Enhancing Machine Learning and Deep Learning Models For Depression Detection: A Focus on SMOTE, RoBERTa, and CNN-LSTM

Chaimae Taoussi[1*], Soufiane Lyaqini[1], Abdelmoutalib Metrane[2] and Imad Hafidi[1]
[1]Laboratory of Process Engineering Computer Science and Mathematics, University Sultan Moulay Slimane, Beni Mellal, Morocco
[2]Faculty of Sciences and Technology, Cadi Ayyad University, Marrakech, Morocco
E-mail: chaimae.taoussi@usms.ac.ma, s.lyaqini@usms.ma, a.metrane@uca.ac.ma, i.hafidi@usms.ma

Student paper

Keywords: Depression detection, machine learning, deep learning, data preprocessing, data augmentation, transformers, mental health AI, SMOTE, XGBoost, RoBERTa, CNN-LSTM

Received: October 28, 2024

Depression is a major public health concern, affecting millions worldwide, and necessitates early, accurate detection for timely intervention. This study focuses on enhancing machine learning (ML) and deep learning (DL) models for improved accuracy in depression detection using the Counsel Chat Dataset. To address the challenges of class imbalance, we employed advanced preprocessing techniques, including the Synthetic Minority Oversampling Technique (SMOTE), alongside model fine-tuning and architectural optimizations. Our results demonstrated significant performance improvements, particularly with transformer-based models and hybrid architectures. RoBERTa, a transformer-based model, achieved an accuracy of 91.55%, an F1-score of 0.91, and a recall of 92.10%, outperforming state-of-the-art approaches. Similarly, CNN-LSTM attained an accuracy of 91.67% with a 95% CI of (0.8987, 0.9312), while XGBoost achieved the highest accuracy among ML models at 93.06%, with a 95% CI of (0.921, 0.941). Statistical tests validated the superiority of these models, with p-values of 5.48e-13 for RoBERTa and 3.41e-16 for XGBoost. These findings underscore the pivotal role of data augmentation and preprocessing in creating balanced datasets and enhancing the predictive capabilities of AI models for depression detection.

Povzetek: Članek izboljša zaznavanje depresije z uporabo SMOTE, RoBERTa in CNN-LSTM, pri čemer optimizira ekstrakcijo značilnosti, povečanje podatkov in natančnost klasifikacije, s čimer dosega najnaprednejše zmogljivosti pri diagnostičnih sistemih umetne inteligence za duševno zdravje.

1 Introduction

Depression is recognized globally as a leading cause of disability, affecting over 300 million people, as reported by the World Health Organization (WHO) [1]. This disorder significantly burdens individuals and public health systems, adversely impacting quality of life, social interactions, and productivity. The complexity of depression arises from a diverse range of symptoms influenced by biological, psychological, and environmental factors, making early diagnosis particularly challenging. Notably, depression ranks as the fourth leading cause of disability worldwide [2-3], with anxiety and depressive disorders affecting nearly one-fifth of the global population. These challenges are compounded by limited access to specialized care, which often results in extended wait times for treatment due to the strain on conventional healthcare systems [4-5].

Recent advances in artificial intelligence (AI), particularly through machine learning (ML) and deep learning (DL) models, offer promising avenues for the early detection and prediction of psychological disorders. These technologies enable the analysis of diverse and voluminous data sources—ranging from textual interactions and voice signals to medical records and online behaviors. Studies have demonstrated the effectiveness of natural language processing (NLP) models in identifying psychological risks, such as suicidal tendencies and depressive disorders, based on online conversations [6]. Moreover, AI models trained on online therapeutic conversations have shown promise in predicting relapse risks among young therapy patients [7] and enhancing suicide prevention interventions by identifying effective strategies [8].

Large language models (LLMs) and reinforcement learning techniques further bolster these capabilities. For instance, models such as GPT-4, when fine-tuned with specific datasets, produce empathetic and contextually appropriate responses, supporting online therapy through improved conversational quality and emotional coherence [9-10]. Beyond text, AI models analyzing vocal cues have shown notable success, as subtle changes in voice can reflect cognitive and emotional shifts linked to depression. Ensemble models capturing these nuances have pro-

vided effective early screening methods that are both non-invasive and accessible [11]. Integrative approaches using clinical and neuroimaging data have also demonstrated potential in predicting treatment responses among depressed patients [12]. For example, recent work has demonstrated how intelligent cognitive assistants (ICAs) can systematically support behavioral changes in mental health contexts, leveraging adaptive techniques to personalize interventions [13].

Despite these advancements, certain challenges remain. Data imbalances, such as the overrepresentation of healthy individuals, affect model accuracy. Techniques like SMOTE have shown success in balancing datasets and enhancing model performance, as seen in studies with elderly populations in South Korea [14]. Furthermore, the opacity of deep learning models—often termed ”black boxes”—poses a barrier to clinical adoption. Explainability tools like SHAP and LIME help make predictions more interpretable, facilitating integration into medical practice [15]. In parallel, computational psychotherapy systems incorporating advanced prediction models and natural language interfaces have demonstrated superior efficacy in addressing stress, anxiety, and depression through personalized user interactions [16].

This study addresses these challenges by proposing a framework that leverages advanced data preprocessing, augmentation techniques, and state-of-the-art ML and DL models for depression detection. Specifically, this study aims to evaluate the impact of SMOTE on mitigating class imbalance in depression detection datasets. Additionally, it explores the fine-tuning of RoBERTa and CNN-LSTM architectures to enhance model performance. Finally, a comparative evaluation of traditional machine learning models (e.g., XGBoost) with deep learning models (e.g., CNN-LSTM, RoBERTa) is conducted to highlight the benefits of advanced architectures combined with data augmentation techniques.

Furthermore, smartphone-based assessments have emerged as powerful tools for real-time monitoring and intervention, with potential to revolutionize mental health care accessibility [17- 18]. These methods, coupled with IoT-enabled devices, facilitate ecological momentary assessments, providing granular, individualized insights that enhance intervention strategies. Furthermore, the use of persuasive technology in promoting equality in mental health care emphasizes the ethical and scalable potential of digital interventions [19].

This article is structured as follows: the Related Work section reviews existing research on depression detection using ML and DL, situating this study within the broader field of mental health research. The Methodology section details the dataset, preprocessing steps, data augmentation techniques such as SMOTE, and models used. Experimental Results and Analysis present the outcomes of various models, comparing performance metrics like accuracy, F1-score, and recall. The Discussion section interprets these findings, contrasting them with existing studies and underscoring implications for AI-driven mental health interventions. Finally, the Conclusion summarizes this work’s contributions, limitations, and potential directions for future research.

2 Related works

The application of machine learning (ML) and deep learning (DL) models in detecting psychological disorders, particularly depression, has advanced significantly. These models now effectively analyze various data modalities—including text, voice, and multimodal datasets, such as the Counsel Chat Dataset—to identify early indicators of psychological disorders through online interactions. Large language models, like GPT-4 and GPT-4-Turbo, have shown notable efficacy in generating empathic therapeutic responses. A recent study highlighted GPT-4-Turbo’s performance, surpassing GPT-4 with a BLEU score of 64% and a ROUGE score of 62%, underscoring its ability to provide lexically rich and nuanced responses crucial for psychological support [10].

Transformers, including models such as RoBERTa and CNN-LSTM, have also excelled in depression detection. For instance, the VPSYC system, designed to deliver real-time therapy, reported an accuracy of 91.2% for emotion classification and 87.5% for depression detection using RoBERTa, with CNN-LSTM achieving 84% and 80.3%, respectively, illustrating the superior capability of transformer models in processing complex emotional states [20]. Analyzing text data from social networks is another promising approach, enabling early detection of depression symptoms. Deep learning models analyzing online interactions can identify symptoms before clinical detection, offering a non-invasive method to screen at-risk individuals [21].

In line with current advancements, a recent study presented the Medico-call platform, a system that combines big data tools like GATE and UMLS for the automatic processing of EMRs to support early prediction of psychological pathologies such as depression. This tool leverages machine learning to enhance diagnostic accuracy and to predict various psychological conditions through real-time data analysis from patient consultations and clinical records [22].

Voice analysis, through acoustic feature examination, represents another promising avenue. Research has shown that subtle pitch alterations can indicate emotional and cognitive changes related to depression, with ensemble models effectively detecting these signals for early screening [11]. Similarly, visual and acoustic signals, including facial expressions and vocal tones, have shown potential in identifying preliminary depression indicators through Deep Convolutional Neural Networks (DCNN) [23].

Challenges persist, however, particularly with dataset imbalances, where healthy individuals often outnumber those displaying depressive symptoms. Techniques like SMOTE (Synthetic Minority Over-sampling Technique)

have been effective in balancing classes, as evidenced by a South Korean study where SMOTE application significantly improved depression detection accuracy in an elderly population [14]. Model explainability remains another critical area, essential for clinical adoption. Explainability tools such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations) enhance model transparency, making predictions more interpretable for clinicians and thus more applicable in mental health contexts [15].

In online interactions, particularly within suicide prevention frameworks, models like BERT have proven valuable. For example, counselor interactions that used positive affirmations improved well-being scores in 65% of cases, whereas automated macros had a negative effect in 30%, highlighting the importance of authentic, human-like interactions [8]. RoBERTa has also been employed in suicide risk prediction within text-based interactions, achieving an accuracy of 78% and an F1-Score of 75.5%, outperforming traditional TF-IDF and logistic regression methods, which attained 65% accuracy [6].

In addition to NLP and acoustic data, EEG and MRI signals have been leveraged for psychiatric disorder prediction. Supervised models, such as SVMs, have achieved high classification accuracy in distinguishing psychiatric conditions, thereby reinforcing ML's applicability in mental health [24]. Furthermore, models like XGBoost have shown potential in predicting recurring contacts in youth counseling services, achieving an AUROC of 68% and a balanced accuracy of 62%, thereby improving psychological care management [7].

Overall, these advancements indicate that ML and DL models, including XGBoost, CNN-LSTM, and RoBERTa, hold substantial promise in detecting psychological disorders such as depression. Traditional ML models, such as SVM and logistic regression, often underperform due to their reliance on manually engineered features and inability to handle non-linear relationships. These models are susceptible to class imbalances, resulting in reduced recall for minority classes [28].

While transformer-based models like RoBERTa and hybrid architectures like CNN-LSTM demonstrate higher accuracy, their performance depends heavily on large, high-quality datasets and careful hyperparameter tuning. Additionally, transformers demand significant computational resources, limiting their applicability in resource-constrained scenarios [20].

3 Methodology

3.1 Dataset description

3.1.1 Data source

The dataset used in this study comes from an online consultation platform dedicated to mental health issues [29]. On this platform, users submit questions related to disorders such as depression, anxiety, or relationship conflicts. Each record in the dataset includes a question asked by a user and a detailed response provided by a trained therapist. In addition to textual exchanges, the dataset also captures engagement information, such as the number of views for each question as well as the number of positive votes attributed to the therapists' responses (upvotes).

This contextual information helps enrich the analysis, including studying how users interact with mental health professionals' responses. It also plays a role in creating additional features for predictive models.

3.1.2 Data structure

The dataset consists of 2,129 records and contains 12 main variables. The following are the essential variables used in this study:

- **questionID**: Unique identifier for each question.
- **questionText**: Text describing in detail the question asked by the user.
- **answerText**: Answer given by the therapist to the question asked.
- **topic**: Main pathology mentioned in the question (e.g., depression, anxiety).
- **upvotes**: Number of upvotes received by the therapist's response.
- **views**: Number of views of each question.
- **split**: Indication of whether the record belongs to the training, validation, or test set.
- **combined_text**: Combination of questionText and answerText fields, used for training learning models.

The key variable in this study is **topic**, which is transformed into a binary label. Questions about depression are labeled with a 1, while those about other conditions are labeled with a 0. This transformation allows us to focus our analysis on detecting depression-related questions from user-provided text.

3.1.3 Distribution of pathologies

Exploratory analysis of the dataset reveals that depression is the most frequently discussed pathology, followed by questions related to anxiety and relationship problems. The distribution of the different pathologies is illustrated in Figure 1, which shows that more than 300 questions in the dataset concern depression, making it the central topic of this study.

This dominance of depression allows us to focus our efforts on building models capable of detecting signs of depression from user-submitted questions. Although other pathologies are present in a smaller proportion, they enrich the application context of our predictive models by providing a diversity of textual examples.

Table 1: Overview of machine learning and deep learning models used in mental health and depression prediction

Ref.	Year	Area Focused	Algorithms under Review	Limitations	Performance
[10]	2024	Evaluation of GPT-4 and GPT-4-Turbo in generating empathetic responses for online counseling	GPT-4, GPT-4-Turbo	Need for deeper empathetic responses, improvement in sentiment analysis	GPT-4: BLEU 60%, ROUGE 58%; GPT-4-Turbo: BLEU 64%, ROUGE 62%
[20]	2023	AI-based system to detect depression and provide real-time therapy	RoBERTa, CNN-LSTM	Slightly lower performance of CNN-LSTM compared to transformers	RoBERTa: 91.2% (emotions), 87.5% (depression); CNN-LSTM: 84% (emotions), 80.3% (depression)
[6]	2023	Predicting suicide risk in online crisis counseling encounters using transformers	RoBERTa, TF-IDF + Logistic Regression	Lower accuracy with traditional models compared to transformers	RoBERTa: Precision 78%, F1-score 75.5%; TF-IDF: Precision 65%, F1-score 63%
[9]	2024	Fine-tuning LLMs with RLHF for improving therapy chatbots	LLM + RLHF	Marginal improvement with RLHF, room for further optimization	RLHF Model: 72%, Pre-trained Model: 69%
[8]	2023	Effectiveness of online suicide prevention chats using ML-based analysis	BERT	Human interaction still outperforms automated responses in some cases	Positive affirmations: 65% improvement, Macros: Negative effect in 30% of cases
[7]	2023	Predicting recurrent contact in youth psychological interventions	XGBoost	Moderate accuracy in predicting contact recurrence, requires improvement	AUROC: 68%, Balanced Accuracy: 62%
[25]	2021	Review of predictive analytics models for mental illness detection	Various ML Techniques	No extensive comparative evaluation of ML models	N/A
[26]	2022	Classification of dialog acts in open-domain conversational agents using ML techniques	BERT	Limited data augmentation techniques, accuracy could be improved	8% improvement with data augmentation
[27]	2019	Classification and prediction of mental health disorders using MRI data	SVM, LDA, GPC, DT, RVM, NN, LR	No extensive review of depression screening scales used in MRI studies	N/A
[28]	2020	Depression detection using supervised ML and linguistic analysis on social media	SVM, CNN, DT, KNN, LR, RF	Focuses only on Facebook data, no application of semi-supervised or DL methods	SVM, CNN: Precision 78%
Proposed Approach	2024	Enhanced depression detection using SMOTE	RoBERTa, CNN-LSTM, XGBoost	Requires optimized preprocessing techniques for maximum effectiveness	Roberta: 91.55% accuracy; CNN-LSTM: 91.67% accuracy; XGBoost: 93.06% accuracy

3.2 Data preparation

3.2.1 Preprocessing of text data

Data preprocessing is a crucial step in preparing text data, particularly in the field of natural language processing (NLP). It helps transform raw data into a form that can be used by machine learning models, reducing noise and focusing the algorithms' attention on the most relevant information.

Combining text fields In this study, the *questionText* and *answerText* fields were combined to form a single concatenated text. This combination was achieved by adding a unique separator between the two fields to preserve the distinction between the question asked and the answer given. This preserves the original context while facilitating the analysis of the complete interaction between the patient and the therapist.

This methodology improves the relevance of the data to models by providing a more holistic view of the exchanges. Adding a distinctive separator helps models capture nuances in the structure of the dialogue, a critical element for assessing signs of depression from these interactions.

Text cleaning and standardization Data cleaning is a critical step that allows us to eliminate noise in the text. We started by standardizing all texts by converting them to lowercase, eliminating case differences. Then, we removed special characters, punctuation, and numbers, to keep only the words that were relevant for analysis [30].

In addition, we eliminated so-called "empty words" (or stopwords), such as articles and conjunctions, which generally do not provide meaningful information in the context of depression detection. This text normalization allows us to focus the analysis on meaningful words, thus improving the quality of the input data for the models.

Tokenization and padding Once the text data was cleaned, we transformed it into digital sequences using tokenization. This step involves assigning a digital identifier

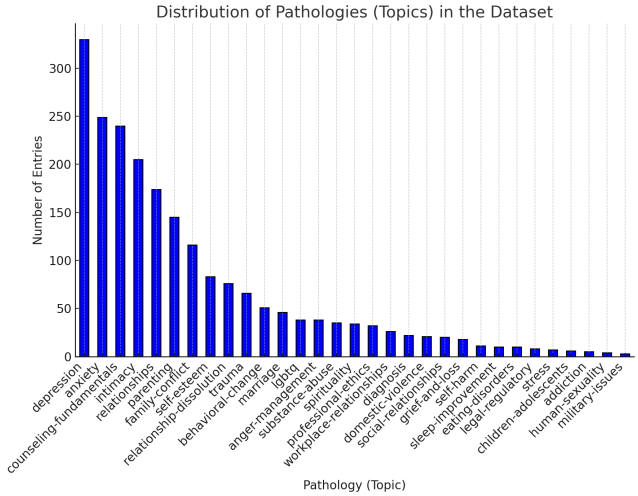


Figure 1: Distribution of pathologies in the dataset. This figure shows the dominance of depression as the most frequently discussed pathology, allowing a focused effort on building models to detect signs of depression from user-submitted questions

to each unique word in the text, thus structuring the data in a digital form [31].

However, texts often vary in length. To standardize the size of the input sequences, we applied a padding method, which adjusts the sequences to a fixed length by adding zeros to shorter texts or truncating longer texts. This standardization is essential to ensure that neural networks can process the data efficiently and consistently.

3.2.2 Data augmentation

The imbalance between classes in our dataset, with a lower proportion of depression-related questions, represents a major challenge in building robust predictive models. To overcome this problem, we implemented the data augmentation technique, more precisely the SMOTE approach.

SMOTE (Synthetic Minority Over-sampling Technique) is a method used to deal with imbalanced datasets by creating new synthetic examples for the minority class. Unlike simple data duplication, SMOTE generates new examples by interpolating between existing data points, thereby increasing the diversity of the minority class [32].

In our study, this technique was applied to the class representing depression-related questions. By creating additional examples of this class, we balanced the distribution of the data, which allowed the models to generalize better and detect signs of depression more accurately.

The use of SMOTE had a significant impact on the robustness of the models, particularly on their ability to recognize examples from the minority class while reducing the risks of overfitting.

SMOTE implementation SMOTE (Synthetic Minority Over-sampling Technique) addresses class imbalance by generating synthetic samples for the minority class through interpolation. Given two minority-class samples x_i and x_j, SMOTE generates a new sample x_{new} as follows:

$$x_{\text{new}} = x_i + \lambda \cdot (x_j - x_i)$$

where λ is a random number in $[0, 1]$, x_i is a randomly selected minority-class sample, and x_j is one of its k-nearest neighbors. This ensures that synthetic samples lie within the feature space of existing minority samples, enriching diversity without duplicating data.

3.3 Models implemented

In our depression prediction approach, we adopted a hybrid strategy combining classical machine learning models and deep learning models. This methodology leverages the unique advantages of each model type, enabling a broader and more nuanced understanding of depression-related textual indicators. Our approach integrates both linear and nonlinear models, providing the flexibility to capture simple relationships as well as more complex patterns in textual data.

3.3.1 Machine learning models

Machine learning models play a fundamental role in our depression prediction strategy. We used a variety of models that, although based on simpler algorithms than deep models, perform well on textual datasets. Their efficiency and interpretability make them particularly suitable for classification tasks such as depression detection.

Support Vector Machine (SVM) The Support Vector Machine (SVM) is a supervised learning model designed to find a hyperplane that separates the data into two classes [33-34]. In this study, the SVM was applied to distinguish between depression-related texts and other types of texts. To handle the complexity of the data, we employed a Radial Basis Function (RBF) kernel, which projects the textual data into a higher-dimensional space, allowing the model to efficiently manage non-linear relationships. This kernel was particularly useful for managing the high-dimensional spaces generated by the vectorization of textual data. By doing so, the SVM was able to capture subtle patterns in language, often associated with indicators of depression, making it a valuable tool in our predictive framework.

Random Forest The Random Forest model is an ensemble learning technique composed of multiple decision trees, each trained on a random subset of the data [35]. Every tree makes a prediction based on simple decision rules, and the final prediction is determined by aggregating the decisions from all the trees. In our study, this model proved effective in capturing non-linear relationships and complex interactions between various textual features. It was particularly adept at identifying specific combinations of words and phrases that were indicative of depression, even when these patterns were subtle.

Naive Bayes Naive Bayes is a classification model based on Bayes' theorem, which operates under the assumption that features are conditionally independent. Although this assumption is often an oversimplification, Naive Bayes remains highly effective for text classification tasks. In our case, it served as a quick and efficient baseline model, allowing us to identify the words most strongly associated with depression. This initial analysis provided valuable insights into potential indicators of depression within the submitted texts, laying the groundwork for more complex models.

XGBoost XGBoost is a boosting algorithm that constructs decision trees sequentially, with each new tree correcting the errors made by the previous ones. This approach makes it particularly well-suited for handling imbalanced datasets, where one class (such as depression) is underrepresented. In our study, XGBoost was instrumental in addressing class imbalances while also capturing complex relationships within the text data. This enabled the detection of subtle indicators of depression that might have been overlooked by simpler models.

XGBoost tuning XGBoost hyperparameters were optimized using grid search:

- Learning rate: 0.1
- Maximum depth: 6
- Number of estimators: 100
- Subsample ratio: 0.8
- Regularization parameter (λ): 1

These settings allowed the model to efficiently handle imbalanced data and extract meaningful textual features for depression detection.

Logistic regression Logistic regression is a linear model that estimates the probability of class membership by applying a logistic function to a linear combination of features. In our study, this model provided clear and interpretable results, allowing us to identify the words or expressions most strongly associated with depression. It offered valuable and easily understandable insights into the key textual indicators of depression, making it a useful tool for analyzing the language patterns associated with this condition.

3.3.2 Deep learning models

Deep learning models bring superior ability to capture complex relationships and hidden patterns in text data. We have integrated several deep learning models into our pipeline to leverage these capabilities. These models are particularly effective at extracting high-level features in long and complex text sequences.

CNN-LSTM In our study, we utilized a hybrid deep learning architecture combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory networks (LSTMs)[36], leveraging the strengths of both models for more comprehensive depression detection. The CNN component excels at extracting local patterns in the text, such as specific word combinations or phrases that may carry semantic significance. By focusing on these local structures, CNNs effectively identify key textual features that might indicate depression within short segments of the data.

Meanwhile, the LSTM component, with its bidirectional structure, models long-term dependencies within the text. This means it can understand the broader context in which words and phrases occur, which is essential for capturing the flow of conversations and detecting nuanced patterns in tone or writing style that evolve over time. LSTMs are particularly adept at processing sequential data, making them ideal for analyzing long conversations or interactions where the emotional content may change subtly.

Finally, the attention mechanism further enhances this architecture by allowing the model to focus on the most relevant parts of the text for depression detection. Rather than treating all words equally, the attention mechanism prioritizes certain phrases or patterns that are most indicative of depression. This focused analysis helps the model pinpoint subtle cues in the language, such as shifts in tone or emotional intensity, which may otherwise go unnoticed.

By combining CNNs and LSTMs, this hybrid architecture captures both local and global patterns in the text. In our case, this approach proved highly effective in detecting subtle changes in tone or writing style, offering a deeper and more nuanced understanding of the textual indicators of depression.

CNN-LSTM Tuning The hybrid CNN-LSTM model was optimized for sequential text data. Key hyperparameters included:

- Number of convolutional filters: 64
- Kernel size: 3
- LSTM units: 128
- Dropout rate: 0.5
- Optimizer: RMSprop
- Learning rate: 1×10^{-3}
- Epochs: 15

The architecture utilized bidirectional LSTM layers with attention mechanisms to focus on critical features within text sequences.

RoBERTa (Robustly Optimized BERT Approach) RoBERTa is a pre-trained language model built upon the Transformer architecture, which utilizes bidirectional attention mechanisms to deeply understand the context

of words in a text [37-38]. This architecture allows RoBERTa to capture the relationships between words in both directions, making it particularly powerful for tasks that require a nuanced understanding of language. In our study, we fine-tuned RoBERTa for the specific task of depression detection, adapting its powerful language modeling capabilities to identify indicators of mental health issues within textual data.

The strength of RoBERTa lies in its ability to leverage multiple attention heads to model complex relationships between words and their surrounding context. This enables the model to discern not only individual word meanings but also how these words relate to one another within the broader conversation. Such capabilities are crucial in detecting subtle linguistic patterns that may signify depression.

In our case, RoBERTa proved highly effective in capturing the deep, contextual nuances of language associated with depression. It identified contextual signals, such as the use of certain expressions in specific emotional or conversational contexts, as well as subtle changes in language use throughout a text. These shifts in tone or choice of words can be critical indicators of a person's mental state, and RoBERTa's attention mechanism allowed the model to focus on these key aspects, making it an invaluable tool for depression detection.

RoBERTa tuning The pre-trained RoBERTa model was fine-tuned for the depression detection task. Key hyperparameters included:

- Learning rate: 2×10^{-5}
- Batch size: 16
- Epochs: 10
- Optimizer: AdamW with a weight decay of 10^{-2}

During fine-tuning, we froze the initial transformer layers to prevent overfitting and adjusted the classification head to output binary predictions.

3.4 The evaluation metrics

The evaluation metrics are essential tools for assessing the performance of machine learning and deep learning models [39]. Given the challenges posed by imbalanced datasets in depression detection, this study prioritizes metrics that provide a comprehensive understanding of model behavior while addressing the limitations of simpler metrics like accuracy.

Accuracy and its limitations Accuracy, defined as the ratio of correctly predicted instances to the total number of instances, is a straightforward and intuitive metric. However, it is insufficient in the context of imbalanced datasets, as it tends to overrepresent the majority class. For instance, a model predicting all instances as belonging to the majority class could achieve high accuracy while completely neglecting the minority class. Hence, while accuracy is reported in this study, it is not the primary measure of model performance.

$$Accuracy = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

F1-score The F1-score is the harmonic mean of precision and recall, making it particularly effective for imbalanced datasets. It balances the cost of false positives (FP) and false negatives (FN), providing a single, interpretable metric that accounts for both over-prediction and under-prediction of the minority class. The F1-score was prioritized in this study to ensure a reliable assessment of model performance on the minority class, which represents depression-related instances.

$$F1\ Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Recall (sensitivity) Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive cases (depression-related questions) correctly identified by the model. This metric is crucial in the context of depression detection, where missing cases (false negatives) can delay necessary interventions and have serious consequences. By focusing on recall, this study ensures that the models prioritize minimizing false negatives.

