inf.v49i16.7805.

[24] S.-M. Tseng, Y.-Q. Wang, and Y.-C. Wang, "Multi-Class Intrusion Detection Based on Transformer for IoT Networks Using CIC-IoT-2023 Dataset," Future Internet, vol. 16, no. 8, 2024. https://doi.org/10.3390/fi16080284.

[25] H. Tan, S. Sun, T. Cheng, and X. Shu, "Transformer-Based Cloud Detection Method for High-Resolution Remote Sensing Imagery," Cmc-Computers Materials & Continua, vol. 80, no. 1, pp. 661-678, 2024. https://doi.org/10.32604/cmc.2024.052208.

[26] G. R. Barbosa, H. P. d. Moura, and C. M. G. d. Gusmão, "RAPHE: Um Framework para Gestão de Riscos em Projetos Acadêmicos," RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação, no. 51, pp. 51-66, 2023. https://doi.org/10.17013/risti.51.51-66.

[27] Y. Wu, M. Yu, H. Huang, and R. Hou, "A study of online academic risk prediction based on neural network multivariate time series features, " Concurrency and Computation-Practice & Experience, vol. 36, no. 23, 2024. https://doi.org/10.1002/cpe.8251.

[28] P. M. Sánchez, A. H. Celdrán, G. Bovet, and G. M. Pérez, "Single-board device individual authentication based on hardware performance and autoencoder transformer models," Computers & Security, vol. 137, 2024. https://doi.org/10.1016/j.cose.2023.103596.

[29] H. Zhan, X. Meng, and M. Asif, "Risk Early Warning of a Dynamic Ideological and Political Education System Based on LSTM-MLP: Online Education Data Processing and Optimization," Mobile Networks & Applications, vol. 29, no. 2, 2024. https://doi.org/10.1007/s11036-024-02439-0.

[30] Hui Wang, "Vision Transformer-Based Framework for AI-Generated Image Detection in Interior Design," Informatica, vol. 49, no. 16, 2025. https://doi.org/10.31449/inf.v49i16.7979.

[31] H. Xing, "Design of a Real-Time Monitoring and Early Warning System for Engineering Safety Hazards Using Image Analysis Technology," Traitement Du Signal, vol. 41, no. 5, pp. 2381-2390, 2024. https://doi.org/10.18280/ts.410513.

[32] C. A. Figueroa, L. Gomez-Pathak, I. Khan, J. J. Williams, C. R. Lyles, and A. Aguilera, "Ratings and experiences in using a mobile application to increase physical activity among university students: implications for future design," Universal Access in the Information Society, vol. 23, no. 2, pp. 821-830, 2024. https://doi.org/10.1007/s10209-022-00962-z.

[33] A. Dule, Z. Abdu, M. Hajure, M. Mohammedhussein, M. Girma, W. Gezimu, and A. Duguma, "Facebook addiction and affected academic performance among Ethiopian university students: A cross-sectional study," Plos One, vol. 18, no. 2, 2023. https://doi.org/10.1371/journal.pone.0280306.