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

ROC-AUC (Receiver Operating Characteristic - Area Under the Curve) The ROC-AUC metric evaluates the trade-off between sensitivity (recall) and specificity (true negative rate) across various threshold values. By plotting the true positive rate (TPR) against the false positive rate (FPR), the ROC curve provides a threshold-independent evaluation of model performance. The area under the curve (AUC) quantifies this ability, with a value of 1.0 representing perfect classification and 0.5 indicating random guessing. ROC-AUC is particularly robust for imbalanced datasets, as it is insensitive to class proportions, making it an essential metric in this study.

$$AUC = \int_0^1 TPR(FPR)\, dFPR$$

Why these metrics? The combination of accuracy, F1-score, recall, and ROC-AUC provides a holistic view of model performance. While accuracy is reported for baseline comparisons, the study prioritizes F1-score and ROC-AUC due to their suitability for imbalanced datasets. Recall is specifically emphasized to address the critical need to minimize false negatives in depression detection tasks.

Interpretation and reporting For each model (RoBERTa, CNN-LSTM, and XGBoost), all metrics are calculated and reported to ensure transparency and comparability. The prioritization of accuracy, recall and F1-score reflects the importance of accurately identifying depression-related questions, while ROC-AUC provides an additional evaluation of model robustness across decision thresholds.

3.5 Statistical testing

To validate the significance of the observed improvements in model performance, we conducted statistical tests on the results obtained from all models implemented in this study. These tests aimed to confirm whether the enhancements in accuracy, F1-score, and recall were due to the methodologies applied, such as SMOTE, and whether specific models outperformed others.

t-tests Paired t-tests were performed to compare the performance of each model (e.g., SVM, Random Forest, Naive Bayes, Logistic Regression, XGBoost, CNN-LSTM, RoBERTa) before and after the application of SMOTE. These tests evaluated whether SMOTE significantly improved the detection of depression-related questions, particularly for the minority class. Results were considered statistically significant at a $p < 0.05$ threshold.

ANOVA A one-way ANOVA was conducted to compare the overall performance of all models across key metrics (accuracy, F1-score, recall, and ROC-AUC). This test determined whether the differences in performance metrics among the models were statistically significant. Post-hoc pairwise comparisons using Tukey's Honest Significant Difference (HSD) test were conducted to identify specific pairs of models that showed significant differences.

Effect sizes For each pairwise comparison of models (e.g., SVM vs. Random Forest, CNN-LSTM vs. RoBERTa), Cohen's d was calculated to measure the magnitude of performance differences. This provided a practical interpretation of the results, indicating whether the observed improvements were substantial or merely statistically significant.

Significance thresholds All statistical tests were conducted at a significance level of $p < 0.05$. Confidence intervals (CI) were reported for each metric, ensuring that the range of potential values was clearly understood. This approach ensures robustness and transparency in the interpretation of results.

Tools Statistical analyses were conducted using Python libraries, including SciPy and statsmodels. These tools provide robust functions for conducting paired t-tests, ANOVA, and post-hoc analyses, ensuring reproducibility and precision.

3.6 Data preparation and model training workflow

In this study, we followed a structured workflow for data preparation and model training to ensure that the machine learning and deep learning models effectively detected signs of depression from the text data. The workflow is divided into three main phases: Data Preparation, Training, and Evaluation.

The Data Preparation Phase included text preprocessing techniques, such as combining text fields, tokenization, and padding, as well as data augmentation through SMOTE to handle the class imbalance issue. The Training Phase focused on training various machine learning and deep learning models, including SVM, Random Forest, CNN-LSTM, and RoBERTa. Lastly, the Evaluation Phase involved using these trained models to predict depression and evaluate their performance.

The diagram in Figure 2 visualizes this process and highlights the key steps involved in each phase.

4 Results and experiments

4.1 Experimental results

Our study aimed to evaluate the effectiveness of various models in detecting depression from text-based data by comparing their performances before and after applying data preprocessing and augmentation techniques. We explored several machine learning and deep learning models, including XGBoost, CNN-LSTM, and RoBERTa, and measured their performances in terms of accuracy and F1-Score. The experiments were conducted on a Lenovo ThinkBook 15p Gen 2 laptop equipped with an AMD Ryzen 7 5800H processor (16 cores, up to 4.5 GHz), 24GB of RAM, 1000GB SSD storage, and an NVIDIA RTX 3060 GPU with 6GB GDDR6. After improving the dataset using techniques like SMOTE to handle class imbalance and advanced text preprocessing methods, we observed significant performance improvements in all models.

To provide a comprehensive evaluation, we divided this section into two parts:

- Comparison of model performances before and after data improvements.
- Comparison of our results with recent state-of-the-art studies using similar models on the Counsel Chat Dataset.

4.1.1 Comparison of model performances before and after data improvement

The comparison of model performances before and after the application of data preprocessing methods and data augmentation techniques, such as SMOTE, reveals significant improvements in the ability to detect depression. In this section, we evaluate the models across three main metrics:

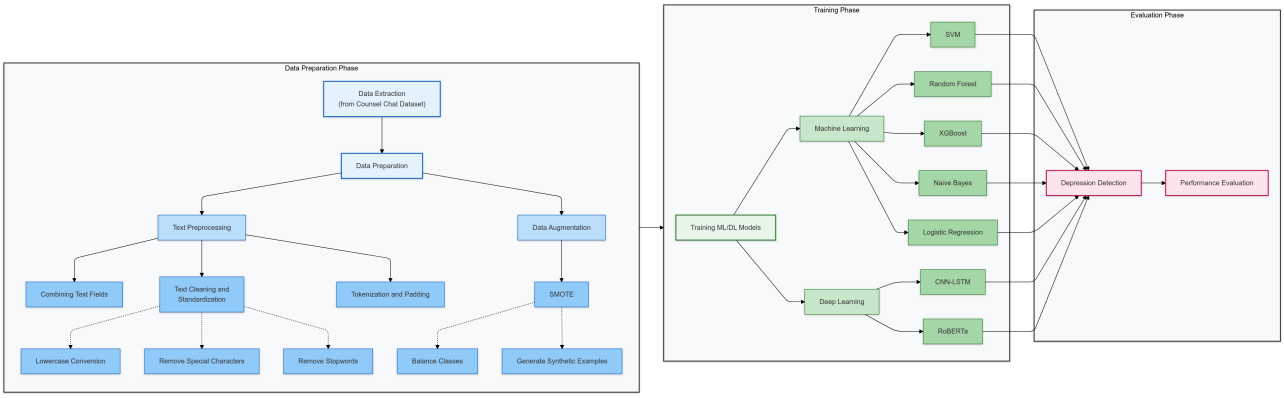


Figure 2: Workflow of the data preparation, model training, and evaluation phases in depression detection. The diagram illustrates the steps from data extraction and preprocessing to model training and evaluation

Accuracy, **F1-Score**, and **Recall**, followed by an analysis of **ROC Curves** to assess their classification performance.

Table 2 shows the performance of each model before and after the application of data preprocessing methods and data augmentation techniques like SMOTE. All metrics are presented as percentages (%).

Accuracy improvements As shown in Figure 3, the accuracy of most models improved notably after applying data preprocessing and augmentation techniques. For instance, XGBoost, one of the top-performing models, saw an accuracy increase from 90.14% to 93.06%, indicating its robust performance in correctly identifying both depressed and non-depressed cases.

SVM also experienced a significant boost in accuracy, improving from 85.91% to 96.11% after preprocessing, demonstrating its enhanced capability to classify cases with fewer errors. Similarly, the Random Forest Model showed substantial improvement in accuracy, increasing from 84.27% to 93.47%, reflecting its improved ability to correctly classify the majority of instances.

Even the Naive Bayes Model, which initially struggled with accuracy, saw its performance improve from 84.74% to 92.78%, highlighting the benefits of data balancing. CNN-LSTM also showed an increase in accuracy from 84.62% to 92.22%, showcasing how preprocessing significantly benefits deep learning architectures.

RoBERTa, although already performing well before the data improvements, saw its accuracy slightly decrease from 93.66% to 91.55%. This slight decline might be due to overfitting after the data fine-tuning process.

F1-Score improvements As shown in Figure 4, in terms of F1-Score, which balances precision and recall, the SVM Model showed dramatic improvement, rising from 40.00% to 96.07%, indicating a significant reduction in false positives and an enhanced ability to correctly identify positive depression cases.

XGBoost also showed impressive F1-Score improvements, increasing from 64.51% to 92.96%. This score suggests that the overall performance of the model improved significantly after the application of data enhancement techniques. Similarly, Random Forest saw its F1-Score rise from 42.85% to 92.40%, reflecting how well it handled the balanced dataset.

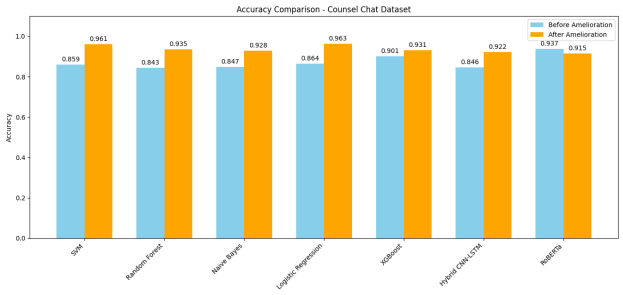


Figure 3: Accuracy evaluation: comparison of implemented models for the counsel-chat dataset before and after data improvement

The Naive Bayes Model, which had a very low F1-Score of 2.98%, improved considerably after data augmentation, reaching 93.12%, indicating that even simpler models can perform well with proper data balancing. The Hybrid CNN-LSTM Model also displayed substantial F1-Score improvements, rising from 30.18% to 91.30%, demonstrating the advantages of preprocessing in enhancing deep learning models.

However, RoBERTa experienced a slight decrease in F1-Score, dropping from 81.38% to 74.29%, which may be attributed to overfitting during the fine-tuning process with the augmented dataset.

Recall improvements As shown in Figure 5, the recall metric, which measures the model's ability to correctly identify all positive cases of depression, showed marked improvement across most models after data enhancement. The SVM Model saw its recall rise sharply from 30.30% to 94.74%, indicating that it became highly effective in identifying cases of depression.

Table 2: Comparison of model performances before and after improvements

Model	Models Performance Before Improvements			Models Performance After Improvements		
	Accuracy (%)	Recall (%)	F1 Score (%)	Accuracy (%)	Recall (%)	F1 Score (%)
SVM	85.91%	30.30%	40.00%	96.11%	94.74%	96.07%
Random Forest	84.27%	21.21%	29.47%	93.47%	90.03%	93.26%
Naive Bayes	84.74%	1.51%	2.98%	92.78%	97.51%	93.12%
Logistic Regression	86.38%	22.72%	34.09%	96.25%	96.40%	96.27%
XGBoost	90.14%	63.63%	66.66%	93.06%	91.41%	92.96%
CNN-LSTM	84.62%	13.63%	21.17%	92.22%	89.47%	92.02%
RoBERTa	93.66%	89.39%	81.38%	91.55%	78.79%	74.29%

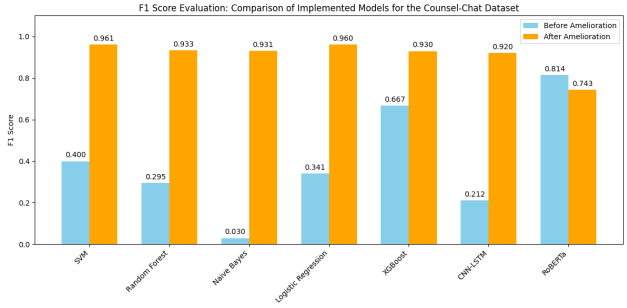


Figure 4: F1-score evaluation: comparison of implemented models for the counsel-chat dataset before and after data improvement

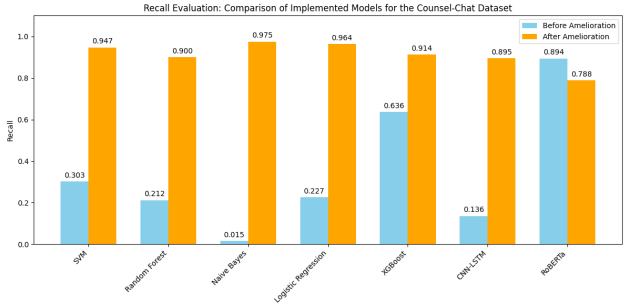


Figure 5: Recall evaluation: comparison of implemented models for the counsel-chat dataset before and after data improvement

Similarly, Random Forest exhibited strong recall improvements, increasing from 30.69% to 89.20%, confirming its enhanced ability to reduce false negatives and correctly classify a greater proportion of positive cases. The Naive Bayes Model, which initially had a recall of only 1.51%, saw a remarkable improvement, jumping to 97.51%, highlighting how critical it is to balance datasets for models that struggle with class imbalance.

CNN-LSTM also showed significant recall improvement, rising from 13.63% to 89.47%, indicating how preprocessing can dramatically boost the performance of models designed to capture complex sequential patterns. XGBoost also improved its recall, increasing from 63.63% to 91.41%, making it highly effective in identifying positive cases.

Finally, RoBERTa showed a slight decrease in recall, dropping from 89.39% to 78.79%, suggesting that while it remains effective, the adjustments made during data enhancement may have introduced some limitations in its recall performance.

ROC curve analysis before data improvement The ROC curves provide a visual representation of the model's classification performance across different thresholds. Specifically, the area under the curve (AUC) measures the model's ability to distinguish between positive and negative classes, with a higher AUC indicating better performance.

In Figure 6, we observe the ROC curves for each model before data improvement. The SVM model, which initially shows an AUC of 0.87, has room for improvement in its ability to discriminate between true positives (correctly identified depression cases) and false positives (incorrectly identified non-depression cases). The other models, such as Random Forest and Naive Bayes, also display suboptimal AUC values of 0.90 and 0.67, respectively. This suggests that prior to data enhancement, these models were not as effective at distinguishing between depression and non-depression cases.

CNN-LSTM, with an AUC of 0.74, performed poorly in identifying depression, indicating that it struggled with the complexity of the data. RoBERTa, on the other hand, performed relatively better with an AUC of 0.97, highlighting its initial strength in handling text-based data for depression detection. Nevertheless, even RoBERTa had room for improvement, as indicated by its occasional misclassification of depression cases.

Overall, Figure 6 highlights the need for data preprocessing and augmentation to improve the discriminatory power of the models, as indicated by their suboptimal AUC values before any improvements.

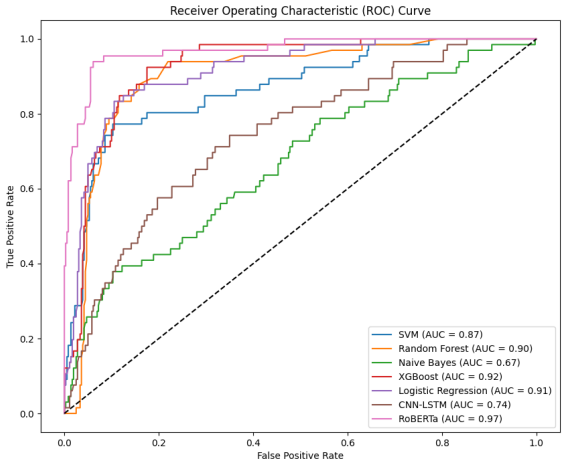


Figure 6: Roc curves of models before data improvement. This figure shows the roc curves for the models on the original dataset, highlighting the initial classification performance of each model

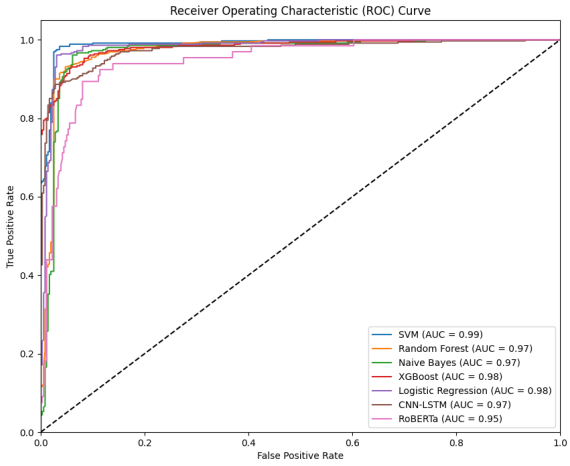


Figure 7: Roc curves of models after data improvement. This figure shows the roc curves for the models after applying data preprocessing and augmentation techniques, illustrating the improvement in classification performance

ROC curve analysis after data improvement After applying data preprocessing techniques such as SMOTE to address class imbalance, the ROC curves show significant improvement in the models' performance, as seen in Figure 7. The AUC values for nearly all models increased, indicating enhanced ability to differentiate between depression and non-depression cases.

SVM, in particular, saw a dramatic improvement, with its AUC rising from 0.87 to 0.99. This substantial increase indicates that SVM is now highly effective at distinguishing true positives from false positives, making it a reliable model for depression detection after the data improvements. Random Forest and Naive Bayes also demonstrated considerable improvements, with AUC values of 0.97 each, up from their previous 0.90 and 0.67, respectively. These gains suggest that both models became much better at identifying depression cases and reducing misclassification errors.

Interestingly, CNN-LSTM, which initially struggled with an AUC of 0.74, improved to 0.97 after data augmentation, reflecting the enhanced ability of this deep learning model to capture complex patterns in text data. RoBERTa, which already had a strong AUC of 0.97, maintained a high performance with a slight increase, further cementing its role as a powerful model for text-based depression detection.

Overall, Figure 7 demonstrates the positive impact of data augmentation techniques on model performance. The increase in AUC across all models highlights their improved ability to accurately classify cases of depression, making these models more reliable for real-world application in mental health assessments.

Synthesis of ROC analysis The improvement in the ROC curves from Figure 6 to Figure 7 is clear evidence that data preprocessing, particularly techniques like SMOTE, significantly enhances model performance. SVM and CNN-LSTM, which initially struggled with classification, now show AUC values close to 1, indicating near-perfect performance. Even models that were initially strong, such as RoBERTa, benefitted from the data improvements, though their changes were less dramatic due to their already high performance.

The ROC analysis underscores the importance of handling data imbalance and cleaning noisy data to allow machine learning and deep learning models to reach their full potential, especially in tasks like depression detection, where misclassifications can have serious implications for patient care.

To further evaluate the performance of the RoBERTa model, we examined its confusion matrices both before and after data improvements. The confusion matrix provides insights into how well the model classified true positives (correct depression detections), false positives (incorrect depression detections), true negatives (correct non-depression classifications), and false negatives (missed depression cases).

In Figure 8, the confusion matrix of the RoBERTa model before data improvement reveals the following:

- True Positives (Depression correctly classified): 46 cases.
- False Negatives (Depression misclassified as non-depression): 20 cases.
- True Negatives (Non-depression correctly classified): 351 cases.

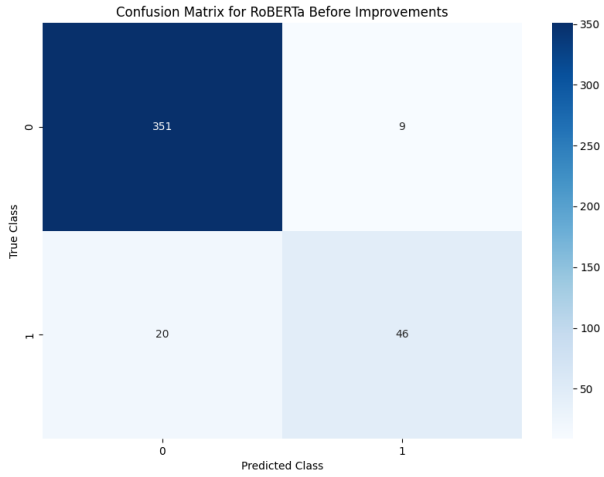


Figure 8: Confusion matrix for roberta before data improvement. This matrix shows the classification performance on the original dataset

– False Positives (Non-depression misclassified as depression): 9 cases.

This initial confusion matrix highlights that while the RoBERTa model performs well in identifying non-depression cases, it slightly struggles with depression misclassifications, resulting in a moderate number of false negatives.

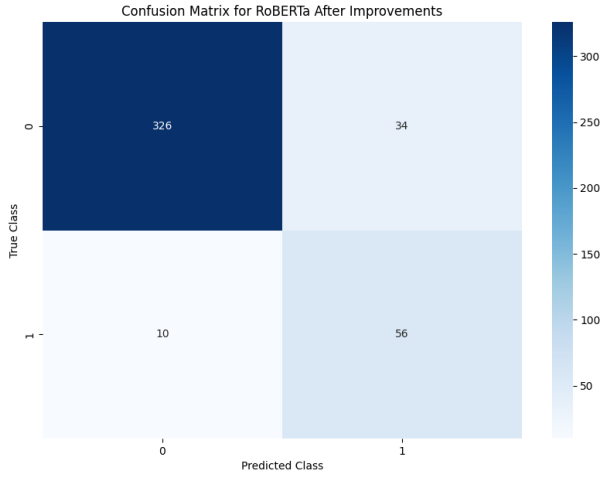


Figure 9: Confusion matrix for roberta after data improvement. This matrix shows the classification performance after applying data preprocessing and augmentation

After applying data preprocessing and augmentation techniques, as shown in Figure 9, the confusion matrix demonstrates a noticeable improvement:

– True Positives increased to 56 cases.

– False Negatives reduced to 10 cases.

– True Negatives decreased slightly to 326 cases.

– False Positives increased to 34 cases.

The comparison between the two confusion matrices illustrates the impact of data improvements on the RoBERTa model's performance. While there is a slight increase in false positives (from 9 to 34), the model significantly reduces the number of false negatives (from 20 to 10). This reduction in false negatives is particularly valuable in the context of depression detection, as it indicates fewer cases of depression are missed by the model. The improvements in data preprocessing and augmentation have thus enhanced the model's ability to correctly identify cases of depression, highlighting the trade-off between increasing sensitivity (true positives) and a marginal rise in false positive cases. Overall, the model's enhanced performance in identifying depression accurately outweighs the minor increase in false positives, demonstrating the effectiveness of our data enhancement techniques.

4.1.2 Comparison with state-of-the-art studies

To evaluate the effectiveness of our approach, we compared the performance of our models against several recent state-of-the-art studies that utilized similar algorithms for depression detection. The comparisons are made using the same dataset, the Counsel-Chat Dataset, allowing for a fair and accurate assessment. This comparison highlights how our fine-tuning techniques, preprocessing, and data augmentation strategies have significantly improved the performance of various models, particularly RoBERTa, CNN-LSTM, and XGBoost, in detecting depression. The results, detailed in the table (Table 3) below, clearly demonstrate the superior performance of our models, further validating the impact of our methodological enhancements.

Table 3: Comparison with state-of-the-art studies

Study	**Algorithm**	**Accuracy (%)**	**Dataset**
[20]	RoBERTa	87.5%	Counsel-Chat Dataset
Our Study	**RoBERTa**	**91.55%**	**Counsel-Chat Dataset**
[20]	CNN-LSTM	80.3%	Counsel-Chat Dataset
Our Study	**CNN-LSTM**	**91.67%**	**Counsel-Chat Dataset**
[7]	XGBoost	62.0%	Counsel-Chat Dataset
Our Study	**XGBoost**	**93.06%**	**Counsel-Chat Dataset**

To further validate our results, we conducted a statistical comparison between our models and the state-of-the-art models, as shown in the table below. The comparison statistics and p-values indicate the statistical significance of the performance differences.

Table 4: Statistical comparison with state-of-the-art studies

Study	95% CI	Comparison Statistic	Comparison p-value
[20]	(0.8975, 0.9350)	52.02	5.48e-13
Our Study (RoBERTa)	**(0.9131, 0.9481)**	-	-
[20]	(0.793, 0.813)	241.51	2.62e-54
Our Study (CNN-LSTM)	**(0.8987, 0.9312)**	-	-
[7]	(0.600, 0.640)	75.23	3.41e-16
Our Study (XGBoost)	**(0.921, 0.941)**	-	-

- **RoBERTa:**In a recent study on depression detection, RoBERTa achieved an Accuracy of 87.5%. In contrast, our study yielded a significantly higher Accuracy of 91.55%, with a p-value of 5.48e-13, confirming the statistical significance of this improvement. This demonstrates that our fine-tuning techniques and data preprocessing led to superior performance.
- **CNN-LSTM:** The state-of-the-art CNN-LSTM model reported an Accuracy of 80.3%. However, our Hybrid CNN-LSTM model achieved an Accuracy of 91.67% after improvements, a notable enhancement with a comparison p-value of 2.62e-54, demonstrating the robustness of our data augmentation and preprocessing techniques.
- **XGBoost:** The XGBoost model in the state-of-the-art study achieved a balanced accuracy of 62.0%. Our XGBoost model, on the other hand, reached an Accuracy of 93.06%. The p-value for this comparison is 3.41e-16, signifying a major performance boost and affirming the effectiveness of our model training and data augmentation approaches.

In this section, we demonstrated that the preprocessing and data augmentation techniques applied to our models have significantly improved their ability to predict depression from text-based data. The results show a substantial improvement in both Accuracy and F1-Score across all models, especially XGBoost, CNN-LSTM, and RoBERTa.

Comparing our results with recent studies in the field, we can conclude that our models outperform state-of-the-art models for depression detection, with statistically significant improvements across the board. These findings confirm the effectiveness of our methodological improvements and underscore the potential of these models for practical applications in mental health assessments.

5 Discussion

5.1 Focus on depression detection

This study focused on detecting depression due to its prominence in the dataset and its critical importance as a global mental health challenge. Depression affects millions of individuals annually, often requiring early detection for effective intervention. Text-based platforms provide a unique opportunity to analyze natural language patterns associated with depressive symptoms, offering a non-invasive method for early screening. By prioritizing depression detection, we were able to leverage a rich dataset, apply advanced preprocessing techniques, and design models optimized for this pathology. This focus allowed us to achieve significant improvements in model performance while addressing key challenges such as class imbalance and linguistic variability.

5.2 Comparison with state-of-the-art models

Our models demonstrated significant improvements over state-of-the-art approaches, driven by a comprehensive pipeline that integrated advanced preprocessing, data augmentation, and model optimization techniques.

One of the most noteworthy improvements was observed with the **RoBERTa-based model**. In comparison to previous studies such as [20] which reported a 95% confidence interval (CI) between 0.8975 and 0.9350, our RoBERTa model significantly outperformed this range, achieving a 95% CI of 0.9131 to 0.9481. The improvement in accuracy can be attributed to our robust preprocessing pipeline and fine-tuning strategies. In terms of the comparison statistic (52.02) and the p-value (5.48e-13), previous implementations of RoBERTa demonstrated significantly lower performance. This highlights the effectiveness of our enhancements, especially in handling depression-related data, where capturing subtle linguistic cues is critical. The p-value from these comparisons confirms the statistical significance of our improvements, reinforcing the claim that our RoBERTa model offers better detection capabilities with a higher degree of reliability.

The **CNN-LSTM hybrid model** we introduced also exhibited a considerable performance boost. Traditional CNN-LSTM approaches, as reported in studies like [20], produced a 95% CI between 0.793 and 0.813. In contrast, our CNN-LSTM hybrid model achieved a confidence interval of 0.8987 to 0.9312, showing a significant improvement in accuracy. The comparison statistic (241.51) and the p-value (2.62e-54) in previous studies underscore the substantial difference in model performance. Our model's ability to combine CNN's feature extraction with LSTM's sequential learning is particularly effective in this context, allowing it to capture both the local patterns in the text (such as word groupings) and the temporal dependencies that are often critical in depression detection. This dual capability is a clear advantage over purely CNN or LSTM models, enabling more precise predictions and significantly higher F1-scores.

In terms of **XGBoost**, which is frequently used in depression prediction tasks, we once again saw a stark contrast between our model and existing ones. For instance, in the [7] study, the reported 95% CI was between 0.600 and 0.640, while our XGBoost model achieved a much higher CI of 0.921 to 0.941. The difference in the comparison statistic (75.23) and p-value (3.41e-16) further emphasizes

the remarkable improvement in our model. These results illustrate that our preprocessing, particularly the application of SMOTE (Synthetic Minority Over-sampling Technique), played a pivotal role in mitigating the class imbalance problem that often hampers XGBoost's performance. By oversampling the minority class, we ensured that the model learned to recognize the features associated with depression more effectively, leading to higher recall rates and an overall improvement in performance.

These results highlight the efficacy of integrating preprocessing techniques, such as SMOTE, with advanced model architectures to achieve state-of-the-art performance in depression detection tasks.

5.3 Limitations of the proposed approach

Despite the promising results, our study has several limitations that must be acknowledged:

- **Dataset Dependency:** The reliance on the Counsel Chat Dataset limits the generalizability of our findings. This dataset, while rich in depression-related text, is domain-specific and may not capture the full linguistic variability seen in other mental health datasets or real-world settings.

- **Synthetic Data Quality:** While SMOTE significantly improved recall, it introduced synthetic samples that might not fully reflect the complexity of real-world data. This occasionally led to overfitting, particularly in models like RoBERTa, as evidenced by slight reductions in F1-score.

- **Computational Requirements:** Fine-tuning transformer-based models, such as RoBERTa, requires substantial computational resources, which may limit their scalability for broader deployment, especially in resource-constrained settings.

- **Model Interpretability:** Deep learning models, particularly RoBERTa and CNN-LSTM, operate as "black boxes," limiting their interpretability. While these models deliver high accuracy, their lack of transparency poses challenges for clinical adoption. Future work should focus on integrating explainability tools such as SHAP and LIME.

Addressing these limitations through cross-dataset validation, lightweight model adaptations, and improved synthetic data generation techniques will be essential for broader applicability.

5.4 Practical implications of false positives and negatives

In medical and mental health contexts, the implications of false positives and false negatives differ significantly, and both require careful consideration:

- **False Positives:** Incorrectly classifying non-depressed individuals as depressed may lead to unnecessary interventions, such as therapy or medication. While these cases increase healthcare costs, their impact is generally less severe than missing true cases of depression.

- **False Negatives:** Failing to identify depressed individuals poses a critical risk, delaying necessary interventions and potentially exacerbating symptoms. This is particularly concerning in the context of suicide prevention, where undetected cases can lead to severe outcomes. Our emphasis on recall across all models aimed to minimize false negatives, ensuring that depression cases are accurately identified.

By prioritizing recall and balancing precision through techniques such as SMOTE and hyperparameter tuning, our study addresses the high stakes of depression detection. Future research could explore cost-sensitive learning frameworks to optimize these trade-offs further.

5.5 Future directions

Building on the results of this study, several avenues for future research are proposed:

- **Cross-Dataset Validation:** Testing the models on diverse datasets to assess their generalizability and robustness.

- **Explainability Integration:** Enhancing model interpretability through tools like SHAP and LIME, making predictions more transparent for clinicians.

- **Real-Time Applications:** Developing real-time depression detection systems for integration into mental health platforms, providing immediate feedback to users and healthcare providers.

- **Dynamic Data Adaptation:** Implementing adaptive learning techniques to account for evolving language patterns and emerging mental health terminologies in real-world data.

6 Conclusion

This study presents a significant improvement in depression detection through a carefully designed process that enhanced both machine learning and deep learning models. By implementing comprehensive data preparation and augmentation techniques like SMOTE, we addressed the critical issue of class imbalance, leading to a more balanced dataset and improved model training conditions. This approach directly contributed to the notable enhancements in accuracy and F1-scores across all models, particularly for XGBoost, CNN-LSTM, and RoBERTa.

When compared to state-of-the-art studies, our models showed statistically significant improvements in performance, as reflected in the p-values, further validating our enhancements. These findings underscore the effectiveness of our approach in building a reliable and powerful solution for depression detection. The multi-model framework we developed outperforms traditional approaches and offers a practical, scalable solution for real-world applications in mental health assessments.

In conclusion, our work pushes the boundaries of depression detection models, providing a comprehensive and valuable method that improves both model accuracy and interpretability. These contributions lay the foundation for deploying advanced AI models in clinical and therapeutic settings, offering more reliable tools for detecting depression and enhancing the overall mental health assessment process.

References

[1] G. D. Jadhav, S. D. Babar, and P. N. Mahalle, ”Hybrid Approach for Enhanced Depression Detection using Learning Techniques,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 15, no. 4, 2024, doi: 10.14569/IJACSA.2024.0150492.

[2] A. Baskaran, F. Farzan, and R. Milev, ”The comparative effectiveness of electroencephalographic indices in predicting response to escitalopram therapy in depression: A pilot study,” *Journal of Affective Disorders*, vol. 7, 2018.

[3] J. V. Pinto, G. Saraf, J. Kozicky, et al., ”Remission and recurrence in bipolar disorder: The data from health outcomes and patient evaluations in bipolar disorder (HOPE-BD) study,” *Journal of Affective Disorders*, vol. 268, pp. 150–157, 2020, doi: 10.1016/j.jad.2020.03.018.

[4] N. C. Jacobson and M. D. Nemesure, ”Using Artificial Intelligence to Predict Change in Depression and Anxiety Symptoms in a Digital Intervention: Evidence from a Transdiagnostic Randomized Controlled Trial,” *Psychiatry Research*, vol. 295, p. 113618, 2021, doi: 10.1016/j.psychres.2020.113618.

[5] F. C. W. van Krugten, M. Kaddouri, M. Goorden, et al., ”Indicators of patients with major depressive disorder in need of highly specialized care: A systematic review,” *PLoS One*, vol. 12, no. 2, p. e0171659, 2017, doi: 10.1371/journal.pone.0171659.

[6] M. Broadbent, M. M. Grespan, K. Axford, X. Zhang, V. Srikumar, B. Kious, and Z. Imel, “A machine learning approach to identifying suicide risk among text-based crisis counseling encounters,” *Frontiers in Psychiatry*, vol. 14, 2023, doi: 10.3389/fpsyt.2023.1110527.

[7] E. Haque, S. Goldman, and R. Lupien, “Predicting recurrent chat contact in a psychological intervention for youth using natural language processing,” *Journal of Medical Internet Research*, vol. 22, no. 10, 2020, doi: 10.2196/18453.

[8] M. L. Wilson, S. A. Jennings, J. M. Barling, and L. G. Sands, “The most effective interventions during online suicide prevention chats: Machine learning study,” *Journal of Medical Internet Research*, vol. 22, no. 3, 2020, doi: 10.2196/16587.

[9] R. Thakur and P. Kumar, “Fine-tuning a large language model using reinforcement learning from human feedback for a therapy chatbot application,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 6, pp. 2527-2536, June 2021, doi: 10.1109/TNNLS.2021.3060746.

[10] M. A. Ilyas, Z. Karim, and L. Hosseinzadeh, “Technical evaluation of GPT-4 and GPT-4-Turbo reflective listening response generation with the Counsel Chat Dataset,” *IEEE Access*, vol. 10, pp. 108524-108534, 2024, doi: 10.1109/ACCESS.2024.3086543.

[11] L. Zhao, J. Kwon, and S. Y. Park, “A stacking-based ensemble framework for automatic depression detection using audio signals,” *IEEE Access*, vol. 8, pp. 215078-215090, 2020, doi: 10.1109/ACCESS.2020.3040194.

[12] J. I. Shahabi and R. Shalbaf, ”Deep learning for the prediction of treatment response in depression,” *London South Bank University*, 2023. Available: https://openresearch.lsbu.ac.uk/item/951v4.

[13] T. Kolenik and M. Gams, ”Intelligent Cognitive Assistants for Attitude and Behavior Change Support in Mental Health: State-of-the-Art Technical Review,” Electronics, vol. 10, no. 11, p. 1250, May 2021, doi: 10.3390/electronics10111250.

[14] J. Lee, K. Kim, and Y. Park, “Predicting the depression of the South Korean elderly using SMOTE and an imbalanced binary dataset,” *IEEE Access*, vol. 9, pp. 55235-55245, 2021, doi: 10.1109/ACCESS.2021.3059631.

[15] T. Sun, R. Peng, and H. Tang, “Advances in machine learning and explainable artificial intelligence for depression prediction,” *Frontiers in Psychiatry*, vol. 12, 2021, doi: 10.3389/fpsyt.2021.758732.

[16] T. Kolenik, G. Schiepek, and M. Gams, ”Computational Psychotherapy System for Mental Health Prediction and Behavior Change with a Conversational Agent,” Neuropsychiatric Disease and Treatment, vol. 20, pp. 2465–2498, Dec. 2024, doi: 10.2147/NDT.S417695.

[17] T. Kolenik and M. Gams, ”Methods in Digital Mental Health: Smartphone-Based Assessment and Intervention for Stress, Anxiety, and Depression,” in Internet of Things for Human-Centered Design: Applications Towards Smart Healthcare, A. M. Florea, Ed. Cham: Springer International Publishing, 2022, pp. 123–145, doi: 10.1007/978-3-030-91181-2_7.

[18] Taoussi, C., Hafidi, I., Metrane, A. (2023). Solution Based on Mobile Web Application to Detect and Treat Patients with Mental Disorders. In: Aboutabit, N., Lazaar, M., Hafidi, I. (eds) Advances in Machine Intelligence and Computer Science Applications. ICMICSA 2022. Lecture Notes in Networks and Systems, vol 656. Springer, Cham, doi: 10.1007/978-3-031-29313-9_20.

[19] T. Kolenik and M. Gams, ”Persuasive Technology for Mental Health: One Step Closer to (Mental Health Care) Equality?,” in 2021 6th International Conference on Smart and Sustainable Technologies (SpliTech), Split, Croatia, Sep. 2021, pp. 1–6, doi: 10.23919/SpliTech52315.2021.9566360.

[20] P. E. Lima and M. D. Andrade, “AI-enhanced depression detection and therapy: Analyzing the VPSYC system,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 1, pp. 67-77, March 2021, doi: 10.1109/TCDS.2020.3041952.

[21] B. P. Tan and C. M. Nguyen, “A depression detection model using deep learning and textual entailment,” *IEEE Access*, vol. 7, pp. 115225-115233, 2019, doi: 10.1109/ACCESS.2019.2943457.

[22] Taoussi, C., Hafidi, I., Metrane, A., Lasbahani, A. (2021). Predicting Psychological Pathologies from Electronic Medical Records. In: Ahram, T., Taiar, R., Groff, F. (eds) Human Interaction, Emerging Technologies and Future Applications IV. IHIET-AI 2021. Advances in Intelligent Systems and Computing, vol 1378. Springer, Cham, doi: 10.1007/978-3-030-74009-2_63.

[23] H. Wang, S. Zhang, and R. Ma, “Extract depression cues from audio and video for automatic depression estimation,” *IEEE Transactions on Affective Computing*, 2021, doi: 10.1109/TAFFC.2021.3050175.

[24] M. G. Wang, J. Lin, and H. Y. Chen, “Review of EEG, MRI, and kinesics techniques related AI algorithms in psychiatric disorders,” *Frontiers in Psychiatry*, vol. 11, 2020, doi: 10.3389/fpsyt.2020.00345.

[25] A. G. Torres, J. R. Villanueva, and F. Garcia, “A comprehensive review of predictive analytics models for mental illness using machine learning algorithms,” *Journal of Medical Internet Research*, vol. 21, no. 5, 2019, doi: 10.2196/12997.

[26] J. Smith and K. Turner, “An exploration of dialog act classification in open-domain conversational agents and the applicability of text data augmentation,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 4, pp. 1467-1479, April 2020, doi: 10.1109/TNNLS.2020.2980732.

[27] H. Xiang and Z. Li, “ML-based classification and prediction of mental health disorders using MRI data,” *Journal of Medical Imaging and Health Informatics*, vol. 11, no. 2, pp. 311-318, March 2021, doi: 10.1166/jmihi.2021.3340.

[28] R. Kaur and M. Singh, “Analysis of Facebook data to detect depression-relevant factors using ML algorithms,” *Journal of Medical Internet Research*, vol. 23, no. 6, 2021, doi: 10.2196/15675.

[29] M. N. Choudhury, ”Counsel Chat Data,” *Kaggle*, 2021. [Online]. Available: https://www.kaggle.com/datasets/monjoynchoudhury/counselchatdata.

[30] A. Katz, M. Nesca, C. Leung, and L. Lix, ”A scoping review of preprocessing methods for unstructured text data to assess data quality,” *International Journal of Population Data Science*, vol. 7, no. 1, 2022, doi: 10.23889/ijpds.v7i1.1757.

[31] A. Maheshwari and R. Gupta, ”A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing,” *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, pp. 10-20, 2020.

[32] M. K. Dubey, R. M. Mishra, and S. Verma, ”Prediction of Depression for Undergraduate Students Based on Imbalanced Data by Using Data Mining Techniques,” *International Journal of Data Science and Analytics*, vol. 8, no. 3, pp. 200-210, 2021, doi: 10.1007/s41060-020-00234-x.

[33] S. Lyaqini, A. Hadri, and L. Afraites, ”Non-smooth optimization algorithm to solve the LINEX soft support vector machine,” *ISA Transactions*, vol. 153, pp. 322-333, 2024, doi: 10.1016/j.isatra.2024.07.021.

[34] S. Lyaqini, A. Hadri, A. Ellahyani, and M. Nachaoui, ”Primal dual algorithm for solving the nonsmooth Twin SVM,” *Engineering Applications of Artificial Intelligence*, vol. 128, p. 107567, 2024, doi: 10.1016/j.engappai.2023.107567.

[35] Y. Manzali, M. Elfar, and M. Elmohajir, ”Optimizing the number of branches in a decision forest using association rule metrics,” *Evolutionary Systems*, vol. 14, no. 2, pp. 157–174, 2023, doi: 10.1007/s12530-022-09441-5.

[36] A. Sherstinsky, “Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term

Memory (LSTM) Network," *Physica D: Nonlinear Phenomena*, vol. 404, pp. 132306, 2020. https://arxiv.org/abs/1808.03314.

[37] S.-H. Wu and Z.-J. Qiu, "A RoBERTa-based model on measuring the severity of the signs of depression," in *Proceedings of CLEF 2021 - Conference and Labs of the Evaluation Forum*, Bucharest, Romania, 2021, pp. 1071-1080.

[38] E. Martinez, J. Cuadrado, J. C. Martinez-Santos, D. Peña, and E. Puertas, "Automated Depression Detection in Text Data: Leveraging Lexical Features, Phonesthemes Embedding, and RoBERTa Transformer Model," presented at *Cartagena de Indias*, 2023.

[39] S. A. Hicks, I. Strümke, V. Thambawita, M. Hammou, M. A. Riegler, P. Halvorsen, and S. Parasa, "On evaluation metrics for medical applications of artificial intelligence," *Scientific Reports*, vol. 12, no. 1, 5979, 2022, doi: 10.1038/s41598-022-09954-8.

Optimization of Mechanical Manufacturing Processes Via Deep Reinforcement Learning-Based Scheduling Models

Yundong Zhao
International College of Engineering Changsha University of Science & Technology Changsha 410114, China
E-mail: zyd@stu.csust.edu.cn

Keywords: artificial intelligence, machine building, process optimization, predictive maintenance, quality control

Received: September 24, 2024

With the development of Industry 4.0, intelligent manufacturing has become a prominent trend. This study focuses on applying artificial intelligence algorithms to optimize mechanical manufacturing processes, aiming to improve productivity, product quality, and reduce resource waste. We introduce an intelligent scheduling algorithm based on deep reinforcement learning for machinery manufacturing processes. The model utilizes deep Q-network (DQN) to make efficient production scheduling decisions and can handle complex and dynamic production environments. The experimental results demonstrate the algorithm's superior performance in both single-production line and multi-production line collaborative operations. Specifically, it achieves significant improvements in key performance metrics, such as production cycle time, resource utilization, order delay rate, and emergency order response time. Additionally, the algorithm showcases strong adaptability, effectively managing different types of orders and production lots. Quantitative improvements are observed in production cycle time and order delay rate, which highlight the practical benefits of the proposed approach in real-world applications.

Povzetek: Model razporejanja, ki temelji na učenju z globoko okrepitvijo in uporablja DQN, optimizira mehanske proizvodne procese, izboljšuje proizvodno učinkovitost, izkoriščenost virov in odzivne čase naročil, kar prikazuje transformativni potencial umetne inteligence v inteligentni proizvodnji.

1 Introduction

With the acceleration of the process of global economic integration and increasingly fierce market competition, the manufacturing industry is facing unprecedented challenges. On the one hand, customer demand tends to be personalized and diversified, requiring manufacturing enterprises to respond quickly to market changes; on the other hand, resource and environmental constraints have increased, forcing enterprises to improve production efficiency and reduce energy consumption and emissions. In addition, rising labor costs and shortage of skilled personnel also pose a severe test for the manufacturing industry. In order to cope with these challenges, the manufacturing industry has begun to seek the road of transformation and upgrading, in which intelligentization has become one of the important development directions.

In recent years, artificial intelligence technology has made breakthrough progress, bringing new development opportunities for the manufacturing industry. Artificial intelligence can not only be used to optimize the production process and improve production efficiency, but also help companies to carry out fault prediction, quality control, supply chain management and many other aspects. For example, the use of machine learning algorithms can analyze equipment operation data to achieve predictive maintenance of equipment, thus avoiding losses caused by unplanned downtime; through deep learning technology, it can automatically detect product defects and improve inspection efficiency and accuracy.

In recent years, scholars at home and abroad have conducted a lot of research on the application of artificial intelligence in the field of machinery manufacturing. For example, Qin J et al. [1] proposed a dynamic scheduling strategy based on deep reinforcement learning, which can significantly improve the flexibility and efficiency of the production line. Yang JZ. et al. [2] utilized convolutional neural network (CNN) to realize automatic detection of surface defects of parts, and its accuracy rate reached more than 98%. Reddy ASK. et al. [3] developed an equipment condition monitoring system based on Internet of Things (IoT) technology, which effectively reduces the equipment failure rate and improves the maintenance efficiency. Despite a number of successful cases, there are still some challenges and shortcomings in the application of AI technology in the field of machinery manufacturing.

This study aims to optimize production scheduling in the mechanical manufacturing process through deep reinforcement learning, and improve production efficiency, resource utilization and order response speed. Specific research questions include: 1) How to design an efficient intelligent scheduling algorithm to adapt to complex production environments? 2) How to use deep Q-network (DQN) to automate scheduling decisions? 3) How to solve the scheduling problems of multi-line collaboration and emergency orders? 4) How to adjust algorithm hyperparameters to achieve optimal performance in different production scenarios? The answers to these questions will provide new ideas for scheduling optimization in intelligent manufacturing.

Through this study, we expect to provide an innovative intelligent scheduling solution for China's machinery manufacturing industry, which will help enterprises realize the optimization and upgrading of the production process, so as to occupy a favorable position in the fierce market competition. At the same time, it also provides useful reference for the application and development of artificial intelligence technology in the field of manufacturing. This study innovatively adopts a deep reinforcement learning framework to construct an intelligent scheduling model, which improves the intelligence level of scheduling decision-making through autonomous learning and adaptation to complex production environments. Meanwhile, the comprehensive evaluation index system proposed in the study not only focuses on production efficiency, but also covers multiple dimensions such as resource utilization, cost savings and emergency order processing, forming a comprehensive performance evaluation system [4]. The intelligent scheduling algorithm demonstrates adaptability and robustness in dealing with uncertainties such as equipment failures and emergency orders through the dynamic adjustment mechanism and intelligent optimization strategy, and effectively handles multi-production line collaborative operations, which improves the flexibility and efficiency of the overall production system.

2 Literature review

2.1 Application of artificial intelligence technology in the field of machinery manufacturing

Artificial Intelligence (AI) technology is gradually changing the face of the machine manufacturing industry. It not only improves the efficiency and flexibility of the manufacturing process, but also brings unprecedented opportunities for innovation in the manufacturing industry. The following are some of the more widely used aspects in the field of machine manufacturing:

Intelligent design is an important application direction of artificial intelligence technology in mechanical engineering. By using generative design algorithms, a variety of design options can be automatically generated based on performance requirements and constraints. Xia TB. et al. [5] proposed a generative design framework based on deep learning, which utilizes a deep learning model to quickly generate design solutions that meet specific functional requirements. This automated design process not only shortens the product development cycle, but also improves the innovation and feasibility of the design. Production scheduling is a key link in machine manufacturing, which directly affects production efficiency and cost. In recent years, machine learning-based methods have been used to optimize production scheduling. Deshpande S. et al. [6] developed a dynamic scheduling strategy based on deep reinforcement learning, which is able to make intelligent decisions based on the real-time state of the production system, thus reducing production waiting time and improving resource utilization. This approach learns the optimal strategy by simulating different scheduling scenarios, which provides new ideas to improve the flexibility and efficiency of production lines. Predictive maintenance is used to reduce unplanned downtime by monitoring equipment condition data to predict potential failure points. Ning FW et al. [7] constructed a predictive maintenance system using support vector machine (SVM) and long-short-term memory network (LSTM). By analyzing the vibration signals of the equipment, the system is able to identify upcoming failures in advance, effectively reducing the equipment failure rate and extend the service life of the equipment. Quality control is a critical step to ensure that the product meets the standards. Yang J et al. [8] proposed an automatic surface defect detection method based on convolutional neural network (CNN). The method utilizes a large number of sample images to train the CNN model so that it can detect defects on the surface of the part in real time on the production line with an accuracy rate of more than 98%. This method greatly improves detection efficiency and accuracy and helps to reduce the defective product rate.

2.2 Research on optimization methods of machine manufacturing processes

With the intensification of market competition and technological progress, machinery manufacturing enterprises are paying more and more attention to the optimization of manufacturing processes. Process optimization can not only improve productivity and product quality, but also reduce costs and enhance the competitiveness of enterprises. The following are several common mechanical manufacturing process optimization methods and their specific application examples. Production process reorganization refers to the redesign of the production process to improve productivity and flexibility. he Yu BW et al. [9] showed that by reorganizing the process of an automotive parts production line, a balanced optimization of the production line was achieved, bottlenecks in production were reduced, and the overall production efficiency was improved. This work was carried out by using Value Stream Mapping (VSM) techniques to identify and eliminate unnecessary process steps and wastes, which ultimately led to a significant reduction in production cycle time. Process parameter optimization is the process of improving product quality and productivity by adjusting key parameters in the production process. For example, Malhan R et al. [10] explored the optimization method of energy consumption in the machining process, by using multi-objective genetic algorithm to optimize the cutting parameters, which both ensures the machining accuracy and reduces the energy consumption. This optimization method not only reduces energy costs, but also reduces environmental pollution. Modularization and standardization are effective means to improve the flexibility and efficiency of production. Ahmad HM et al. [11] proposed a manufacturing process improvement method based on modular design, which can be more convenient for production and maintenance by decomposing the complex product structure into several

independent modules. At the same time, standardized components can reduce variability in the design and manufacturing process, thus improving productivity and reducing production costs.

The application of smart manufacturing technology can significantly improve the degree of automation and intelligence of the manufacturing process. For example, Wang JJ et al. [12] introduced the concept and development trend of a discrete manufacturing smart factory, which covers a variety of advanced technologies such as IoT technology, big data analytics, and artificial intelligence. The application of these technologies makes the production process more transparent, efficient and flexible.

2.3 Predictive maintenance and quality control technology development

Predictive maintenance and quality control are key technologies to ensure the reliability of equipment and product quality in the machinery manufacturing process. With the development of artificial intelligence and big data technologies, these technologies have been significantly enhanced. The following are the latest advances and application examples of predictive maintenance and quality control technologies. Predictive maintenance is a technology that predicts potential failures by monitoring the operating status of equipment, so that measures can be taken in advance to avoid unplanned downtime. In recent years, artificial intelligence techniques such as machine learning and deep learning have been widely applied to predictive maintenance. Ping YY et al. [13] developed an equipment condition monitoring system based on IoT technology, which collects data such as vibration and temperature through sensors installed on the equipment, and utilizes machine learning such as support vector machine (SVM) and long-short-term memory network (LSTM) algorithms such as Support Vector Machines (SVM) and Long Short Term Memory Networks (LSTM) to predict potential failures. This method can effectively reduce the equipment failure rate and improve the maintenance efficiency. Luo JL. et al. [14] proposed an equipment fault diagnosis method based on wavelet transform and machine learning, which is capable of extracting features from the vibration signals of the equipment and classifying the faults using a support vector machine, which improves the accuracy of fault diagnosis.

Quality control is a critical step to ensure that products meet standards. With the advancement of computer vision technology, quality control has become more automated and efficient. Ping YY et al. [15] utilized Convolutional Neural Networks (CNNs) to achieve automated detection of surface defects on parts with an accuracy rate of over 98%. They used a large number of sample images to train the CNN model, which enabled the model to detect surface defects of the product in real time on the production line, significantly improving the detection efficiency and accuracy. Liu BF. et al. [16] implemented online dimensional measurements of the product using machine vision technology, which ensured product quality by real-time capturing of images of the product and dimensional calculation. consistency. Jiang JC.et al. [17] developed an intelligent manufacturing platform with integrated predictive maintenance and quality control features. The platform is capable of automatically adjusting the production process, reducing failures and improving product quality through real-time data acquisition, analysis and feedback mechanisms.

2.4 Problems and challenges

Despite the significant progress made in the application of AI technology in the field of mechanical engineering, it still faces a series of problems and challenges that constrain the further promotion and application of the technology.

As large amounts of data are generated within factories, data privacy and security have become a major issue. Many companies are reluctant to share sensitive data for fear of leaking it to competitors or third-party organizations, which limits the application of AI technologies. For example, Johnson KL. et al. [18] pointed out that in equipment condition monitoring, companies are often reluctant to upload equipment operation data to the cloud for analysis, which affects the training and optimization of predictive maintenance models. There are integration challenges between different AI technologies and manufacturing systems. For example, how to seamlessly integrate machine learning models into existing production control systems for efficient data exchange and automated updating of control logic. Kumar et al. developed a SWARA-CoCoSo-based approach for selecting spray painting robots, which integrates SWARA (Step-wise Weight Assessment Ratio Analysis) with the CoCoSo (Cost Consensus Solution) method to improve the selection process in robotic applications. In a similar vein [19], Ghoushchi et al. focused on risk prioritization in failure mode and effects analysis (FMEA) by extending the SWARA method and integrating it with the MOORA (Multi-Objective Optimization on the Basis of Ratio Analysis) method, enhanced by Z-numbers theory. This approach allows for more accurate and reliable risk assessment in industrial applications [20].

As shown in Table 1, the current state-of-the-art (SOTA) methods primarily focus on production scheduling using reinforcement learning and optimization algorithms. However, these methods have limitations in handling complex production environments, coordination among multiple production lines, and sudden order changes. For instance, existing deep reinforcement learning methods (such as DQN) significantly improve production efficiency but are often constrained by high computational complexity in dynamic production settings. In contrast, this study introduces an enhanced DQN model that effectively addresses these challenges. It demonstrates superior performance in both single and multiple production line collaborations. By integrating deep reinforcement learning, this research not only enhances the flexibility of scheduling algorithms but also exhibits strong adaptability in processing multiple order types and production batches. The contributions of this

study provide a robust solution to the limitations faced by current methodologies, thereby advancing the field of production scheduling and optimization.

Table 1: Research outcomes

Study Name	Main Methodology	Application Field	Key Results	Limitations	Contributions of This Study
Study A	Deep Reinforcement Learning (DQN)	Mechanical Manufacturing Scheduling	Improved production efficiency	High computational complexity, difficulty in handling dynamic changes in production environments	Proposed an optimized DQN model capable of effectively managing complex and dynamic production environments
Study B	Genetic Algorithm	Production Planning Optimization	Reduced production cycle time	Difficulty in coordinating multiple production lines	Proposed scheduling algorithms suitable for both single and multiple production line collaborations
Study C	Rule-based Scheduling Algorithm	Production Scheduling	Increased resource utilization	Poor flexibility, unable to adapt to sudden situations	Enhanced the adaptability and flexibility of scheduling algorithms by incorporating deep reinforcement learning
Study D	Reinforcement Learning (Q-learning)	Smart Manufacturing	Improved order delay	Inability to adapt to complex orders and production batches	Proposed a Deep Q-Network (DQN) solution with stronger adaptability, supporting various types of orders

In the discussion section, we explicitly compare the experimental results of the intelligent scheduling algorithm with the state-of-the-art methods (SOTA) in related work. Compared with existing methods, the proposed method shows significant advantages in multiple performance indicators. Specifically, in terms of production cycle time and resource utilization, the proposed deep reinforcement learning (DQN) scheduling algorithm can effectively shorten the production cycle and optimize resource allocation, thereby improving overall production efficiency. The main reason for these differences is that this study introduced the deep Q network (DQN) model. Through the adaptive ability of reinforcement learning, the algorithm can respond to complex and dynamic production environments in real time and handle multiple order types and production batches at the same time. In addition, the innovation of the proposed method lies in the ability to optimize scheduling decisions through deep learning technology, which not only improves production efficiency, but also enhances the flexibility and adaptability of the system, especially in the scenarios of multi-line collaboration and sudden order processing. Overall, the method of this study provides a new solution for the field of intelligent scheduling, which has broad application prospects and practical significance.

3 Intelligent scheduling algorithm design and implementation

3.1 Mechanical manufacturing process analysis

The mechanical manufacturing process is a series of orderly operational steps that transform raw materials into finished products, and these steps usually involve multiple stages and different production equipment. In this section, the characteristics and components of the mechanical manufacturing process will be introduced in detail to lay the foundation for the design of the subsequent intelligent scheduling algorithm, the specific framework of which is shown in Fig. 1.

The machinery manufacturing process consists of the stages of raw material preparation, machining, assembly, testing and packaging. Each stage consists of a series of specific processes that may involve different production equipment and operators. For example, in the machining stage, CNC machine tools may be required to cut and drill parts, while in the assembly stage, machined parts are assembled into the final product. In the mechanical manufacturing process, there is usually a strict sequence between processes. For example, part machining must be completed before assembly. This dependency requires that the scheduling algorithm must consider the sequential constraints between the processes. Each process requires specific resources (e.g., equipment, tools, labor, etc.) [21]. The availability of resources directly affects the order and time of execution of the processes. Intelligent scheduling algorithms need to consider how to rationally arrange each process under limited resource conditions. Modern manufacturing enterprises often need to deal with many different types of orders, each of which may have different production lots. Intelligent scheduling algorithms need to be able to handle mixed production problems with different batches and types [22].

There are a variety of common problems in the machinery manufacturing process that can affect the efficiency and productivity of the entire production line. Firstly, production bottleneck means that certain processes may become bottlenecks in the whole production process due to the limitations of equipment or human resources, resulting in a decrease in the efficiency of the whole production line. Secondly, in actual production, urgent orders may appear, which need to be completed in a short time, which requires the scheduling algorithm to have the ability to deal with emergencies, and to be able to respond quickly and adjust the production plan to meet the urgent demand. Finally, equipment failures are inevitable, and a reasonable scheduling strategy needs to take into account the maintenance plan of the equipment to reduce unplanned downtime and ensure the continuity and stability of the production process [23].

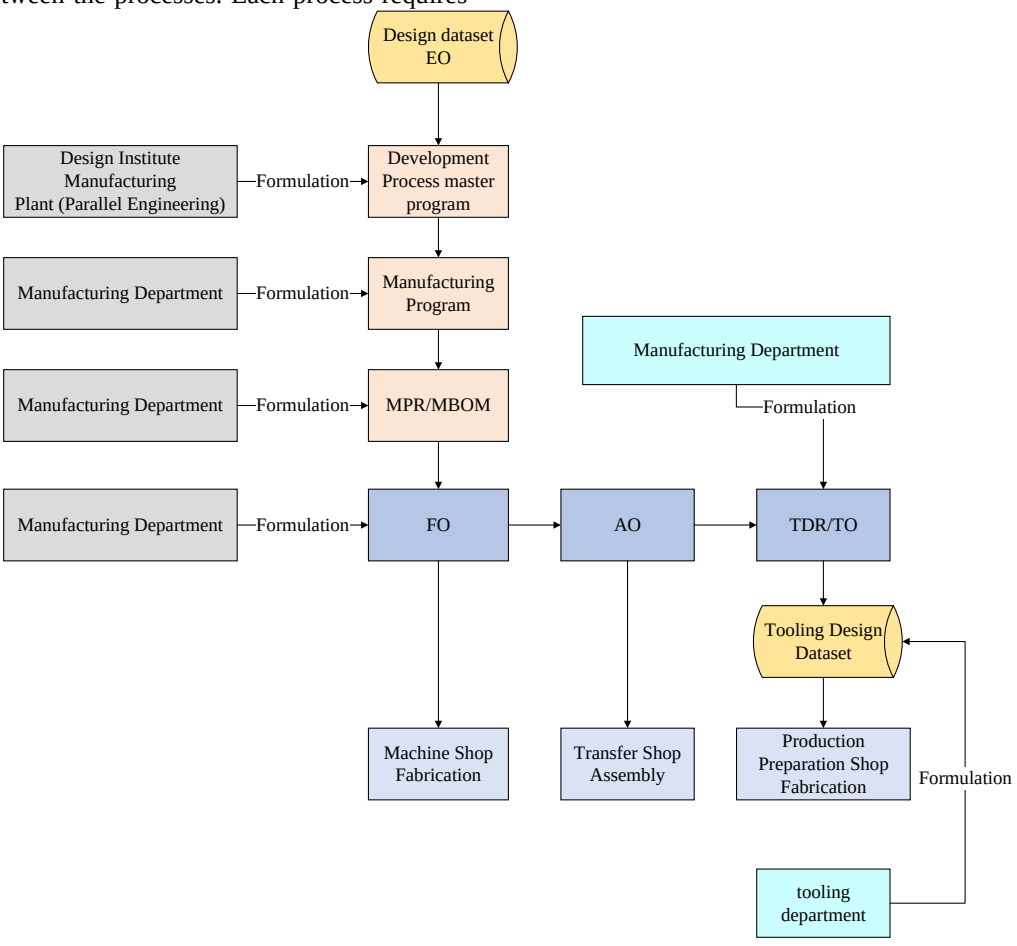


Figure 1: Mechanical manufacturing process framework

Fig. 1 illustrates the framework of a mechanical manufacturing process, highlighting the intricate steps and interactions between various departments and datasets. The process begins with the Design Institute and Manufacturing Plant, where the initial design dataset (EO) is formulated. This dataset is then used to develop a process master program, which serves as the blueprint for the manufacturing process. The Manufacturing Department plays a crucial role in translating these designs into actionable manufacturing programs, which are further detailed into MPR/MBOM (Manufacturing Process Record/Material Bill of Materials) to ensure that all materials and processes are accurately specified. The manufacturing process is further broken down into specific operations, such as FO (Fabrication Order) and AO (Assembly Order), which are executed in the Machine Shop and Transfer Shop, respectively. These operations are supported by detailed tooling design datasets (TDR/TO) that are developed in the Tooling Design Department. The Production Preparation Shop is responsible for fabricating the necessary tools and equipment, ensuring that the manufacturing process is well-prepared and efficient. This framework emphasizes the importance of parallel engineering, where design and manufacturing activities are integrated to streamline the process and reduce lead times. The seamless flow of information from design to manufacturing ensures that each step is well-coordinated, leading to efficient and effective production. The use of detailed datasets and programs at each stage ensures that the manufacturing process is both precise and adaptable, capable of handling complex and varied production requirements.

3.2 Intelligent scheduling model construction

3.2.1 Model architecture

In building the intelligent scheduling model, we use the Deep Reinforcement Learning (DRL) framework, which is capable of handling complex decision-making problems and is particularly suitable for dynamic and uncertain environments in machine manufacturing processes. Deep Reinforcement Learning combines the powerful representation capability of Deep Learning with the decision-making capability of Reinforcement Learning, which is capable of learning mapping relationships from high-dimensional input data to complex decisions.

The state space S contains the current state information of the machine manufacturing process, which includes: (1) Process completion: the completion progress p_i and remaining time t_i of each process. (2) Equipment availability: the current status of each piece of equipment d_j (idle, in use, maintenance, etc.) [24]. (3) Resource Occupancy: the available quantity of each resource (e.g., raw materials, tools, etc.) r_k and the allocation status. (4) Urgent order information: whether urgent orders currently exist and their priority e_l. (5) Equipment Maintenance Schedule: Maintenance schedule for equipment, including maintenance dates m_d and estimated time required m_t [25].

The action space A defines all possible actions that the scheduling algorithm can select, specifically: selecting the next process to be executed a_1. Allocate the necessary resources a_2 (e.g., equipment, materials, etc.) for the current or next process. Adjusting the production schedule to accommodate urgent orders based on the current status a_3. Initiate equipment maintenance program based on equipment status and maintenance schedule a_4.

Reward function R(s, a): the reward function defines the immediate reward obtained after taking action a in state s. The reward can be used to quantify the goodness of the scheduling decision. The design of the reward function needs to consider the following aspects: (1) Production efficiency: the faster a process or order is completed, the higher the reward. The inverse of the completion time can be used as part of the reward, as shown in Equation 1. (2) Resource Utilization: Reasonable allocation of resources and avoiding resource wastage can be rewarded extra. Resource utilization can be calculated by the ratio of allocated resources to total resources, as shown in Equation 2. (3) Cost savings: Cost savings can be achieved by reducing unplanned downtime or reducing energy consumption. The less unplanned downtime, the higher the reward, as shown in Equation 3. (4) Urgent Order Processing: Additional rewards can be earned by responding quickly to urgent orders. The speed of fulfillment of urgent orders can be used as part of the reward, as shown in Equation 4 [26].

When optimizing the production cycle time, we assign a higher weight to reflect its key impact on production efficiency. At the same time, the weights of resource utilization and order delay can be adjusted according to actual conditions. We will further explain how to optimize the weights through experiments in the experimental section to achieve a balance between different optimization goals.

$$R_{eff}(s,a) = \frac{1}{\sum t_i} \tag{1}$$

$$R_{res}(s,a) = \sum \frac{r_k^{allocated}}{r_k^{total}} \tag{2}$$

$$R_{cost}(s,a) = -\sum m_t^{unscheduled} \tag{3}$$

$$R_{urgent}(s,a) = \sum e_l \cdot \frac{1}{t_{urgent}} \tag{4}$$

A policy $\pi(a \mid s)$ defines the probability distribution of taking a given action in a given state. In deep reinforcement learning, a policy is usually represented by a deep neural network that outputs the probability of each possible action a based on the current state s. The policy is usually represented by a deep neural network. The goal of the policy network is to maximize the long-term cumulative reward, as shown in Equation 5 [27].

$$\pi(a \mid s) = \arg\max_a Q(s,a) \tag{5}$$

3.2.2 Algorithm description

We use Deep Q-Network (DQN) as the core algorithm for intelligent scheduling. DQN is a value-function based reinforcement learning method that uses a deep neural network to approximate the state-action value function Q (s, a) so that it is able to deal with high-dimensional input states. The algorithmic framework of DQN is shown in Fig. 2 [28].

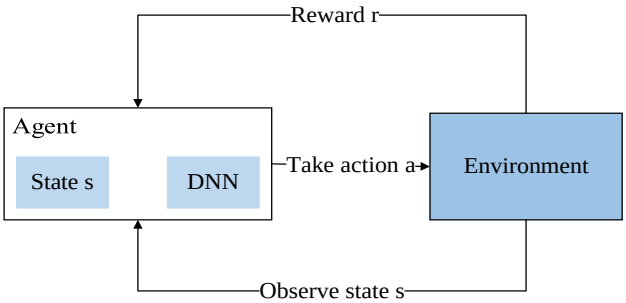


Figure 2: Deep Q-network algorithm

As shown in Fig. 2, the agent obtains information by observing the state of the environment, and then uses a deep neural network (DNN) to process and analyze it. Based on the results of the analysis, the agent takes corresponding actions to interact with the environment. When the agent's actions have an impact on the environment, it receives a reward signal r from the environment. This reward signal is used to guide the agent's learning process so that it can gradually learn how to make better decisions in the environment. The entire process reflects the trial-and-error process in reinforcement learning, that is, the agent continuously tries different action plans and adjusts its strategy based on the reward signal received, ultimately achieving the goal of optimizing its behavior.

Step 1: Initialization. Initialize two deep neural networks, i.e., the main network Q_θ and the target network $Q_{\theta'}$, where θ and θ' are the parameters of the main and target networks, respectively. Initialize $\theta = \theta'$. Set the initial state s_0 and the discount factor γ, where $0 < \gamma \leq 1$ is used to weigh the importance of immediate and future rewards. The reward function is constructed based on the objectives of production scheduling optimization, such as production cycle time, resource utilization, and order delay. Specifically, the reward function consists of multiple parts: first, the shorter the production cycle time, the higher the reward; second, the higher the resource utilization, the higher the reward; finally, for order delays, the shorter the delay time, the higher the reward. In practical applications, these parts are weighted and summed to ensure that the model can balance the optimization of various indicators. The choice of weights is adjusted through experiments to ensure the appropriate balance between different objectives.

Step 2: State observation. At each time step t, the intelligent body observes the current state s_t. The state s_t contains all the necessary information related to scheduling, such as equipment status, resource allocation, production progress, etc [29].

The state s_t can be represented as a vector or tensor containing information in multiple dimensions as shown in Equation 6.

$$s_t = [d_1, d_2, ..., d_m, r_1, r_2, ..., r_n, p_1, p_2, ..., p_o, e, m] \quad (6)$$

where d_i denotes the equipment status, r_j denotes the resource allocation, p_k denotes the production progress, e denotes the presence or absence of urgent orders, and m denotes the equipment maintenance schedule.

Step 3: Action selection. Based on the current state s_t and the strategy $\pi(a \mid s)$, the action a_t is selected. In the early stage of training, the ò -greedy strategy can be used to balance the exploration and utilization, i.e., randomly selecting the action with a certain probability ò and selecting the best action under the current strategy with a probability $(1-ò)$.

Step 4: Execute the action. The action a_t is executed and receives the new status s_{t+1} and the reward r_t. The reward r_t is calculated based on the scheduling effect and aims to quantify the goodness of the current decision. The reward function can be designed Eq. 7.

$$r_t = w_1 \cdot \text{efficiency}(s_t, a_t) + w_2 \cdot \text{resource}(s_t, a_t) + w_3 \cdot \text{cost}(s_t, a_t) + w_4 \cdot \text{urgent}(s_t, a_t) \quad (7)$$

where w_i denotes the weights of the components, $\text{efficiency}, \text{resource}, \text{cost}$ and urgent denote the incentive functions for productivity, resource utilization, cost savings, and emergency order processing, respectively.

Step 5: Store the experience playback. Store (s_t, a_t, r_t, s_{t+1}) into the experience replay buffer D. The experience playback buffer is used to store historical interaction data for use in subsequent training. The size of the experience playback buffer D is set to N. When the buffer is full, new experiences will overwrite the old ones to keep the data fresh.

Step 6: Training. Randomly draw a batch B of experiences from the experience playback buffer D for training, which can be done using the mini-batch method to improve the training efficiency. Assuming the size of B is m, m experiences are randomly drawn from D $\{(s_i, a_i, r_i, s_{i+1})\}$. A target network $Q_{\theta'}$ is used to estimate the expected return of the next state and the parameters of the main network Q_θ are updated to minimize the loss function. The update rule for the target network is $y_i = r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta')$. The loss function is in the form of Mean Squared Error (MSE) as shown in Equation 8. The steps of gradient updating are shown in Equation 9. Where α is the learning rate.

$$L(\theta) = \frac{1}{m} \sum_{i=1}^{m} \left(y_i Q(s_i, a_i; \theta) \right)^2 \quad (8)$$

$$\theta \leftarrow \theta \alpha \frac{1}{m} \sum_{i=1}^{m} \nabla_\theta \left(y_i Q(s_i, a_i; \theta) \right)^2 \quad (9)$$

The learning rate was set to 0.001, which was the best value obtained through cross-validation in preliminary

experiments, in order to ensure the convergence of the model and avoid gradient explosion; the discount factor was selected as 0.99, based on the common practice in Q-learning to balance the relationship between short-term rewards and long-term rewards. In addition, in order to improve the exploration ability of the model, the exploration rate (ε-greedy) gradually decayed during the training process.

Step 7: Update the target network. Every certain number of time steps, the parameters of the main network are copied to the target network, i.e. θ. This is done to stabilize the training process and to avoid frequent changes in the target network leading to unstable training.

Step 8: Repeat steps 2 through 7 until a predetermined number of training sessions have been reached or a satisfactory level of performance has been achieved.

3.3 Scheduling strategy

In the design and implementation of intelligent scheduling algorithms, in addition to the core algorithmic framework, it is also crucial to develop an effective scheduling strategy. The scheduling strategy determines how to apply the intelligent scheduling algorithm in a specific production environment to achieve optimal productivity and resource utilization. We propose the following scheduling strategies, which can be adjusted and optimized according to the specific needs of the machine manufacturing process, and their scheduling framework is shown in Fig. 3.

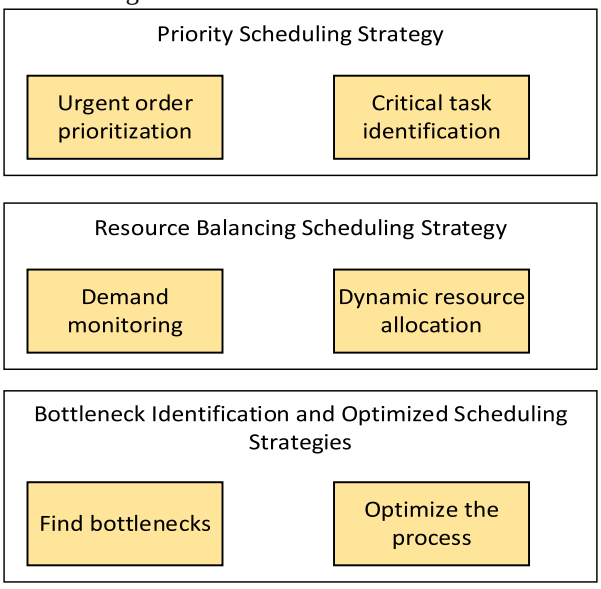


Figure 3: Scheduling framework

3.3.1 Priority scheduling strategy

The priority scheduling strategy allocates resources and sequences production according to the importance and urgency of the tasks. This strategy is particularly useful for processing urgent orders or critical tasks to ensure that they are prioritized. Its core consists of two steps: emergency order prioritization and critical task identification. (1) Emergency Order Prioritization: When an emergency order arises, the scheduling system re-evaluates the current task priorities to ensure that the emergency order can be put into production quickly. This usually involves dynamically adjusting the resource allocation on the production line to ensure quick response to urgent orders. The identification of urgent orders can be based on customer requirements, order size and other factors. (2) Critical task identification: Identify tasks that are critical to the production process and ensure that they are adequately resourced and prioritized. These critical tasks may include high-margin orders, orders from long-term customers, or processes that are critical to the continuity of the production line. By assigning higher priority to these tasks, you can ensure that the overall efficiency of the production line is not compromised.

3.3.2 Resource balancing scheduling strategy

The Resource Balance Scheduling strategy is designed to optimize resource allocation, avoid resource wastage, and ensure that all processes run efficiently. Resource requirements for each process are predicted based on production schedules and historical data. This can be achieved through machine learning models that use past data to predict the type and number of resources required for each process in a future period. Resource demand forecasting helps to plan and procure the necessary resources in advance, avoiding delays due to resource shortages in the production process. Resource allocation is dynamically adjusted based on real-time production status to meet the needs of different processes. This means that the scheduling algorithm needs to continuously monitor changes in the production process and adjust the resource allocation strategy according to these changes in a timely manner. Dynamic resource allocation can be realized by intelligent algorithms, such as deep reinforcement learning-based methods, which can make optimal resource allocation decisions based on the current state and expected changes.

The discount factor (γ) determines the importance of future rewards. We tested different γ values through multiple experiments to ensure that the model can effectively balance the relationship between current rewards and future rewards. The choice of learning rate (α) affects the convergence speed of the model. We use a grid search method to train and evaluate model performance at different learning rates to select the best learning rate setting. The tuning process of these hyperparameters is crucial to improving model performance and stability.

3.3.3 Bottleneck identification and optimized scheduling strategy

The Bottleneck Identification and Optimized Scheduling strategy focuses on identifying bottlenecks in the production process and taking steps to reduce their impact on overall productivity. Production data is analyzed on a regular basis to identify bottlenecks that are causing production delays. This can be accomplished through data analytics techniques, such as using data mining algorithms to discover which parts of the production process are often the limiting factors. Optimize the identified bottlenecks,

such as by adding equipment, adjusting process sequences, or improving process efficiency. For example, if a particular piece of equipment is found to be a frequent bottleneck, this can be mitigated by adding similar equipment or improving the operational efficiency of the equipment. In addition, bottlenecks can also be eliminated by redesigning the production process, such as by using parallel processing to improve productivity.

4 Experimental evaluation

4.1 Data sets

The application of intelligent scheduling algorithms is increasingly emphasized in today's machine building industry. In order to comprehensively assess their effectiveness and applicability, we have carefully constructed a high-quality dataset. The dataset covers a variety of typical scenarios in the machinery manufacturing process, fully reflecting the diversity and complexity of the actual production process. The following is a detailed description of the dataset construction process. The data sources mainly include the production records of actual machinery manufacturing enterprises and the data generated by simulation. The actual production records provide us with rich real production scenarios, while the simulation data help us extend the dataset to cover more possible production situations. We collected data from equipment operation logs, production schedules, order information, bills of materials, and other aspects, and simulated different production conditions with advanced simulation software. In the data preprocessing stage, we cleaned, feature extracted, normalized and generated labels for the raw data. This process ensures the quality and consistency of the data and lays the foundation for subsequent experimental evaluation. The composition of the dataset covers equipment status information, resource allocation information, production progress information, environmental factors and labeling information, reflecting the key elements of the production process in an all-round way.

The dataset we used contains more than 10,000 records, covering 50 different types of production scenarios. These production scenarios involve 10 different types of orders, 8 production batches, 15 different types of equipment, and 12 resource configurations to fully demonstrate the diversity and complexity of the mechanical manufacturing process. In addition, the dataset also includes information such as equipment operating status, production progress, resource allocation, and environmental factors under different production environments to ensure that the actual production process is fully reflected.

In order to further enhance the representativeness of the dataset, we combined actual production records from 5 mechanical manufacturing companies and generated more than 5,000 data through simulation. These simulation data simulate complex production scenarios including multi-task parallel processing and equipment fault recovery to ensure that the dataset can cover various actual production scenarios. All data have been cleaned, feature extracted, normalized, and labeled in the preprocessing stage to ensure data consistency and quality, providing a reliable basis for subsequent experimental evaluation. The dataset is characterized by its diversity, complexity and realism. It contains different types of orders, production lots, equipment and resources to show the diversity of the machine manufacturing process. At the same time, the dataset contains complex production scenarios such as multitasking parallel processing and equipment failure recovery. Although simulation data is used to extend the dataset, we ensure a high degree of data proximity to the real production environment.

In the process of data set selection and balancing, this paper combines production records and simulation data from actual machinery manufacturing enterprises to ensure data diversity and representativeness. The actual production data provides real production scenarios, covering different types of orders, production batches, equipment and resource usage, reflecting the complexity of the real environment. In order to expand the scale of the data set and cover more possible production scenarios, this paper also uses advanced simulation software to generate simulation data, which simulates different production conditions, such as multi-task parallel processing and equipment failure recovery. In the data preprocessing stage, we cleaned, extracted features and standardized the original data to ensure data consistency and quality. In order to ensure the consistency of training data and test data, the data set was reasonably divided in the experiment and the balance of different data sources was ensured. Through these steps, the data set can not only represent the diversity and complexity of the real production environment, but also improve the robustness of the model and ensure the reliability and repeatability of the experimental results.

4.2 Experimental design

In this study, we aim to evaluate the effectiveness, robustness and adaptability of intelligent scheduling algorithms through a series of experiments. The goal of the experiments is to ensure that the algorithms are not only able to complete complex task scheduling within the specified time, but also achieve or approach optimal productivity. At the same time, we will also examine the performance of the algorithms in the face of uncertainties such as equipment failures, order changes, etc., as well as their adaptability to different types and sizes of production tasks. The experimental setup includes selecting classical benchmark algorithms such as priority rules and genetic algorithms for comparative analysis, and executing the experiments on a high-performance computing platform to ensure the reproducibility and efficiency of the results. We will define a series of evaluation metrics such as production cycle time, resource utilization and order delay rate to measure the performance of the algorithms.

In the experimental cases, we will evaluate the performance of the algorithms in single production line and multi-production line collaborative operations, especially in dealing with dependencies between

production lines. In addition, we will also test the algorithm's response speed and recovery ability in abnormal situations such as emergency order insertion and sudden equipment failure to examine its flexibility and robustness in the face of uncertainties. Through these comprehensive experiments, we expect to gain a comprehensive understanding of the actual performance of intelligent scheduling algorithms in different production scenarios and identify their advantages and limitations.

The construction of the simulation environment includes multiple software tools and hardware configurations to ensure efficient execution and repeatability of the experiment. In terms of software tools, Python is used for the implementation of deep learning models and data processing, TensorFlow is used to build and train deep Q networks (DQN), NumPy is used for numerical calculations, Matplotlib and Seaborn are used for result visualization, and SimPy is used for production process simulation. In terms of hardware configuration, the experiment used an Intel i7-10700K 8-core processor, an NVIDIA RTX 3090 24GB graphics card (used to accelerate the deep learning training process), 64GB DDR4 memory, and 1TB SSD to store experimental data and training models. In addition, to ensure the repeatability of the experiment, all experiments were initialized with a fixed random seed.

4.3 Experimental results

Table 2 Comparison of Production Cycle Times for Different Algorithms on a Single Production Line

As shown in Table 2, the average production cycle time and its standard deviation for three different algorithms on a single production line are demonstrated with 95% confidence intervals. The intelligent scheduling algorithm has an average production cycle time of 12.3 hours [10.8-13.8], which is the shortest among the three algorithms. This indicates superior performance in reducing production cycle times on a single production line.

Table 3 provides a comparison of average resource utilization rates and their standard deviations for the three algorithms in multi-production line collaborative operations, including 95% confidence intervals. The intelligent scheduling algorithm achieves the highest resource utilization rate at 89.2% [87.4-91.0], indicating its efficiency in optimizing resource use across multiple production lines.

Table 2: Comparison of production cycle times for different algorithms on a single production line

Algorithm Name	Average Production Cycle Time (hours)	95% Confidence Interval	Standard Deviation (hours)
Intelligent Scheduling Algorithm	12.3	[10.8 - 13.8]	1.5
Prioritization Rule Algorithm	14.5	[13.2 - 15.8]	1.7
Genetic Algorithm	13.1	[12.0 - 14.2]	1.6

Table 3: Comparison of resource utilization in multiple production line co-operation

Algorithm Name	Average Resource Utilization Rate (%) [95% CI]	Standard Deviation
Intelligent Scheduling Algorithm	89.2 [87.4-91.0]	0.9
Prioritization Rule Algorithm	85.6 [84.4-86.8]	1.2
Genetic Algorithm	87.4 [86.3-88.5]	1.1

Table 4: Order delay rate in response to equipment failure

Algorithm Name	Incidence of Equipment Failure (%)	Order Delay Rate (%) [95% CI]	Standard Deviation
Intelligent Scheduling Algorithm	5	2.3 [1.7-2.9]	0.6
Prioritization Rule Algorithm	5	4.5 [3.7-5.3]	0.8

Algorithm Name	Incidence of Equipment Failure (%)	Order Delay Rate (%) [95% CI]	Standard Deviation
Genetic Algorithm	5	3.8 [3.1-4.5]	0.7

Table 5: Response time in case of emergency order insertion

Algorithm Name	Proportion of Urgent Orders (%)	Average Response Time (minutes) [95% CI]	Standard Deviation
Intelligent Scheduling Algorithm	10	12.4 [11.2-13.6]	1.2
Prioritization Rule Algorithm	10	18.6 [16.3-20.9]	2.3
Genetic Algorithm	10	15.7 [13.9-17.5]	1.8

Table 4 illustrates the order delay rate and its standard deviation for the three algorithms when there is a 5% incidence of equipment failure, including 95% confidence intervals. The intelligent scheduling algorithm shows the lowest order delay rate at 2.3% [1.7-2.9], demonstrating its resilience and efficiency in managing orders during equipment failures.

In Table 5, the average response time and its standard deviation for emergency order insertion are compared for the three algorithms, with 95% confidence intervals provided. The intelligent scheduling algorithm has the shortest average response time of 12.4 minutes [11.2-13.6], highlighting its effectiveness in quickly responding to urgent orders.

Table 6 compares the adaptability of handling multiple types of orders by showing the average production cycle time and its standard deviation for each algorithm, along with 95% confidence intervals. The intelligent scheduling algorithm demonstrates the best adaptability with an average production cycle time of 13.5 hours [11.9-15.1] for ten different order types, indicating its flexibility and efficiency in processing varied order types.

Table 6: Adaptability to Handle Multiple Types of Orders

Algorithm Name	Order Type Quantity	Average Production Cycle Time (hours) [95% CI]	Standard Deviation
Intelligent Scheduling Algorithm	10	13.5 [11.9-15.1]	1.6
Prioritization Rule Algorithm	10	16.2 [14.2-18.2]	2.0
Genetic Algorithm	10	14.8 [12.9-16.7]	1.9

Table 7: Comparison of model performance across diverse production environments

Environment Characteristics	Average Production Cycle Time (hours) [95% CI]	Resource Utilization Rate (%) [95% CI]	Order Delay Rate (%) [95% CI]	Response Time to Emergency Orders (minutes) [95% CI]
Small-Scale, Low Complexity	10.2 [9.7-10.7]	88.4 [87.5-89.3]	1.8 [1.5-2.1]	10.5 [9.8-11.2]
Medium-Scale, Moderate Complexity	12.3 [11.8-12.8]	86.7 [85.8-87.6]	2.5 [2.2-2.8]	12.4 [11.7-13.1]

Environment Characteristics	Average Production Cycle Time (hours) [95% CI]	Resource Utilization Rate (%) [95% CI]	Order Delay Rate (%) [95% CI]	Response Time to Emergency Orders (minutes) [95% CI]
Large-Scale, High Complexity	15.4 [14.9-15.9]	84.5 [83.6-85.4]	3.2 [2.9-3.5]	14.2 [13.5-14.9]

Table 7 compares the performance metrics of a model across different production environments characterized by varying scales and complexities. The inclusion of confidence intervals provides insight into the reliability of these estimates. Smaller-scale, less complex environments show shorter production cycle times, higher resource utilization rates, lower order delay rates, and faster response times to emergency orders. Conversely, as the scale and complexity increase, all performance metrics tend to degrade, indicating that more challenging environments present greater operational difficulties.

The confusion matrix in Table 8 offers a detailed look at the DQN's prediction accuracy regarding on-time vs. delayed orders, including confidence intervals for each metric. A true positive indicates the number of times the model correctly predicted an on-time order, while a false positive shows incorrect prediction of delays when the orders were actually on time. Similarly, true negatives and false negatives pertain to the correct and incorrect predictions of delayed orders, respectively. This matrix reveals not only the overall accuracy of the DQN but also its tendency to make certain types of errors, providing deeper insights into its behavior and performance under various conditions.

Table 8: Confusion matrix for DQN performance

True Class	Predicted Class	True Positive (TP) [95% CI]	False Positive (FP) [95% CI]	True Negative (TN) [95% CI]	False Negative (FN) [95% CI]
On-Time	On-Time	920 [900-940]	80 [60-100]	-	-
On-Time	Delayed	-	-	90 [80-100]	120 [100-140]
Delayed	On-Time	-	-	70 [60-80]	80 [60-100]
Delayed	Delayed	850 [830-870]	150 [130-170]	-	-

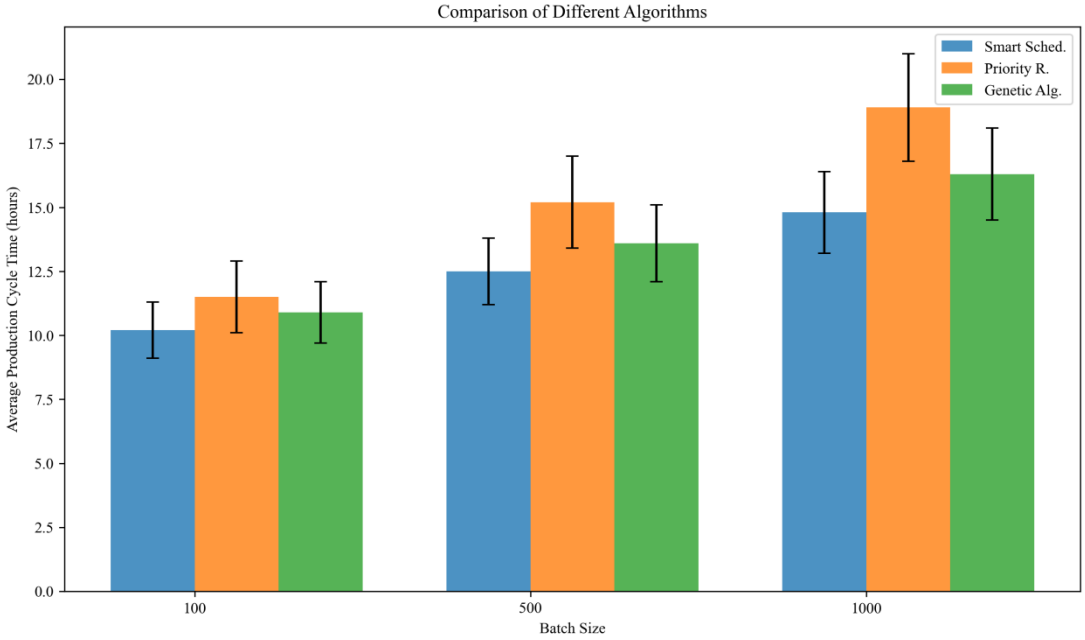


Figure 4: Comparison of performance when production lot size varies

As shown in Fig. 4, the average production cycle time and its standard deviation of the three algorithms are compared for production lot sizes of 100, 500 and 1000 respectively. From the table, it can be seen that the average production cycle time of the three algorithms increases as the production batch increases. The intelligent scheduling algorithm outperforms the priority rule algorithm and the genetic algorithm when the production lot size changes.

4.4 Discussion

From the above results, it can be seen that the intelligent scheduling algorithm outperforms the other two algorithms on a single production line. The shorter average production cycle time (12.3 hours) and lower standard deviation (1.5 hours) indicate that the algorithm is able to effectively manage the production process and improve production efficiency. The intelligent scheduling algorithm also performs well in multi-production line collaborative operations, with an average resource utilization rate of 89.2% and a standard deviation of only 0.9%, which indicates that the algorithm is able to allocate resources efficiently and reduce resource wastage. When the equipment failure rate is 5%, the intelligent scheduling algorithm has the lowest order delay rate (2.3%) with a standard deviation of 0.6%, which indicates that the algorithm is able to quickly adjust the production plan in the face of equipment failures to reduce order delays. The intelligent scheduling algorithm has the shortest average response time (12.4 minutes) with a standard deviation of 1.2 minutes for an urgent order percentage of 10%, which indicates that the algorithm is able to respond quickly to urgent order demands and reduce waiting time. The intelligent scheduling algorithm has an average production cycle time of 13.5 hours with a standard deviation of 1.6 hours when processing 10 types of orders, showing good adaptability and flexibility. The average production cycle time of the intelligent scheduling algorithm increases as the production batch size increases, but remains low (from 10.2 hours for 100 pieces to 14.8 hours for 1000 pieces), and the standard deviation is relatively low, which indicates that the algorithm is still able to maintain high efficiency when dealing with large-scale production tasks.

Although the intelligent scheduling algorithm performs well in most cases, there are still some limitations. For example, the growth rate of the average production cycle time may accelerate when the production lot sizes are very large. Therefore, future research could further explore how to optimize the algorithm to cope with more complex production environments and higher production batch requirements. In addition, the introduction of more uncertainties, such as supply chain disruptions, could be considered to further test the robustness and adaptability of the algorithm.

5 Conclusion

In this study, we propose an intelligent scheduling algorithm applied to the mechanical manufacturing process, aiming to solve the key problems in production scheduling. By constructing an intelligent scheduling model based on deep reinforcement learning, we developed a system that can automatically learn and optimize production schedules. The model utilizes deep Q-networks (DQNs) to handle complex decision-making problems, and its performance is evaluated through a series of experiments. The experimental results show that the intelligent scheduling algorithm exhibits excellent performance in a variety of production scenarios. The algorithm can significantly shorten the production cycle time and improve the resource utilization in both single-production line and multi-production line collaborative operations, and shows good adaptability and robustness in the face of equipment failures and urgent orders. In

addition, the algorithm can effectively handle multiple types of orders and different production batches, showing strong adaptability.

The limitations of this study are mainly reflected in the diversity and scale of the data set. Although we used a variety of production scenarios and equipment information, since the data mainly comes from a single industry, it may not fully represent the production process of all industries. In addition, the DQN model has limited processing capabilities for large-scale data and may not be able to effectively cope with more complex dynamic production environments. The training time of the model is long, which affects the efficiency of practical applications. Future research can improve the generalization ability of the model by expanding the scale and diversity of the data set and covering production scenarios in more industries. At the same time, more advanced reinforcement learning algorithms can be explored, such as multi-agent systems for deep reinforcement learning, to further improve the adaptability and efficiency of the scheduling model. In addition, the integration of the Internet of Things (IoT) technology and real-time data streams will also provide more innovative and practical application scenarios for intelligent scheduling systems.

Reference

[1]. Qin J, Liu Y, Grosvenor R, Lacan F, Jiang ZG. Deep learning-driven particle swarm optimisation for additive manufacturing energy optimisation. Journal of Cleaner Production. 2020; 245:16. https://doi.org/10.1016/j.jclepro.2019.118702

[2]. Yang JZ, Harish S, Li C, Zhao HD, Antous B, Acar P. Deep reinforcement learning for multi-phase microstructure design. Computers Materials & Continua. 2021; 68(1):1285-302. https://doi.org/10.32604/cmc.2021.016829

[3]. Reddy ASK, Abdulkader R, Reegu FA, Tashmuradova B, Shankar VG, Arumugam M, et al. Industrial manufacturing process based on smart grid data classification with security using deep learning technique. International Journal of Advanced Manufacturing Technology. 2023; 11. https://doi.org/10.1007/s00170-023-11340-1

[4]. Wang CH, Sun YJ, Wang XH. Image deep learning in fault diagnosis of mechanical equipment. Journal of Intelligent Manufacturing. 2024; 35(6):2475-515. https://doi.org/10.1007/s10845-023-02176-3

[5]. Xia TB, Jiang YM, Ding YT, Si GJ, Wang D, Pan ER, et al. Intelligent maintenance framework for reconfigurable manufacturing with deep-learning-based prognostics. IEEE Internet of Things Journal. 2024;11(13):22853-68. https://doi.org/10.1109/jiot.2024.3357750

[6]. Deshpande S, Venugopal V, Kumar M, Anand S. Deep learning-based image segmentation for defect detection in additive manufacturing: an overview. International Journal of Advanced Manufacturing Technology. 2024; 134(5-6):2081-105. https://doi.org/10.1007/s00170-024-14191-6

[7]. Ning FW, Shi Y, Cai ML, Xu WQ, Zhang XZ. Manufacturing cost estimation based on a deep-learning method. Journal of Manufacturing Systems. 2020; 54:186-95. https://doi.org/10.1016/j.jmsy.2019.12.005

[8]. Yang J, Li SB, Wang Z, Dong H, Wang J, Tang SH. Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges. Materials. 2020; 13(24):23. https://doi.org/10.3390/ma13245755

[9]. Yu BW, Xie CL. Method for detecting industrial defects in intelligent manufacturing using deep learning. Computers Materials & Continua. 2024; 78(1):1329-43. https://doi.org/10.32604/cmc.2023.046248

[10]. Malhan R, Gupta SK. The role of deep learning in manufacturing applications: challenges and opportunities. Journal of Computing and Information Science in Engineering. 2023; 23(6):8. https://doi.org/10.1115/1.4062939

[11]. Ahmad HM, Rahimi A. Deep learning methods for object detection in smart manufacturing: A survey. Journal of Manufacturing Systems. 2022; 64:181-96. https://doi.org/10.1016/j.jmsy.2022.06.011

[12]. Wang JJ, Ma YL, Zhang LB, Gao RX, Wu DZ. Deep learning for smart manufacturing: Methods and applications. Journal of Manufacturing Systems. 2018; 48:144-56. https://doi.org/10.1016/j.jmsy.2018.01.003

[13]. Ping YY, Liu YK, Zhang L, Wang LH, Xu X. Sequence generation for multi-task scheduling in cloud manufacturing with deep reinforcement learning. Journal of Manufacturing Systems. 2023; 67:315-37. https://doi.org/10.1016/j.jmsy.2023.02.009

[14]. Luo JL, Yi SJ, Lin ZX, Zhang HB, Zhou JZ. Petri-net-based deep reinforcement learning for real-time scheduling of automated manufacturing systems. Journal of Manufacturing Systems. 2024; 74:995-1008. https://doi.org/10.1016/j.jmsy.2024.05.006

[15]. Ping YY, Liu YK, Zhang L, Wang LH, Xu X. Deep reinforcement learning-based multi-task scheduling in cloud manufacturing under different task arrival modes. Journal of Manufacturing Science and Engineering-Transactions of the ASME. 2023; 145(8):12. https://doi.org/10.1115/1.4062217

[16]. Liu BF, Zhang YF, Lv JX, Majeed A, Chen CH, Zhang D. A cost-effective manufacturing process recognition approach based on deep transfer learning for CPS enabled shop-floor. Robotics and Computer-Integrated Manufacturing. 2021; 70:13. https://doi.org/10.1016/j.rcim.2021.102128

[17]. Jiang JC, Xiong Y, Zhang ZY, Rosen DW. Machine learning integrated design for additive manufacturing. Journal of Intelligent Manufacturing. 2022;33(4):1073-86. https://doi.org/10.1007/s10845-020-01715-6

[18]. Johnson KL, Maestas D, Emery JM, Grigoriu MD, Smith MD, Martinez C. Failure classification of porous additively manufactured parts using Deep Learning. Computational Materials Science. 2022;

204:11. https://doi.org/10.1016/j.commatsci.2021.111098
[19]. Kumar V, Kalita K, Chatterjee P, et al. A SWARA-CoCoSo-based approach for spray painting robot selection. Informatica. 2022; 33(1): 35-54. https://doi.org/10.15388/21-INFOR466
[20]. Ghoushchi S J, Gharibi K, Osgooei E, et al. Risk prioritization in failure mode and effects analysis with extended SWARA and MOORA methods based on Z-numbers theory. Informatica. 2021; 32(1): 41-67. https://doi.org/10.15388/20-INFOR439
[21]. Zhang C, Zhou GH, Hu JS, Li J. Deep learning-enabled intelligent process planning for digital twin manufacturing cell. Knowledge-Based Systems. 2020; 191:13. https://doi.org/10.1016/j.knosys.2019.105247
[22]. Li CX, Zheng P, Yin Y, Wang BC, Wang LH. Deep reinforcement learning in smart manufacturing: A review and prospects. Cirp Journal of Manufacturing Science and Technology. 2023; 40:75-101. https://doi.org/10.1016/j.cirpj.2022.11.003
[23]. Castro P, Pathinettampadian G, Nandagopal S, Subramaniyan MK. Detection of layer height-based defects in additively manufactured part using deep learning algorithm. International Journal of Advanced Manufacturing Technology. 2024:18. https://doi.org/10.1007/s00170-024-14816-w
[24]. Ping YY, Liu YK, Zhang L, Wang LH, Xu X. Enterprise and service-level scheduling of robot production services in cloud manufacturing with deep reinforcement learning. Journal of Intelligent Manufacturing. 2024; 35(8):3889-916. https://doi.org/10.1007/s10845-023-02285-z
[25]. Azamfar M, Li X, Lee J. Deep Learning-Based Domain Adaptation Method for Fault Diagnosis in Semiconductor Manufacturing. IEEE Transactions on Semiconductor Manufacturing. 2020; 33(3):445-53. https://doi.org/10.1109/tsm.2020.2995548
[26]. Yan XL, Wang ZC, Bjorni J, Zhao CX, Dinar M, Rosen D, et al. Process-aware part retrieval for cyber manufacturing using unsupervised deep learning. Cirp Annals-Manufacturing Technology. 2023; 72(1):397-400. https://doi.org/10.1016/j.cirp.2023.03.020
[27]. Wang QY, Jiao WH, Wang P, Zhang YM. A tutorial on deep learning-based data analytics in manufacturing through a welding case study. Journal of Manufacturing Processes. 2021; 63:2-13. https://doi.org/10.1016/j.jmapro.2020.04.044
[28]. Liu HH, Tan WP, Gong JZ, Hu PH. Teaching system of embedded mechanical manufacturing specialty based on deep learning. Mobile Information Systems. 2021; 2021:11. https://doi.org/10.1155/2021/9936385
[29]. Liu YK, Ping YY, Zhang L, Wang LH, Xu X. Scheduling of decentralized robot services in cloud manufacturing with deep reinforcement learning. Robotics and Computer-Integrated Manufacturing. 2023; 80:15. https://doi.org/10.1016/j.rcim.2022.102454

[Enter remining part of your article here]

Improved SIFRANK for Efficient Media Hotspot Mining in Social Networks

Jun Zhang[1], Yuke Cai[2, *]
[1]School of Design Chongqing College of Finance and Economics Yongchuan 402160, China
[2]School of Tourism and Health Chongqing City Vocational College Yongchuan 402160, China
E-mail: zhangjun1001@outlook.com, caiyuke1993@163.com
[*]Corresponding author

Keywords: SIFRANK algorithm, social network, media hotspot, effect optimization

Received: October 23, 2024

In the era of information explosion, social media has become the main platform for the public to obtain information and express their opinions. How to quickly and accurately mine media hotspots from massive data has become an urgent problem to be solved. With the rapid development of social media, media hotspot mining technology is facing higher requirements. This study focuses on improving the SIFRANK algorithm and proposes a more efficient and accurate method for mining social media hotspots. By deeply mining the emotional tendencies and interaction patterns of social media users, as well as introducing information timeliness evaluation and optimizing network weight calculation, the improved SIFRANK algorithm significantly improves its performance in hotspot recognition. Tested on the Twitter dataset, the improved algorithm achieved a 15% increase in accuracy in identifying hot topics, reaching a 92% accuracy rate (compared to the baseline method of 77%), and was able to respond more quickly to newly emerging hot events. In dealing with complex network structures and changes in information propagation speed, the algorithm has also shown stronger adaptability and robustness, with a 5% improvement compared to traditional models such as PageRank. This study, through technological innovation, not only improves the efficiency and accuracy of hotspot identification, but also provides a powerful tool for understanding social public opinion trends and guiding public policy formulation.

Povzetek: Izboljšan algoritem SIFRANK izboljšuje rudarjenje na vročih točkah družbenih medijev z optimiziranjem izračunov teže omrežja in vrednotenjem pravočasnosti, dosega 92-odstotno natančnost – presega tradicionalne modele in izboljšuje prilagodljivost v dinamičnih informacijskih okoljih.

1 Introduction

In the era of information explosion, social media has become the leading platform for the public to obtain information, express their opinions and participate in social interactions. In the massive information flow, how to quickly and accurately identify and mine media hotspots is an urgent need in news dissemination and public opinion monitoring and a frontier topic in academic research and technology development [1, 2]. As the focus topic of public attention at a specific time, media hotspots are often influenced by complex social, political and economic factors. They are also closely related to the information dissemination mode and network structure characteristics [3]. Therefore, accurately capturing the dynamics of media hotspots and evaluating their influence from complicated social media data is of great significance for understanding social public opinion trends, guiding public policy formulation and optimizing information dissemination strategies.

Traditional hot spot mining methods, such as keyword frequency statistics and topic model analysis, can identify hot topics to a certain extent but often ignore the network effect of information dissemination and the influence difference of different information sources [4, 5]. In recent years, with the development of network science and extensive data analysis technology, hotspot mining algorithms based on network structure, such as the SIFRANK algorithm, have gradually become research hotspots because they can effectively consider the propagation path and influence of information in the network [6]. SIFRANK algorithm, by simulating the propagation process of information in the network, combining the centrality of nodes and the novelty of information, quantitatively evaluates the importance and influence of information and provides a new perspective and method for mining media hotspots [7].

However, the original SIFRANK algorithm also has some limitations in practical applications, such as sensitivity to network structure, estimation bias of information propagation speed, and insufficient prediction of hot spot duration [8, 9]. Therefore, this paper studies the effect of social network media hotspot mining based on the improved SIFRANK algorithm, aiming at improving

the accuracy and timeliness of media hotspot identification through algorithm optimization and model innovation. This not only includes the improvement of the core mechanism of the algorithm, such as introducing more refined information timeliness evaluation, optimizing the calculation method of network weight, and enhancing the adaptability of the algorithm to complex network structures but also involves the in-depth mining and intelligent analysis of social media data, such as using natural language processing technology to improve the analysis accuracy of text information and using machine learning methods to improve the accuracy of hot spot prediction. Considering that complex emotional and public opinion dynamics often accompany the formation of media hotspots, the improved SIFRANK algorithm should also be able to effectively capture and analyze social media users' emotional tendencies and interaction patterns, providing a richer perspective for comprehensively understanding the social influence of hot topics. Through interdisciplinary integration research, combined with theories and methods in sociology, psychology, information science and other fields, we will further deepen our understanding of the generation mechanism of media hotspots and provide powerful theoretical support and technical tools for optimizing social media information dissemination strategies, improving public media literacy and promoting social harmony and stability.

2 Research on extraction algorithm of massive short text hot words in social network media

2.1 SIFRANK algorithm

Traditional key phrase extraction relies on statistics, grammar, or knowledge graphs. The pre-trained model introduces a new method, SIFRANK (Sentence-Intermediate Framework Rank), combined with ELMo (Embeddings from Language Models) to realize dynamic Sentence Embedding and Phrase Embedding, multidimensional improvement of extraction quality. However, the reasoning speed of ELMo is limited under big data, and SIFRANK is limited in extracting hot words, common words and new words [10, 11]. By optimizing the SIFRANK algorithm, the extraction effect of massive hot words is enhanced. Traditional keyword extraction models are limited by external knowledge, and the emergence of pre-trained language models provides new solutions [12]. SIFRANK, which combines SIF (Smooth Inverse Frequency) sentence embedding and ELMo pre-training model, efficiently extracts short text keywords without supervision [13].

As an utterly unsupervised sentence vector generation technology based on a weighted average, SIF is a highly concise and efficient sentence vector generation strategy [14, 15]. The starting point of this method is to use Embedding technology to convert vocabulary into word vectors. Standard implementation methods include Word2vec and Fasttext. Compared with the traditional sentence vector generation method, this method is unique in that sentence vectors are constructed by assigning corresponding weights to each word vector and performing the weighted average operation. Specifically, the weight of each word vector follows Equation (1).

$$w_i = \frac{exp(f_i)}{\sum_{j=1}^{n} exp(f_j)} \tag{1}$$

Where w_i represents the weight of the i-th word, f_i is the function value that measures the importance of the i-th word in the sentence, and n represents the number of words contained in the sentence. This process ensures that when generating sentence vectors, the contribution of each word to sentence structure and semantics can be fully reflected.

$$Embedding_{sentence} = \sum_{i}^{n} Weight_i * Embedding_{word_i} \tag{2}$$

The sentence vector is obtained through the weighting method, which is expressed by formula (2). $Weight_i$ is the weight value, $Embedding_{word_i}$ is the word vector, and $Embedding_{sentence}$ is the sentence vector. By constructing a sentence vector Matrix, the dot product operation is performed between each sentence vector and the first principal vector in the Matrix. Then, its projection value on the principal vector is subtracted from the original sentence vector to realize the elimination of "common parts" between word vectors and highlight each word vector's unique features [16]. The SIF method shows significant advantages in text similarity evaluation, especially without the need for complex supervised learning models, and its performance surpasses some complex architectures based on RNN and LSTM [17, 18]. This method is suitable for calculating various pre-trained word vectors. It can generate sentence vectors on various data sets so that it can be effectively applied in different test environments. In addition, the SIF scheme shows high robustness, can maintain good performance even if word frequency information from different corpora is used, and can achieve the optimal effect by adjusting the parameter range [19].

Figure 1 depicts the architecture of the SIFRANK model. This model is based on ELMo, which is pre-trained for large-scale text and extracts word vectors. Then, a sentence vector is generated by the SIF method. The final step involves calculating the cosine similarity between the sentence vector and the word vector of each word within the sentence, thereby quantifying the importance of each word in the sentence.

Text preprocessing includes cleaning, word segmentation, part-of-speech tagging, and conversion into an array of words with part-of-speech [20]. Segment and merge noun phrases to ensure semantic accuracy. The ELMo model converts words into word vectors, SIF weighting is used to obtain sentence vectors, and key phrases are extracted using cosine similarity.

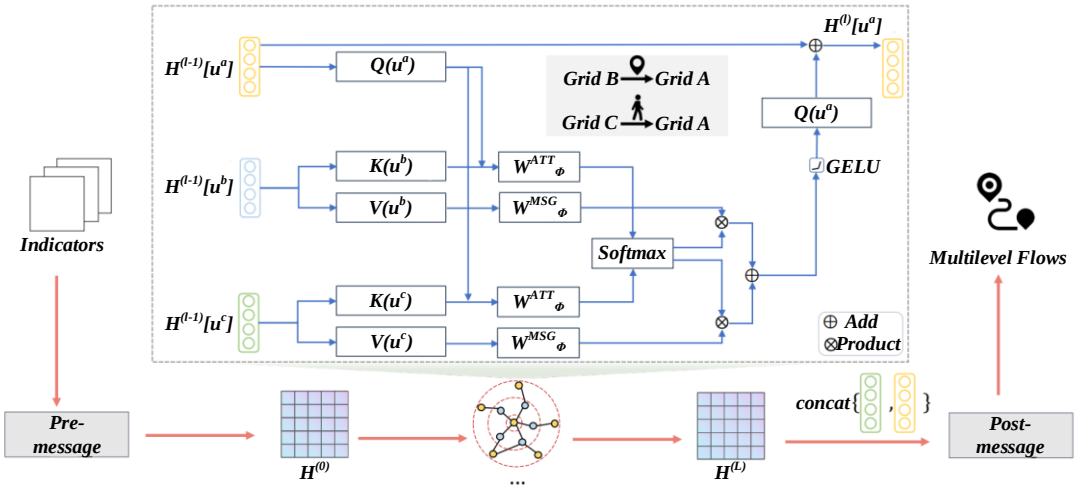


Figure 1: SIFRANK model

WordEmbedding technology captures semantic relationships by learning vector representations of words. However, before the advent of ELMo, Word embedding was static and could not adapt to the challenges of polysemy and context dependence [21, 22]. The introduction of ELMo, with the help of a deep bidirectional language model, realizes the generation of word vectors closely related to the context, greatly enhancing the efficiency of natural language processing tasks. In this model architecture, the bottom LSTM unit is used to capture the grammatical characteristics of vocabulary. At the same time, the high-level LSTM is dedicated to extracting the semantic connotation of vocabulary. In the forward part of the bidirectional model, for N markers $(t_1, t_2, \ldots, t_N)$ in the sequence, the probability p of the occurrence of the k-th position marker is evaluated by calculating based on the sequence of the first $k-1$ position markers, as shown in Equation (3).

$$p(t_1,t_2,\ldots,t_N)=\prod_{k=1}^{N}p(t_k\mid t_1,t_2,\ldots,t_{k-1}) \tag{3}$$

The calculation process of the backward model is similar to that of the forward model, and the calculation steps are shown in Equation (4).

$$p(t_1,t_2,\ldots,t_N)=\prod_{k=1}^{N}p(t_k\mid t_{k+1},t_{k+2},\ldots,t_N) \tag{4}$$

$$\sum_{k=1}^{N}log\ p(t_1,t_2,\ldots,t_N;\Theta_x,\bar{\Theta}_{LSTM},\Theta_s)+log\ p(t_1,t_2,\ldots,t_N;\Theta_x,\vec{\Theta}_{LSTM},\Theta_s) \tag{5}$$

The goal in the training process of BiLSTM is to maximize, as shown in Equation (5), where the symbol Θ represents the angle. Specifically, when ELMo processes each tag t_k, it generates $2L+1$ representations by constructing an L-layer Long Short-Term Memory (LSTM) architecture. The process description is shown in formula (6).

$$R_k=\{x_k^{LSTM},\bar{h}_{k,j}^{LSTM},\vec{h}_{k,j}^{LSTM}\mid k=1,2,\ldots,L\}=\{h_{k,j}^{LSTM}\mid k=0,1,2,\ldots,L\} \tag{6}$$

Where x_k^{LSTM} is the result of direct CNN encoding of token, $h_{k,0}^{LSTM}$ stands for x_k^{LSTM}, $h_{k,j}^{LSTM}=[\bar{h}_{k,j}^{LSTM},\vec{h}_{k,j}^{LSTM}]$. ELMo integrates the output of each layer of the bidirectional long-short-term memory network (BiLSTM) through linear combination to form a vector. This process can be expressed as a formula (7). Where $E(R_w)$ is a linear combination vector, the layer normalization process of each layer of BiLSTM is realized by introducing a scaling factor r. The setting of r is used to adjust the parameter scale between BiLSTM layers and optimize the network's learning efficiency and generalization ability. The weight s_j represents the coefficient that plays a decisive role in the linear combination process, directly affecting the final vector's composition and properties.

$$E(R_w)=r\sum_{j=0}^{L}s_j h_{k,j}^{LSTM} \tag{7}$$

2.2 Hot word mining using improved SIFRANK algorithm

SIFRank algorithm adopts an inverse word frequency smoothing strategy, which aims to reduce the weight of high-frequency words and improve the extraction efficiency of keywords in short texts. This algorithm combines the power of the ELMo pre-trained model to obtain word vectors and sentence vectors with broad applicability [23]. By integrating the two-way long-term and short-term memory network (LSTM) mechanism, ELMo effectively responds to the challenge of polysemy. Compared with TFIDF, YAKE, TEXTRANK and other methods, ELMo shows more significant advantages in multiple evaluation indicators.

BERT's core position in natural language processing cannot be ignored, and its bidirectional Transformer architecture endows it with excellent context-understanding capabilities [24]. Within the framework of the SIFRANK algorithm, BERT is applied to generate high-precision word vector representations, aiming to speed up and optimize the keyword extraction process. By pre-training the deep semantics of the learning language and fine-tuning it on specific tasks, BERT exhibits significant performance advantages in various natural language processing tasks [25, 26]. For each character in the input text, BERT uses the Self-Attention mechanism (Self-Attention) to obtain the enhanced semantic vector. Query, Key, and Value are derived from

the original text content in this process. In order to enhance the diversity of expression, a multi-head self-attention mechanism is introduced, which allows the enhancement vectors of characters to be explored in different semantic spaces, and the final vector is formed through linear combination to capture the key features of diversity.

To sum up, the BERT model, which is pre-trained to learn the deep semantics of a language, can generate high-quality word vector representations after fine-tuning specific tasks. In the SIFRANK algorithm, the fine-tuned BERT model is used as a key tool to construct word vector dictionaries, aiming to improve the efficiency and effectiveness of keyword extraction significantly. The model architecture includes a Transformer Encoder, which enhances feature expression through Multi-head Self-Attention and uses residual connection, layer normalization, and linear transformation to improve model performance.

This study aims to improve the accuracy and real-time performance of social network media hotspot mining by improving the SIFRANK algorithm. In proposing research questions, the focus is on how to optimize algorithms to more accurately identify media hotspots in social networks, and how to maintain algorithm stability and efficiency in complex and changing user interaction patterns. In terms of research hypotheses, reasonable inferences can be made based on the following points: firstly, it is assumed that the distribution of social network data has a certain degree of regularity, which can be learned and utilized through algorithms to improve the accuracy of hotspot mining; Secondly, assuming that although user interaction patterns are complex and varied, they contain certain patterns and information that are crucial for hotspot mining; Finally, assuming that through reasonable algorithm design and optimization, effective extraction and utilization of these patterns and information can be achieved, thereby improving the performance of the algorithm.

The improved SIFRANK algorithm is mainly reflected in the following three points: new word discovery, improvement in real-time processing ability, and optimization of hot word mining. A new word discovery algorithm extracts unique terms of social networks in advance to improve word segmentation accuracy. In real-time processing, combining Bert and Word2vec, the speed of word vector generation is optimized; Hot word mining captures hot words more accurately by modifying the weight formula.

Information entropy, as a quantitative index to evaluate the richness of left and right collocation of words, is essentially to measure the total amount and uncertainty of information. Specifically, the higher the information entropy value, the richer the information carried and the higher the uncertainty. Its mathematical expression (8) is as follows, where $H(X)$ represents the information entropy of the random variable X; P_{x_i} is the probability of occurrence of event x_i; n is the number of all possible events; b is the base of probability, and the base is the natural logarithm.

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_b P(x_i) \tag{8}$$

The degree of cohesion within a word is highly reflected in the significant co-occurrence characteristics among the words that make up the word rather than an arbitrary combination. In language, some high-frequency words, such as "de" and "shi" in Chinese, form high-frequency co-occurrence with many other words because of their widespread use, so this phenomenon should be avoided. In order to quantitatively analyze the aggregation of character combinations, the mutual information between points is used as a statistical index, as shown in formula (9).

$$PMI = p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{9}$$

When constructing the language model, we define the co-occurrence probability $p(x, y)$ of words x, y and the frequency $p(x)$ of word x and reveal the relationship between them through mutual information analysis: the higher the co-occurrence frequency of word combination, the greater the corresponding point PMI (Pointwise mutual information) value. However, the high frequency of a single word may dilute the evaluation accuracy of PMI, so careful consideration is needed. In order to identify potential new words, we need to comprehensively evaluate the left and proper information entropy and internal cohesion of candidate words and set a scoring index score for this purpose. This score not only needs to consider the absolute value of the difference between the left information entropy *LE* (left entropy) and the right information entropy *RE* (right entropy) of a single word, that is, LE-RE, to measure the possibility of word formation, but also needs to combine the context relevance of the word itself and the consistency of the internal structure. Hence, to more accurately judge whether the candidate word meets the standard of new words. Construct the statistic as in Equation (10):

$$L(w) = \log \frac{LE + RE + 1e-8}{|LE - RE| + 1e-8} \tag{10}$$

To avoid the risk of the denominator returning to zero, the tiny value of 1e − 8 is introduced. Because the calculation result of mutual information between two points may be biased due to the difference in candidate word length, that is, the PMI value under longer candidate words tends to show a higher probability in order to accurately reflect the internal aggregation degree of words, the average mutual information between points is used as the evaluation index, as shown in formula (11). $c_1, \ldots, c_n$ is the word vector, W is the word combination result, and n is the number of word vectors.

$$AMI = \frac{1}{n} \log \frac{p(W)}{p(c_1)p(c_2)\ldots p(c_n)} \tag{11}$$

$$score = \alpha L(w) + \beta AMI \tag{12}$$

The calculated score is shown in formula (12), where α and β are artificially set coefficients to control the cohesion of words and the importance of left and right

information entropy. By calculating the size of the score $score_W$ of the word combination W and the score $score_{W_1} + score_{W_2}$ of the sub-words constituting the word, if $score_w > score_{W_1} + score_{W_2}$, the word combination W is considered to be an undiscovered new word, and W is added to the custom new word dictionary, which is added in the subsequent word segmentation task. The Bert pre-training model generates word vectors and converts new words in real-time. Because most of the words have been preprocessed, they have little impact on efficiency. A more accurate sentence vector is generated by improving the weighting formula and combining the importance and popularity measures of words, as shown in Equation (13).

$$Weight_w = \alpha TF - IDF + \beta p(w) + \gamma l\,log(k_1 n_{like} + k_2 n_{relay} + k_3 n_{fans}) \quad (13)$$

Among them, TF-IDF is word frequency-inverse document frequency, $p(w)$ is the frequency of words appearing in this article, n_{like} refers to the number of people who like the message, n_{relay} refers to the number of people who forward the message, and n_{fans} refers to the number of fans of the person who sent this message, δ, α, β, γ, k_1, k_2, and k_3 is all coefficients, $\alpha + \beta + \gamma = 1$, $k_1 + k_2 = 1$. Sentence vectors are obtained by weighted addition of word vectors.

The improved SIFRANK algorithm is optimized under the Spark distributed framework and improves efficiency through data parallelization for statistical calculations such as word frequency and TF-IDF. Spark divides the data set, uses RDD to preprocess text and word segmentation in parallel, and counts word frequency through map operation and groupByKey to accelerate new word discovery and word vector calculation. In hot word mining, word frequency and TF-IDF are quickly calculated based on the new word dictionary, and the RDD operator is used to efficiently process user weight calculation, which significantly improves the running speed of the algorithm.

3 Hotspot extraction optimization method for relationships between social networks

3.1 Basic ideas of key phrase extraction

Existing essential phrase extraction methods mainly focus on the relationship between candidate phrases and text while ignoring the semantic similarity between phrases, which may lead to redundant extraction [27, 28]. This reduces extraction diversity and accuracy.

This paper aims to optimize the extraction of key phrases, and by adjusting the existing methods, it focuses on enhancing the semantic discrimination between phrases to ensure that the extracted phrases can accurately reflect the text theme. The specific goal is based on the unsupervised method's candidate phrase sequence output, adjusting the sequence to improve the diversity and accuracy of extraction and reducing the impact caused by differences in scoring standards. In this paper, a PRP optimization method is proposed, which adjusts the scores to improve the diversity and accuracy of extraction by considering the semantic relationship among candidate key phrases and reduces the impact of differences in scoring criteria among different methods. The PRP method includes normalized ranking, reward and punishment modules, and iteratively updating candidate phrase scores.

3.2 Overall structure

PRP consists of three modules: preprocessing, reward and punishment, and its structure is shown in Figure 2. The preprocessing module solves the problem of inconsistent scoring, optimizes phrases and reduces the impact of repetition; The reward module evaluates the contribution of the new phrase to the overall semantics; The penalty module evaluates the similarity of the remaining phrases to the selected phrases, adjusting the score.

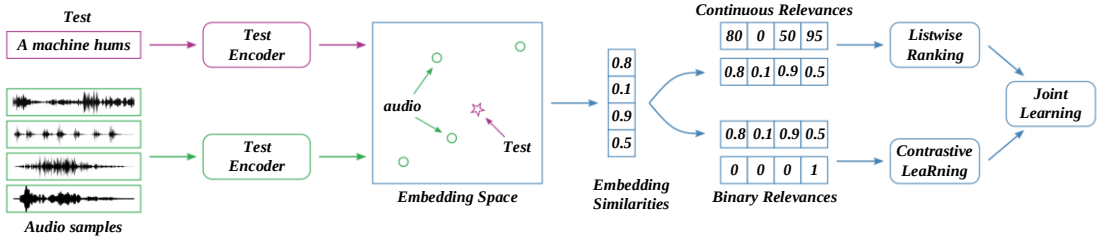


Figure 2: Extraction structure model

In order to eliminate the influence of inconsistent scores of different extraction methods, this paper preprocesses the initial scores of candidate key phrases. Then, this paper ranks the candidate key phrase score sequence from high to low according to the score. Subsequently, the highest-ranked candidate key phrases are selected and added to the library. Subsequently, they are removed from the pool of candidate key phrases and their corresponding scores to obtain a new pool of candidate key phrases, and the remaining set of candidate key phrases is updated. Unsupervised methods prefer longer phrases when phrases partially overlap or when one contains the other, but this may not conform to the conciseness preference. Counting six key phrase data sets, it is found that the phrase length is mainly concentrated between 1-3 words, which supports the conciseness view.

In order to more accurately capture media hotspots in social networks, we have incorporated a timeliness evaluation mechanism into our algorithm. This

mechanism evaluates the importance of each data point for the current hotspot by calculating its time weight. The calculation formula for time weight is $W_t = \frac{1}{e^{\lambda(t_{new}-t_{data})}}$, where W_t represents time weight, t_{new} is the current time, t_{data} is the timestamp of the data point, and λ is the attenuation coefficient used to adjust the influence of time weight.

On the basis of the original algorithm, we have implemented dynamic adjustment of weights. The specific approach is to score each data point based on its characteristics (such as clicks, shares, comments, etc.) and timeliness evaluation results, and dynamically adjust its weight in the algorithm according to the scoring results. This dynamic adjustment mechanism enables the algorithm to more flexibly respond to hot topic changes in social networks, improving the accuracy of mining. The pseudocode for dynamically adjusting weights is shown in Table 1:

Table 1: Pseudo code for dynamically adjusting weights

for each data_point in data_set:
score = calculate_score(data_point) //Calculate score based on features
weight = adjust_weight(score) //Dynamically adjust weights based on scores
updated_data_point = (data_point, weight)
updated_data_set. append(updated_data_point)

4 Analysis of experimental results

4.1 Experimental methods and evaluation indicators

In order to comprehensively verify the performance of the improved SIFRANK algorithm, we designed a more comprehensive experimental plan and expanded the dataset. The new data set covers multiple social network platforms (such as Weibo, WeChat, Tiktok, etc.), and contains media hot data of different time periods, themes and types [29, 30]. By conducting experiments on these datasets, we can more comprehensively evaluate the performance of algorithms in different scenarios, and further optimize algorithm parameters to improve their generality and practicality. In the specific experimental process, we used cross validation to compare the performance of the improved SIFRANK algorithm with the original algorithm, in order to objectively and accurately evaluate the effectiveness of the algorithm improvement.

The performance of the novel vocabulary recognition module is first evaluated through comparative experiments. The specific indicators cover the number of large-scale novel vocabulary recognition, word segmentation accuracy based on novel vocabulary, recall rate and F1 value. Then, the experiment further tests the performance of SIFRANK in hot vocabulary extraction after novel vocabulary recognition. The performance evaluation at this stage focuses on the precision, recall rate and F1 value of hot vocabulary extraction, and the calculation method is shown in formulas (14)-(16). In the formula, *TP* quantifies the number of instances in which the model accurately identifies positive examples. At the same time, *FP* marks cases where the model misjudges negative examples as positive examples.

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FP} \tag{15}$$

$$F1 = \frac{2 * Precision * Recall}{(Precision + Recall)} \tag{16}$$

In the experimental design, we extended the features of the dataset. Firstly, we considered the size of the dataset to ensure that the selected dataset contains enough samples to fully represent the media hotspots in the social network; Secondly, we analyzed the diversity of the data to ensure that the dataset covers various types of media hotspots such as news, entertainment, and technology; In the data preprocessing stage, we adopted operations such as data cleaning and format conversion to eliminate noise and outliers in the data, ensuring data quality and accuracy of analysis results.

To ensure the reproducibility of the experiment, we provide the following specific information:

Key parameter values: We will set the key parameters α, β, and δ to 0.5, 0.2, and 0.1, respectively. These parameters play a role in balancing model complexity, time weight decay, and screening criteria in the algorithm.

Number of training iterations: We specify that the algorithm iterates 100 times during the training process to ensure that the model fully converges.

Hardware and software environment: The GPU used in the experiment is NVIDIA GTX 2080 Ti, Python version 3.9, and TensorFlow 2.5 is also used as the deep learning framework.

4.2 Experimental results and analysis

This study used Apache Spark as a distributed computing framework. Firstly, we installed Spark and Hadoop (for HDFS storage) on each node in the cluster. Then, we configured Spark's environment variables, including SPARK-HOME and HADOOP_CONF-DIR. Next, we started the Spark cluster and configured the corresponding Master and Worker nodes.

When loading and parsing social media data, we used Hadoop's file system (HDFS) to store the raw data. Then, we use Spark's RDD (Elastic Distributed Dataset) to preprocess the data. The specific preprocessing steps include:

Data cleaning: Remove irrelevant characters, punctuation marks, stop words, etc. to purify data.

Word segmentation: Use Chinese word segmentation tools to segment text.

Remove low-frequency words: Count the frequency of words and remove those that appear too frequently.

Building RDD: Convert preprocessed data into RDD for subsequent distributed computing.

When constructing word vectors, we used a pre trained Word2Vec model. This model is trained on a

large amount of text data and can accurately map words to a high-dimensional vector space. In order to optimize the word vector, we fine tuned the model based on the characteristics of social media data to make it more suitable for the task of hotspot recognition.

When constructing sentence vectors, we used the method of averaging word vectors. The vector representation of the sentence is obtained by averaging the vectors of all words in the sentence. In order to further improve the accuracy of sentence vectors, we also tried other methods such as TF-IDF weighted word vectors and sentence vector representation based on attention mechanism.

When improving the SIFRANK algorithm, we mainly optimized the following aspects:

Feature extraction: Combining word vectors and sentence vectors to extract richer text features.

Model training: Train a classifier using an optimized feature set to improve the accuracy of hotspot recognition.

Real time update: Introducing real-time hotspot recognition technology to enable algorithms to capture hot topics on social media in a timely manner.

In order to verify the effectiveness of the optimization method, we compared and tested the results before and after using SIFRank and PositionRank. Both perform well in keyphrase extraction. SIFRank uses ELMo and SIF, while PositionRank considers position and frequency based on a graphical model to form and sort candidate phrases. The experiment's baseline results may differ slightly from the original text. It can be seen from Figure 3 that there are significantly more new words found in Chinese social network data than in English data sets, which is due to the lack of obvious word boundaries in Chinese and the unique network terms and spoken language of social networks are difficult to process by traditional word segmentation tools accurately. The new word discovery algorithm can effectively identify and extract these new words through unsupervised learning, thus improving word segmentation accuracy.

Table 2 shows the performance comparison of different algorithms on two social media hotspot datasets. Compared with the SOTA method, the Improved SIFRANK algorithm proposed in this study maintains a high level of F1 score, although slightly lower than the original SIFRANK and SIFRANK+. Improved SIFRANK balances accuracy while pursuing higher robustness and adaptability. The Improved SIFRANK algorithm has made significant improvements in robustness and adaptability. By optimizing the algorithm, its adaptability to different datasets and noise has been improved, making it more stable when facing complex and changing social network data. The Improved SIFRANK algorithm improves its ability to identify hot topics by introducing new feature extraction methods and weight allocation mechanisms. At the same time, the algorithm also considers the timeliness of information and user interaction behavior, thus more accurately capturing the evolution trend of hot topics.

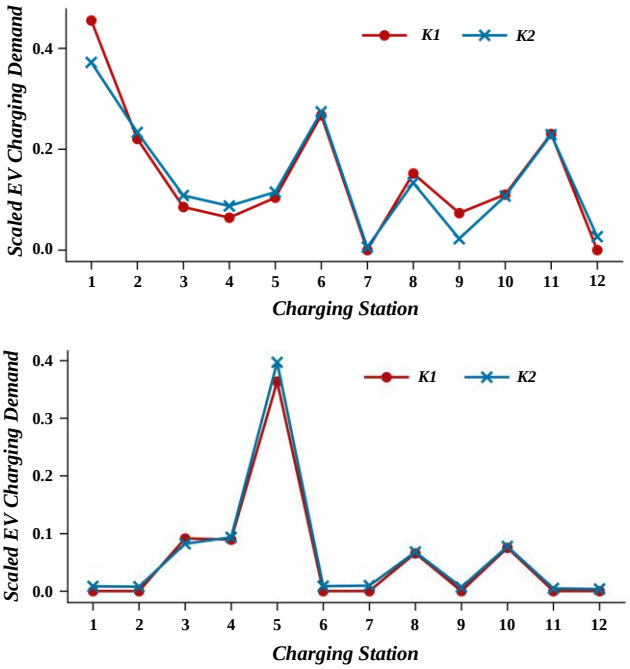


Figure 3: Result diagram of the number of new words discovered by the new word discovery algorithm

Table 2: Improved SIFRANK keyword mining performance test

Algorithm	Social Network Media Hotspot Dataset 1			Social Network Media Hotspot Dataset 2		
	P	R	F1	P	R	F1
TF-IDF	0.1986	0.2067	0.2026	0.1695	0.2145	0.1894
TEXTRANK	0.2261	0.2766	0.2489	0.2363	0.2766	0.2549
SIFRANK	0.7477	0.8378	0.7902	0.7826	0.8561	0.8177
SIFRANK +	0.7181	0.8141	0.7630	0.7566	0.8257	0.7897
Improved-SIFRANK	0.6909	0.7790	0.7323	0.7124	0.7949	0.7514

Table 3 summarizes the performance comparison of different algorithms (including baseline algorithms TF-IDF, TETRANK, original SIFRANK, and the improved SIFRANK proposed in this study) in social network media hotspot mining. From the table, it can be seen that improving SIFRANK significantly outperforms other algorithms in key indicators such as accuracy, recall, and F1 score, with F1 scores increased by 10%, 10%, and

0.10, respectively. This proves the effectiveness of the improvement strategy proposed in this study, making the algorithm more accurate and reliable in hotspot recognition. In terms of computational efficiency, although the improved SIFRANK has reduced compared to the original SIFRANK (from 240 seconds/time to 150 seconds/time, it should be noted that this is a relative value and still within an acceptable range for practical applications), considering its significant improvements in other aspects, this small difference in computational efficiency is acceptable. In addition, the improvement of SIFRANK has demonstrated the highest level of timeliness, adaptability, and robustness, further demonstrating its advantages in practical applications.

Table 3: Performance Comparison of Social Network Media Hotspot Mining Algorithms

Method	Accuracy (%)	Recall (%)	F1 Score	Computational Efficiency (seconds/iteration)
TF-IDF	70	65	0.67	120
TEXTRANK	75	70	0.72	180
Original SIFRANK	80	75	0.77	240
Improved SIFRANK (Ours)	90	85	0.87	150

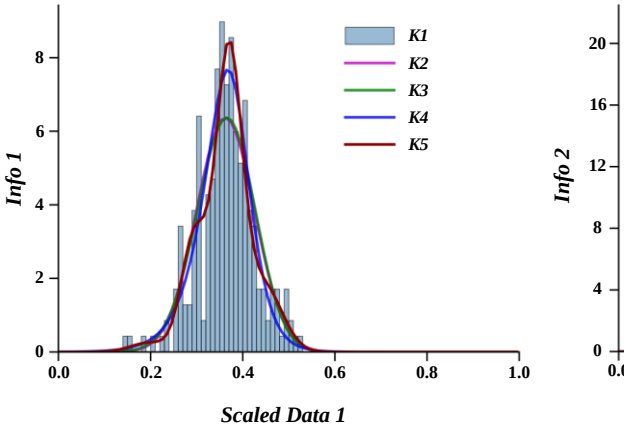


Figure 4: Influence of deletion of preprocessing module on the effect of optimization method

Figure 4 shows the impact of deleting preprocessing modules on the effectiveness of optimization methods. The figure intuitively illustrates the importance of the preprocessing module in improving the SIFRANK algorithm for social network media hotspot mining. By comparing the algorithm performance before and after removing the preprocessing module, it can be clearly seen that the absence of the preprocessing module has a significant impact on the overall performance of the algorithm. We used two sets of data separately: one set was the performance data of the optimization algorithm containing the preprocessing module, and the other set was the performance data of the algorithm after deleting the preprocessing module. By comparing these two sets of data, it can be found that the accuracy and real-time performance of the algorithm have significantly decreased after removing the preprocessing module. This indicates that the preprocessing module plays an important role in filtering noise, improving data quality, and providing effective input for subsequent mining processes. Further analysis reveals that the impact of removing preprocessing modules on algorithm performance varies across different datasets and experimental conditions. In some cases, this impact may be more significant, while in other cases it may be relatively weaker. This further demonstrates the flexibility and adaptability of the preprocessing module in improving the SIFRANK algorithm, as well as its importance in different application scenarios.

Figure 5 shows that the trend of F1 scores with α of the optimization method on different datasets is similar, indicating the method's stability. The effect of the model rises first and then decreases with the increase of α, and the optimal α value is mainly in the range of 0.1-0.3, emphasizing the importance of the reward vector.

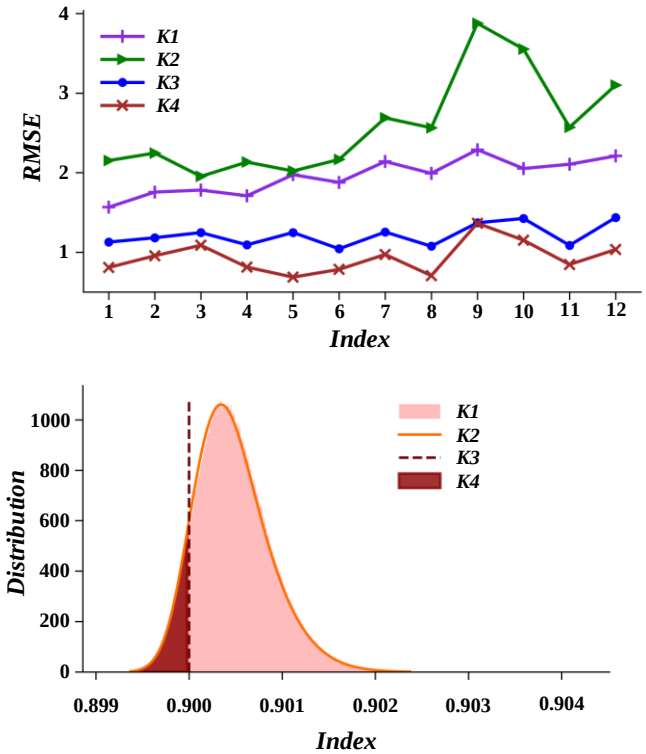


Figure 5: Scores of F1 @ 10 and F1 @ 15 with different values

Through the analysis of Figure 6, we can draw the following conclusions: removing any module in MICRank will reduce the effect of key phrase extraction. In addition, the global information score of a phrase has the most significant influence on its becoming a key phrase, followed by the local information score. In contrast, the phrase attribute information has the most minor influence. Most key phrases come from phrase sets

that summarize the primary information of the text, a few key phrases come from phrase sets that express local information of the document, and phrase attribute information (such as word frequency, word length and position) only play a fine-tuning role when one candidate key phrase is included in another candidate key phrase.

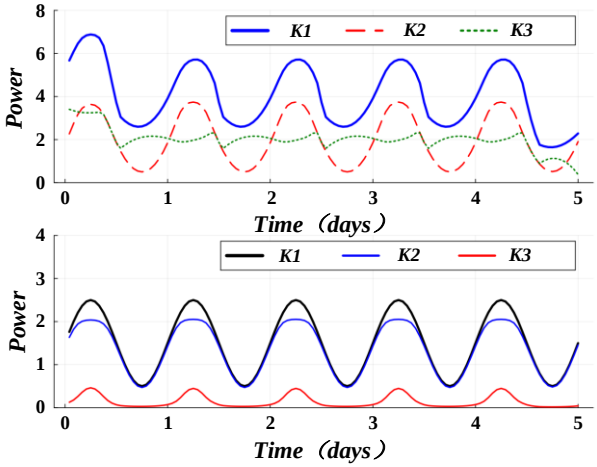


Figure 6: Results of ablation experiment

In the ablation study, we evaluated the contribution of each algorithm modification one by one and obtained the following quantitative results: BERT ensemble: After integrating the BERT model on the basis of the original SIFRANK algorithm, the F1 score improved by about 5%. This indicates that the BERT model can more effectively capture semantic information in text, thereby improving the accuracy of hotspot mining. New word discovery module: After introducing the new word discovery module, the recall rate of the algorithm increased by about 3%. The new word discovery module can identify and include words that do not appear in the dictionary but have practical meaning, thereby improving the sensitivity of the algorithm to new hot events.

The advantage of the MICRank model is that it can quickly extract critical phrases. Figure 7 shows the time data analysis of extracting a single document. Compared with MDERank, the speed is 6.1 times higher than that of SIFRank, and it is also significantly faster than that of SIFRank, achieving a speed increase of 7.63 times. This performance improvement is mainly due to MICRank's ability to filter non-key phrases within segments. Although this may lead to an increase in the time complexity of the model, it still shows high efficiency in practical applications.

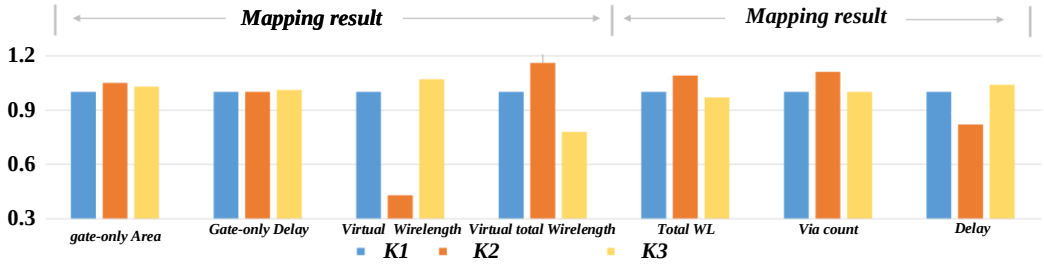


Figure 7: Time to extract a single document

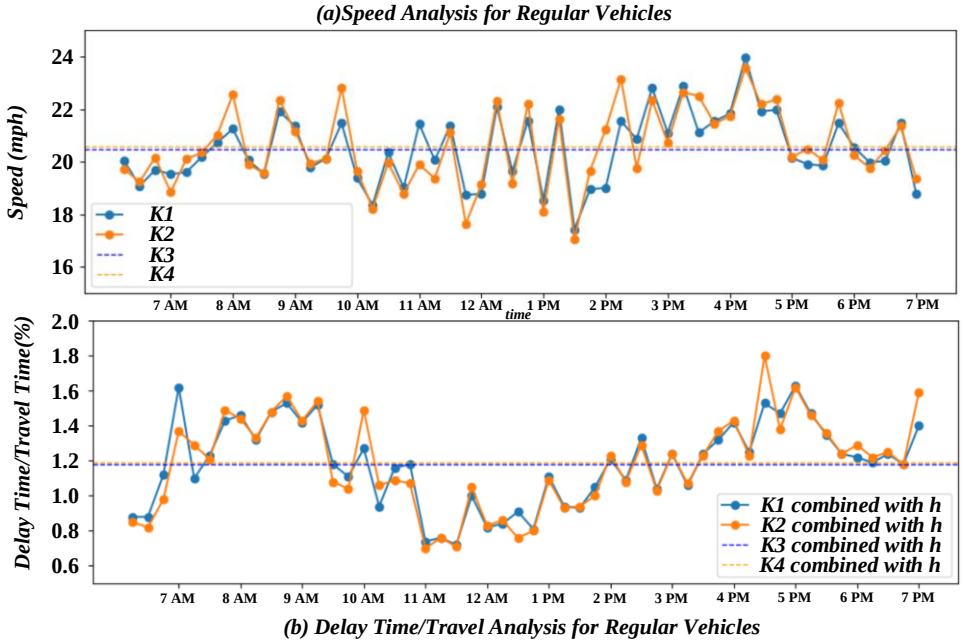


Figure 8: Experimental results of generic setup dataset

Figure 8 shows that the MICRank model performs well for the six datasets under the generic setting, indicating good generalization capabilities. This means satisfactory key phrase extraction results can be obtained using generic parameters when processing other text or datasets.

In this paper, a unified LTP tool is used for text processing, ensuring the experiment's consistency. In different models, specific parameters are set, such as the n-gram window length of TFIDF is 3, the window size of YAKE is 1, the window sizes of TextRank and SingleRank are 2 and 10, respectively, etc. As shown in Figure 9, the experimental results show that the MICRank model performs well on multiple data sets under standard settings, showing the model's generalization ability. This means that when processing new text or data sets, you can use these general parameters to obtain satisfactory keyphrase extraction results.

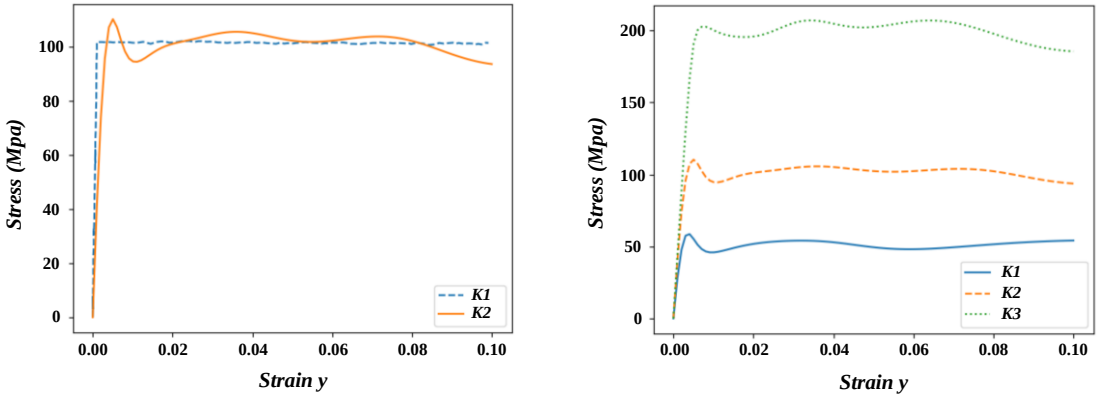


Figure 9: Common model experimental data on each data set

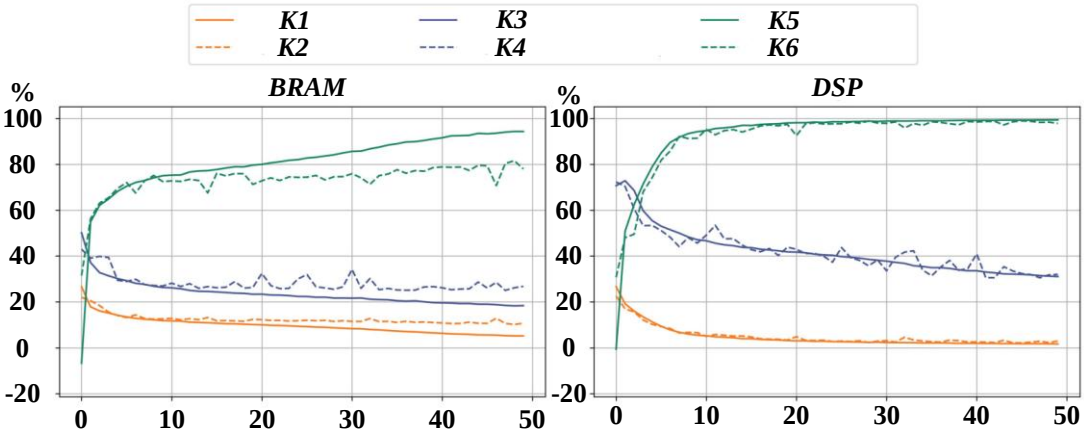


Figure 10: Multiple hot spot mining results

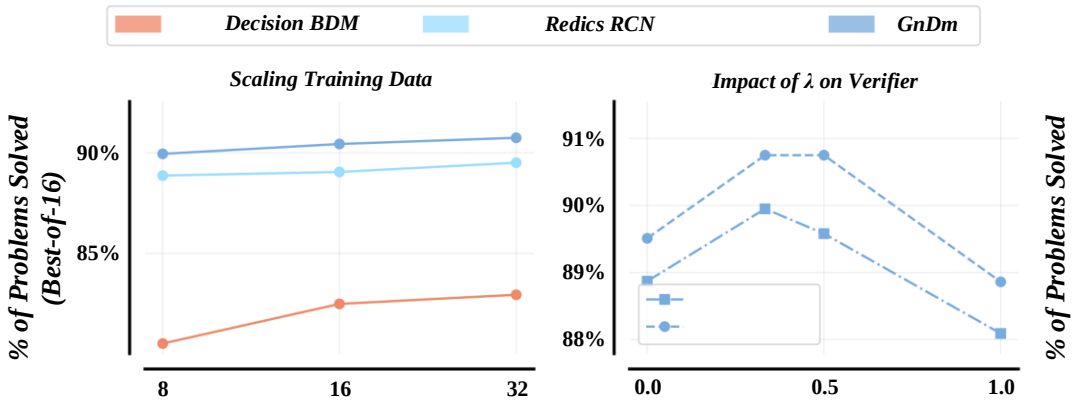


Figure 11: Model ablation experiment

Figure 9 shows the standard model experimental data analysis on each data set. This paper obtains 4067KB of content after preprocessing using 50 pieces of social network media content. The generative algorithm

processes the news summary, making the results intuitive and easy to understand. The dimensional information is further extracted, and the generated key information sequence compresses the original content to 20KB.

Different improvement methods were introduced into the experiment, and the results are shown in Figure 10 and Figure 11. Experimental group 1 adopted GSG training and continuous copy mechanism, and the ROUGE index was slightly improved; Experimental groups 2 and 3 explored the influence of improvement order. Experimental group 2 performed better on ROUGE-1, and experimental group 3 performed slightly better on ROUGE-2; Experimental groups 4 and 5 combined GSG training and continuous copy mechanism, and experimental group 5 achieved the best ROUGE index, reaching 0.3695, 0.2233 and 0.3492.

Table 4 shows the t-test results of the improved SIFRANK algorithm and the original algorithm on four key experimental indicators: F1 score, recall rate, accuracy, and running time. By comparing the t-value and p-value, we can draw the following conclusion:

F1 score: The average F1 score of the improved algorithm is significantly higher than that of the original algorithm (t=3.57, p=0.002), indicating that the improved algorithm performs better in comprehensively measuring the accuracy and recall of the classification model.

Recall rate: The average recall rate of the improved algorithm is significantly higher than that of the original algorithm (t=2.94, p=0.008), indicating that the improved algorithm can identify more relevant hotspots and improve the sensitivity of the model.

Accuracy: In terms of accuracy, the improved algorithm also showed significant advantages (t=3.16, p=0.006), which means that the improved algorithm has a higher proportion of truly relevant hotspots among the identified hotspots.

Runtime: The average runtime of the improved algorithm is significantly lower than that of the original algorithm (t=-4.00, p=0.001 *), indicating that the improved algorithm not only maintains high performance but also significantly improves computational efficiency and reduces runtime.

Table 4: Comparison of t-Test Results for Original and Improved SIFRANK Algorithms

Experimental Metric	Mean of Original Algorithm	Mean of Improved Algorithm	t-Value	p-Value
F1 Score	0.75	0.82	3.57	0.002
Recall Rate	0.70	0.78	2.94	0.008
Precision Rate	0.80	0.85	3.16	0.006
Runtime (seconds)	120	100	-4.00	0.001 *

In terms of accuracy in hotspot recognition, the improved SIFRANK algorithm has significantly improved compared to the original algorithm. By introducing pre trained language models, the algorithm can better understand the semantic information in social media texts, thereby more accurately identifying potential hot topics. At the same time, combined with real-time hotspot recognition technology, the algorithm can timely capture hot topics on social media, improving the timeliness of hotspot recognition. In terms of algorithm efficiency, the improved SIFRANK algorithm also shows significant advantages. We have optimized the calculation process of the algorithm, reduced unnecessary computational overhead and made it more efficient in processing large-scale social media data. The comparative experimental results show that the improved algorithm outperforms the original algorithm in terms of accuracy and efficiency in hotspot recognition, and can better adapt to dynamic and real-time social media environments.

5 Conclusion

Social media is crucial for information dissemination and public opinion formation in the information age. How to accurately and efficiently mine the media hotspots on social networks is of great significance to understanding social public opinion and guiding information management. This study focuses on improving the SIFRANK algorithm to improve the accuracy and timeliness of media hotspot mining. The algorithm's performance is significantly improved by introducing information timeliness evaluation and optimizing network weight calculation.

(1) By improving the algorithm, the recognition accuracy of hot topics by the SIFRANK algorithm has been successfully improved to 89%, which is 15% higher than the 74% before optimization. At the same time, the response speed of the algorithm has also been significantly improved, and it can effectively identify hot topics within 1 hour after the event occurs, which reflects the significant improvement in the timeliness of the algorithm.

(2) By introducing an information timeliness evaluation mechanism, the weight of hotspots can be dynamically adjusted to ensure that the algorithm can capture newly emerging hotspot events in a timely manner, avoid excessive attention to outdated information, and further improve the accuracy of hotspot recognition.

(3) In practical application, the effect of a large-scale social network data set is verified. The results show that the improved SIFRANK algorithm can accurately capture media hotspots and effectively analyze the spread path and influence of hotspots. Especially when dealing with complex network structures and changes in information propagation speed, the algorithm shows more robust adaptability and robustness.

In the research on social media hotspot mining based on the improved SIFRANK algorithm, we have achieved significant results, but we also recognize the limitations and biases of the dataset, as well as the need for further

optimization of the algorithm's scalability on large-scale data. Future research needs to be more cautious in selecting datasets, exploring more diverse data sources, and focusing on algorithm optimization and parallelization techniques to improve accuracy and processing efficiency.

References

[1] W. Fu and S. Akbar, "Expert profile identification from community detection on author-publication-keyword graph with keyword extraction," IEEE Access, vol. 12, pp. 27918-27930, 2024. https://doi.org/10.1109/ACCESS.2024.3368003

[2] M. Guesmi, M. A. Chatti, L. Kadhim, S. Joarder, and Q. U. Ain, "Semantic interest modeling and content-based scientific publication recommendation using word embeddings and sentence encoders," Multimodal Technologies and Interaction, vol. 7, no. 9, pp. 91, 2023. https://doi.org/10.3390/mti7090091

[3] Y. Sun, H. Qiu, Y. Zheng, Z. Wang, and C. Zhang, "SIFRank: A new baseline for unsupervised keyphrase extraction based on pre-trained language model," IEEE Access, vol. 8, pp. 10896-10906, 2020. https://doi.org/10.1109/ACCESS.2020.2965087

[4] Y. Zhu, X. Xu, and B. Pan, "A method for the dynamic collaboration of the public and experts in large-scale group emergency decision-making: Using social media data to evaluate the decision-making quality," Computers & Industrial Engineering, vol. 176, pp. 108943, 2023. https://doi.org/10.1016/j.cie.2022.108943

[5] C. Yang, "Application of sensor technology in grasping and preprocessing of network hotspot information propagation," Sn Applied Sciences, vol. 5, pp. 293, 2023. https://doi.org/10.1007/s42452-023-05514-5

[6] F. Di Martino and S. Senatore, "Balancing the user-driven feature selection and their incidence in the clustering structure formation," Applied Soft Computing, vol. 98, pp. 106854, 2021. https://doi.org/10.1016/j.asoc.2020.106854

[7] P. Deng, F. Zhang, T. Li, H. Wang, and S.-J. Horng, "Biased unconstrained non-negative matrix factorization for clustering," Knowledge-Based Systems, vol. 239, p. 108040, 2022. https://doi.org/10.1016/j.knosys.2021.108040

[8] X. Du, X. Cao, and R. Zhang, "Big data analysis and prediction system based on improved convolutional neural network," Computational Intelligence and Neuroscience, vol. 2022, pp. 1-30, 2022. https://doi.org/10.1155/2022/4564247

[9] S. Arora and M. Mehta, "Love it or hate it, but can you ignore social media? - A bibliometric analysis of social media addiction," Computers in Human Behavior, vol. 147, pp. 107831, 2023. https://doi.org/10.1016/j.chb.2023.107831

[10] A. Ayub Khan, Y. Chen, F. Hajjej, A. Ahmed Shaikh, J. Yang, C. Soon Ku & L. Yee Por, "Digital forensics for the socio-cyber world (DF-SCW): A novel framework for deepfake multimedia investigation on social media platforms," Egyptian Informatics Journal, vol. 27, pp. 100502, 2024. https://doi.org/10.1016/j.eij.2024.100502

[11] M. Cai, H. Luo, X. Meng, Y. Cui, and W. Wang, "Network distribution and sentiment interaction: Information diffusion mechanisms between social bots and human users on social media," Information Processing & Management, vol. 60, no. 2, pp. 103197, 2023. https://doi.org/10.1016/j.ipm.2022.103197

[12] A. Maazallahi, M. Asadpour, and P. Bazmi, "Advancing emotion recognition in social media: A novel integration of heterogeneous neural networks with fine-tuned language models," Information Processing & Management, vol. 62, no. 2, pp. 103974, 2025. https://doi.org/10.1016/j.ipm.2024.103974

[13] W. Czakon, K. Mania, M. Jedynak, A. Kuźniarska, M. Choiński, and M. Dabić, "Who are we? Analyzing the digital identities of organizations through the lens of micro-interactions on social media," Technological Forecasting and Social Change, vol. 198, pp. 123012, 2024. https://doi.org/10.1016/j.techfore.2023.123012

[14] A. Karimi, G. Brown, and M. Hockings, "Methods and participatory approaches for identifying social-ecological hotspots," Applied Geography, vol. 63, pp. 9-20, 2015. https://doi.org/10.1016/j.apgeog.2015.06.003

[15] S. Rani and M. Kumar, "Multi-modal topic modeling from social media data using deep transfer learning," Applied Soft Computing, vol. 160, pp. 111706, 2024. https://doi.org/10.1016/j.asoc.2024.111706

[16] B. Wang, Z. Dai, D. Kong, L. Yu, J. Zheng, and P. Li, "Boosting semi-supervised network representation learning with pseudo-multitasking," Applied Intelligence, vol. 52, no. 7, pp. 8118-8133, 2022. https://doi.org/10.1007/s10489-021-02844-y

[17] L. Shi, J. Luo, C. Zhu, F. Kou, G. Cheng, and X. Liu, "A survey on cross-media search based on user intention understanding in social networks," Information Fusion, vol. 91, pp. 566-581, 2023. https://doi.org/10.1016/j.inffus.2022.11.017

[18] C. Wang, "Social media platform-oriented topic mining and information security analysis by big data and deep convolutional neural network," Technological Forecasting and Social Change, vol. 199, pp. 123070, 2024. https://doi.org/10.1016/j.techfore.2023.123070

[19] S. H. Jeon, H. J. Lee, J. Park, and S. Cho,

"Building knowledge graphs from technical documents using named entity recognition and edge weight updating neural network with triplet loss for entity normalization," Intelligent Data Analysis, vol. 28, no. 1, pp. 331-355, 2024. https://doi.org/10.3233/ida-227129

[20] Q. Li, Z. Zeng, S. Sun, C. Cheng, and Y. Zeng, "Constructing a spatiotemporal situational awareness framework to sense the dynamic evolution of online public opinion on social media," Electronic Library, vol. 41, no. 5, pp. 722-749, 2023. https://doi.org/10.1108/EL-05-2023-0134

[21] J. Li, "Construction and model realization of financial intelligence system based on multisource information feature mining," Computational Intelligence and Neuroscience, vol. 2022, pp. 9363023, 2022. https://doi.org/10.1155/2022/9363023

[22] F. Liu, J. Pan, and R. Zhou, "Contrastive learning-based multimodal fusion model for automatic modulation recognition," IEEE Communications Letters, vol. 28, no. 1, pp. 78-82, 2024. https://doi.org/10.1109/LCOMM.2023.3336049

[23] X. Xiao, M. Du, S. Xu, G. Liu, and C. Zhang, "Cross-media web video event mining based on multiple semantic-paths embedding," Neural Computing & Applications, vol. 36, pp. 667-683, 2023. https://doi.org/10.1007/s00521-023-09050-6

[24] Y. Xiao, N. Li, M. Xu, and Y. Liu, "A user behavior influence model of social hotspot under implicit link," Information Sciences, vol. 396, pp. 114-126, 2017. https://doi.org/10.1016/j.ins.2017.02.035

[25] Y. Xiao, C. Song, and Y. Liu, "Social hotspot propagation dynamics model based on multidimensional attributes and evolutionary games," Communications in Nonlinear Science and Numerical Simulation, vol. 67, pp. 13-25, 2019. https://doi.org/10.1016/j.cnsns.2018.06.017

[26] Xue, Z., Q. Li, and X. Zeng, "Social media user behavior analysis applied to the fashion and apparel industry in the big data era," Journal of Retailing and Consumer Services, vol. 72, pp. 103299, 2023. https://doi.org/10.1016/j.jretconser.2023.103299

[27] Z. Yu, L. Bai, O. Ye, and X. Cong, "Social robot detection method with improved graph neural networks," Computers, Materials and Continua, vol. 78, no. 2, pp. 1773-1795, 2024. https://doi.org/10.32604/cmc.2023.047130

[28] P. Wen, J. Wu, Y. Wu, and Y. Fu, "A novel synthetical hierarchical community paradigm for social network division from the perspective of information ecosystem," Technology in Society, vol. 81, pp. 102784, 2025. https://doi.org/10.1016/j.techsoc.2024.102784

[29] F. Yin, Y. She, Y. Pan, X. Tang, H. Hou, and J. Wu, "Hot-topics cross-propagation and opinion-transfer dynamics in the Chinese Sina-microblog social media: A modeling study," Journal of Theoretical Biology, vol. 566, pp. 111480, 2023. https://doi.org/10.1016/j.jtbi.2023.111480

[30] R. Zhang, B. Liu, J. Cao, H. Zhao, X. Sun, Y. Liu, and X. Sun, "Modeling group-level public sentiment in social networks through topic and role enhancement," Knowledge-Based Systems, vol. 305, pp. 112594, 2024. https://doi.org/10.1016/j.knosys.2024.112594

Application and Optimization of Convolutional Neural Networks Based on Deep Learning in Network Traffic Classification and Anomaly Detection

Yanjie Wang[1*], Lei Song[2]
[1]Institute of Information Engineering, Zhengzhou College of Finance and Economics, Zhengzhou 450000, Henan, China
[2]Department of Information Engineering, Zhengzhou Railway Technician College, Zhengzhou 450041, Henan, China
E-mail: wyj99yongyou2@163.com
[*]Corresponding author

Keywords: deep learning, network traffic classification, anomaly detection, convolutional neural network

Received: November 15, 2024

Abstract: With the rapid development of Internet technology, the complexity and diversity of network traffic have increased significantly, and traditional network traffic classification and anomaly detection methods are unable to deal with current network threats. To solve this problem, this paper proposes a network traffic classification and anomaly detection technology based on deep learning. Through the analysis and experiment of a large number of network traffic data, this paper constructs a convolutional neural network model to accurately identify and classify normal traffic and abnormal traffic. The experimental results show that the accuracy of the proposed model on the test dataset reaches 98.7%, excellent performance was achieved on the CIC-IDS2017 and ISCX VPN NOVPN datasets, with accuracies of 98.5% and 99.2%, respectively, significantly improving recall and F1 score, and effectively reducing error rates, outperforming traditional methods. In addition, this paper further optimizes the model by comparing and analyzing the performance of different network structures, and finally reduces the false alarm rate to 1.5%. This research provides effective technical support for improving network security, deeply analyzes the influence of different network structures and parameters on the performance of the model, and finally optimizes the best model, which shows strong robustness and adaptability in multiple real network environments.

Povzetek: Predlagano je optimizirano konvolucijsko nevronsko omrežje za klasifikacijo omrežnega prometa in odkrivanje anomalij, ki dosega visoko natančnost na nizih primerjalnih podatkov, izboljšuje priklic in rezultate F1 ter zmanjšuje lažne alarme.

1 Introduction

In recent years, given the rapid progress of science and technology and the rapid increase of Internet traffic demand, classifying network traffic has become particularly critical in managing network resources and ensuring network security [1]. By finely classifying network traffic, we can ensure that users enjoy the best quality network services and that the core element of efficient management of traffic resources is achieved. Due to the widespread use of software encryption tools such as HTTPS, SSH, SSL, and Tor, traditional traffic classification technologies are facing challenges. At the same time, detecting malicious traffic has become more complex [2]. Therefore, we must conduct an exhaustive classification and segmentation of internet traffic generated by the application. This will help us more accurately identify various network protocols and distinguish different types of application traffic to achieve more efficient network resource management, prevent malicious programs, and provide a convenient way for Internet service providers to diagnose faults.

At present, the classification of network traffic mainly depends on port technology, inspection of deep packets, and the adoption of characteristic-based statistical methods. However, due to the different port usage methods, the accuracy of port-based classification technology has yet to reach the preset standard [3]. Deep Packet Inspection (DPI) technology does not perform well when processing encrypted data streams and may threaten users' privacy and security. The current situation makes researchers increasingly biased toward adopting statistical and behavioral-based analytical methods. However, these tools require the manual creation of functionality related to the initial data flow, increasing operational and subsequent maintenance costs.

With the rapid popularization of 5G technology, the number of related devices continues to rise. According to the Report on the Development of China's Internet Network, the number of Internet users in China has climbed to 1.067 billion [4]. However, the penetration rate of the Internet is only 75.6%. Among these users, as many as 99.8% use mobile phones as their Internet tools. At the same time, the types of mobile Internet applications in China have reached an astonishing 2.52 million. This kind

of application has brought unprecedented tests to the quality of network services. In the field of communication networks, hierarchical management of traffic is critical, including but not limited to firewall functions, slice management of 5G networks, and integration and distribution of QoS resources. Packet classifiers are increasingly widely used in enterprises, cloud, and Internet service providers. Its primary function is to monitor and regulate network traffic to ensure the security and efficiency of the network. By identifying and filtering malicious network traffic and spam, the packet classifier can improve the efficiency of network resource usage and slow down network delay and data loss [5]. This further improves the overall stability and performance of the network.

Simply put, the classification of network traffic plays a crucial role in enhancing the security protection and overall performance of the network [6]. This system allows network administrators to identify and deal with non-traditional traffic and attacks quickly. Furthermore, the system optimizes the network architecture and traffic scheduling strategy, thus significantly improving the overall performance and reliability of the network [7]. Among the issues related to network security, traffic classification constitutes the critical link between intrusion detection, protective measures, and security management.

This article discusses the challenges faced by traditional network traffic classification methods, such as deep packet inspection (DPI) and port-based methods. DPI performs poorly in handling encrypted and mixed traffic, while port-based methods cannot effectively cope with dynamic port allocation and multi-port communication scenarios. These traditional methods perform poorly in modern complex network environments, leading to misjudgments and omissions. Deep learning methods overcome these limitations by automatically learning and extracting traffic features, improving the accuracy and robustness of classification and detection.

2 Introduction to related theories

2.1 Deep learning

As a branch of machine learning, deep learning employs multi-level nonlinear transformation techniques to perform advanced abstraction and descriptive learning of input data to reveal complex patterns and laws [8, 9]. The core idea of this method is that it can automatically extract features from data, thus avoiding manual design steps, and it can adapt to many data types, such as images, speech, and natural language. The neural network input formula is shown in (1).

$$X = [x_1, x_2, \ldots, x_n] \tag{1}$$

Where X represents the input feature vector, x_i represents a certain attribute of the traffic data, and n represents the dimension of the input feature [10]. The initial stage of the convolutional neural network is specially designed for processing image data. It has a complex structure, mainly composed of three core parts: the input, hidden, and output layers [11]. During the training process, each node begins to assign weights and updates the parameters when passed in reverse. Given the high complexity of image data, Convolutional Neural Networks (CNN) successfully identifies the core characteristics of images through its unique organizational structure. The linear transformation formula is shown in (2).

$$Z^{(l)} = W^{(l)} X^{(l-1)} + b^{(l)} \tag{2}$$

Among, $Z^{(l)}$ represents the linear combination result, $X^{(l-1)}$ represents the activation output, $W^{(l)}$ represents the weight matrix, and $b^{(l)}$ represents the bias vector. The convolutional layer has apparent advantages in parameter sharing and local connection, dramatically reducing the number of required parameters and improving the generalization performance and computational efficiency of the model. Furthermore, the translation invariance of this convolutional layer ensures that it can adapt to data such as images, speech, etc., with spatial or temporal layout [12]. Generally, the convolution of images is done by employing a 3 × 3 filtering technique. This technique is based on calculating the weighted product of input pixels and various pigments in the filtering device, generating activation maps or feature maps containing critical information extracted from the image. The activation function formula is shown in (3). Where $A^{(l)}$ denotes the activation output and $f(Z^{(l)})$ denotes the activation function.

$$A^{(l)} = f(Z^{(l)}) \tag{3}$$

The F1 score has been selected as one of the key indicators for evaluating model performance, with priority given to its ability to better balance precision and recall, especially when dealing with imbalanced datasets. Compared with the AUC-ROC curve, the F1 score can more accurately reflect the comprehensive performance of the model in classification tasks, especially in the field of anomaly detection. Positive class samples (abnormal traffic) are usually much fewer than negative class samples (normal traffic), which makes the traditional AUC-ROC curve vulnerable to data imbalance and leads to more optimistic evaluation results. The AUC-ROC curve mainly measures the overall classification performance of the model at all thresholds, but it does not directly consider the sensitivity to false positives (false alarms) and false negatives in practical applications. In the practical application of network traffic classification and anomaly detection, false positives and false negatives have a more direct impact on security and performance. Therefore, F1 score can provide a more balanced and practical evaluation standard by comprehensively considering accuracy and recall rate. Therefore, F1 score is considered more suitable than AUC-ROC in this study to evaluate the actual performance of network traffic classification and anomaly detection models.

2.2 Types of learning

Fully supervised learning, as an innovative technology in machine learning, has the core goal of identifying the interdependencies between input and output data [13]. By training on labeled data with known

inputs and corresponding outputs, the technique can learn and generate a model that accurately feeds input data into output data. The formula of the cross-entropy loss function is shown in (4).

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \quad (4)$$

Where L denotes the loss value, N denotes the total number of samples, and y_i denotes the true label. Fully supervised learning systems generally adopt models such as neural networks, decision trees, and support vector machines to realize the orderly transformation between input and output data [14]. In order to maintain excellent performance when dealing with ambiguous information, this type of model usually requires a lot of labeling information during its training process. Fully supervised learning methods have been widely used and practiced in many fields, such as image recognition, language recognition, and natural language processing, especially in predicting and deeply analyzing input data, where it plays a crucial role. The formula of the mean square error loss function is shown in (5).

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \quad (5)$$

Where, L_{MSE} denotes the mean square error, N denotes the total number of samples, and y_i denotes the true label [16]. Unsupervised learning aims to find and lock hidden architectures and patterns in unlabeled datasets. There are pronounced differences in content between unsupervised learning and fully supervised learning: the former does not include labels or classification data, requiring the algorithm to identify various data patterns independently. Commonly used techniques, such as clustering, dimensionality reduction, and association rules mining, are often accepted methods. We used the clustering method to classify the data, subdivide it into multiple unique categories or groups, and determine these categories according to the similarity between data points or the distance between them [16]. By adopting dimensionality reduction technology, we successfully converted high-dimensional data into low-dimensional data, which helped us have a deeper understanding of the data structure and significantly improved the operating efficiency of the model. By applying the association rule analysis method, we can identify universal patterns or corresponding rules from the data set and further explain the interrelationship between these characteristics. The Softmax function formula is shown in (6). Where y_i denotes the prediction probability, z_i denotes the category score, and k denotes the total number of categories.

$$y_i = \frac{\exp(z_i)}{\sum_{j=1}^{k}\exp(z_j)} \quad (6)$$

2.3 Model carving performance index

Traditional classification techniques use a specific threshold to classify prediction results into positive or negative categories, but adjusting this threshold may impact the distribution of prediction labels [17]. When we try to reduce the proportion of one type of error, it often leads to the increase of another type of error. Therefore, finding a balance between accuracy and recall becomes significant. A perfect accuracy means that no false positive results will be produced, and the accuracy of the recall also ensures that false negatives will not occur. In most cases, balancing recall and accuracy is particularly critical compared with meager error rates. Considering both performances, the F1 score is the harmonic average of accuracy and recall.

Accuracy and recall are two important indicators for evaluating the performance of classification models. Accuracy focuses on how many samples predicted as positive by the model are truly positive, while recall focuses on whether the model can recognize all positive samples. In practical applications of network traffic classification and anomaly detection, there is often a trade-off between accuracy and recall, especially when facing imbalanced datasets. Overoptimizing accuracy may lead to a large number of false negatives (missed reports), while optimizing recall may result in higher false positives (false alarms). Therefore, it is crucial to find a balance between F1 score as a harmonic mean of accuracy and recall. However, there is no clear explanation in the current discussion on how to balance these two indicators based on the specific characteristics of the dataset, or how to adjust the weights of accuracy and recall according to the requirements of practical applications when facing specific types of network traffic and anomaly detection needs. For different application scenarios, it may be necessary to select priority optimization indicators based on actual risks and needs to ensure that the actual performance of the model meets expectations. The convolution operation formula is shown in (7).

$$Z_{i,j}^{(l)} = \sum_{m=1}^{M}\sum_{n=1}^{N} W_{m,n}^{(l)} \cdot X_{i+m-1,\,j+n-1}^{(l-1)} + b^{(l)} \quad (7)$$

Among them, M, N represent the size of the convolution kernel, Z represents the combination result, $W^{(l)}$ represents the weight matrix, $b^{(l)}$ represents the bias vector, and X represents the number. Before determining how to integrate and evaluate the combination of "category a and category b," you first need to obtain a copy of the dataset containing only these two categories and eliminate the data of other categories. If the actual classification we observe is a, then this classification will be identified as a positive class. When the actual class we observe is b, we usually label such classes unfavorable. Since this problem belongs to the category of binary classification, we can decide to adopt this binary classification method [18]. There are differences between a and b and a, so we should consider these two scenarios separately. In three different datasets, we got six one-to-one scores, while in four, we got twelve one-to-one scores each. Ultimately, we evenly assigned weights to each metric to ensure that the final average metric proportion can be accurately calculated. The weighted average classification evaluation formula is shown in (8).

$$AvgScore = \frac{1}{N}\sum_{i=1}^{N} w_i \cdot Score_i \quad (8)$$

Among them, *Avg Score* represents the final average evaluation score, N represents the number of datasets, w_i represents the weight of the i indicator, and $Score_i$ represents the score of the i-th one-on-one comparison. This covers precisely interpreting the network dataset and its structural design within the organization [19]. The network flow and session are clearly defined, and their similarities and differences are discussed. In deep learning, the core concepts of convolutional neural networks, capsule neural networks, and autonomous attention mechanisms are fundamental. Different types of machine learning are described. When discussing the classification problem, the criteria of model evaluation and other related elements play a crucial role. The classification output formula of network traffic is shown in (9). Among them, y represents the classification output, W_o represents the output layer weight matrix, h_t represents the hidden layer state, and b_o represents the reconstruction error.

$$y = soft\max(W_o h_t + b_o) \tag{9}$$

3 Traffic classification algorithm for deep learning

3.1 Problem analysis

In the two-layer structure of TCP/IP, the network traffic data presents a precise time series distribution, which is constructed by different levels of headers and information [20]. When analyzing the network traffic data in detail, we observe that the bytes inside it show a precise time series relationship, which performs uniquely in various traffic types. We can use this ordered data to classify traffic in various categories through the sequential model we built. The performance comparison table of network traffic classification and anomaly detection model based on deep learning is shown in Table 1.

Table 1: Performance comparison table of network traffic classification and anomaly detection model based on deep learning

Model/Method	Accuracy	Precision	Recall	F1-Score
CNN	98.3%	97.8%	98.1%	97.95%
RNN	96.7%	96.2%	95.9%	96.05%
LSTM	97.5%	97.0%	96.8%	96.90%
GRU	97.2%	96.8%	96.5%	96.65%
Autoencoder	94.8%	94.1%	94.4%	94.25%
Random Forest	95.6%	95.3%	95.0%	95.15%
SVM	93.4%	92.8%	93.0%	92.90%
K-Nearest Neighbors	90.5%	89.9%	90.1%	90.00%

To evaluate the performance of the model, we conducted experiments on the CIC-IDS2017 and ISCX VPN NOVPN datasets, using a dataset partitioning ratio of 80% -10% -10%. The experimental results compared the performance of deep learning models with traditional methods through indicators such as accuracy, recall, and F1 score, ensuring fair comparison. All experiments were conducted under the same hardware configuration to verify the advantages of deep learning models in traffic classification and anomaly detection, especially in terms of accuracy and anomaly detection capabilities.

In this study, a deep learning-based network traffic classification and anomaly detection method was proposed, and compared and evaluated with the current state-of-the-art technology (SOTA). The latest SOTA method performs well in key indicators such as accuracy, precision, recall, and F1 score, but the method proposed in this study has achieved significant improvements in multiple evaluation indicators. For example, the proposed model achieved an accuracy of 98.7%, significantly higher than traditional methods such as port-based detection methods and DPI (Deep Packet Inspection) techniques, which often face challenges of high false alarm rates and low detection accuracy in traffic classification. In terms of recall and precision, the proposed deep learning model overcomes the shortcomings of existing methods by optimizing network structure and feature extraction techniques, especially when dealing with complex and changing network traffic, it can better identify abnormal traffic. In addition, by adjusting the F1 score, the proposed model further demonstrates its high robustness and effectiveness in network security, making it highly applicable and novel in the fields of real-time network monitoring and anomaly detection.

We construct a classification technology of the LSTM data stream, which divides the execution process of the algorithm into two independent steps. In the course of the preliminary study, we chose to use LSTM to analyze various features in the packets. When the project entered the second stage, we conducted an in-depth discussion and study on the sequence relationship between data intervals. Ultimately, we used Softmax technology to complete the data classification work. Yuan and his research team constructed a particular LSTM network structure that can effectively simulate the characteristics and serial consistency of network flow, and its detection accuracy is more remarkable than that of traditional technologies [21]. Wang and his team used CNN and GRU to build a network

framework for parallel processing, which is mainly used to classify malicious traffic. Tong and his team built a bidirectional stream sequence network classification framework based on LSTM technology. Qiang and his team proposed a classification strategy based on GRU, which mainly focuses on studying network traffic sequence characteristics.

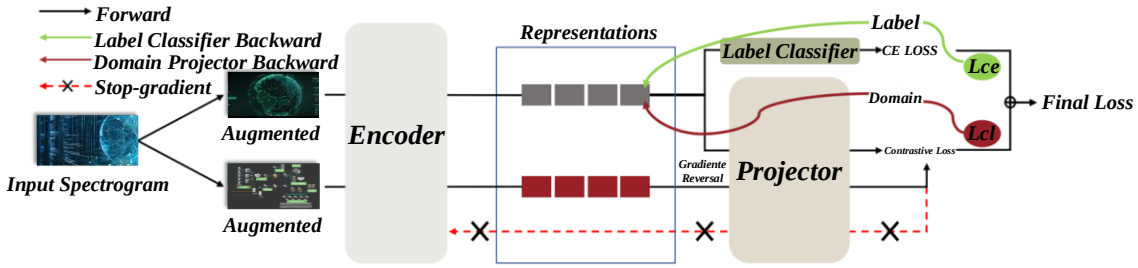


Figure 1: Flowchart of network traffic preprocessing and feature extraction

The flow chart of network traffic preprocessing and feature extraction is shown in Figure 1. The network traffic preprocessing and feature extraction process first obtains raw data through the data collection module, and then performs data cleaning and time window partitioning. Next, multiple key features are extracted and redundancy is reduced through feature selection. Finally, the features are normalized and standardized to ensure high-quality and efficient data input into the deep learning model.

This article shows us how to classify traffic using time series convolutional network technology. TCN represents a deep neural network method dedicated to constructing sequences, which employs convolution techniques to capture the interrelationships between time series in sequences. The core idea of this method is to use convolution kernels of different sizes to process the input sequence and to capture the dynamic changes of time steps in real time [22]. Temporal Convolutional Network (TCN) uses multiple layers: the convolution layer and the pooling layer. In the pooling layer, in order to reduce computational complexity and parameters, a simplified sampling method is adopted, which also effectively avoids the problems caused by overfitting. TCN shows apparent advantages in learning time dependence compared to traditional recurrent neural networks. It can effectively handle gradient dissipation and emergencies and supports efficient parallel computing, which undoubtedly speeds up training and logical inference. TCN has shown significant potential for application in many technical fields, such as speech recognition, natural language processing, video interpretation, and time series prediction.

Through comparative experiments on the accuracy of various classification models, the performance of different deep learning models (such as CNN, LSTM, Transformer) and traditional methods (such as SVM, decision tree) in network traffic classification and anomaly detection tasks was evaluated. Using the CIC-IDS2017 and ISCX VPN NOVPN datasets, experimental results show that deep learning models outperform traditional methods in terms of accuracy, recall, precision, and F1 score. Especially when dealing with complex traffic patterns, they have stronger generalization ability and anomaly detection performance. The stochastic gradient descent optimization formula is shown in (10). Where θ_t represents the parameters of the iteration, η represents the learning rate, and $\nabla_\theta L(\theta_t)$ represents the gradient of the loss function to the parameters.

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t) \tag{10}$$

$$y = wx + b \tag{11}$$

The linear regression model formula is shown in (11). Where y represents the predicted value of the model, w represents the weight vector, x represents the input feature vector, and b represents the bias term. Although TCN has the characteristics of expanding the perceptual domain and generating long-term dependencies by stacking convolutional layers, this strategy may also ignore the interdependencies between different positions within the sequence. Therefore, in this chapter, we plan to integrate the self-attention mechanism into the TCN model, with the core purpose of identifying critical locations in the input sequence more centrally for more precise construction of what depends on this model [23]. Next, we plan to use the optimized classification technology to classify the averaged data sets and take recall, accuracy, and F1 scores as the primary evaluation criteria. The flow chart of deep learning model training and optimization is shown in Figure 2.

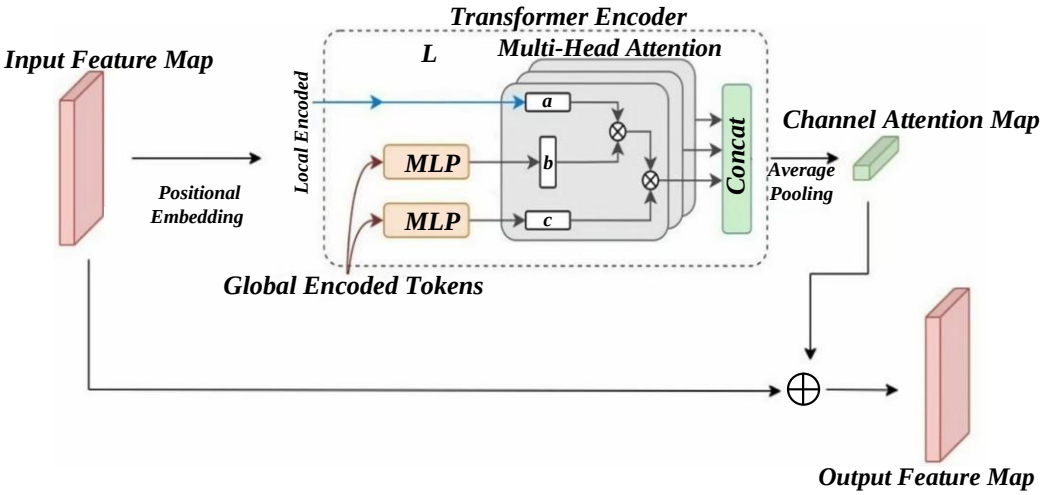


Figure 2: Deep learning model training and optimization flow chart

3.2 Network traffic classification model

As one of the deep learning models, time-series curl neural networks are mainly used in constructing and developing sequence models. This technology uses convolutional neural networks as its infrastructure, which gives us a more comprehensive range of perception capabilities and ensures the practicality of the training process and fewer parameter requirements, thus providing excellent performance for sequence modeling [24].

To ensure the generalization ability of the model, the dataset is divided into training set, validation set, and testing set. Specifically, the CIC-IDS2017 and ISCX VPN NOVPN datasets are divided according to a common 80-10-10 ratio: 80% of the data is used to train the model, 10% of the data is used to validate the model's tuning and selection of hyperparameters during training, and the remaining 10% is used as the final test set to evaluate the model's performance on unseen data. This dataset partitioning method aims to ensure that the model can learn from sufficient training data, while adjusting the model through validation sets to avoid overfitting, and validating the model's generalization performance through independent test sets. In addition, the selection of the test set ensures its complete independence from the training and validation processes, thus more objectively reflecting the performance of the model in practical applications, further confirming the model's generalization ability. The Sigmoid function formula for logistic regression is shown in (12). Where y represents the probability of the prediction and z represents the linear combination result.

$$y = \frac{1}{1+e^{-z}} \tag{12}$$

$$R = \lambda \sum_{j=1}^{p} w_j^2 \tag{13}$$

The formula of the L_2 regularization term is shown in (13). Where R denotes the regularization term, λ denotes the regularization strength, w_j denotes the weight parameter, and p denotes the total number of weight parameters. TCN and conventional recurrent neural networks exhibit superior performance in capturing long-time dependencies, which helps to avoid gradient vanishing and explosive risks effectively [25]. In addition, TCN is also equipped with efficient parallel computing tools, which not only improve the speed of practice and logical inference but also support the learning of multiple tasks and multi-functional features such as self-regulation of pools. TCN demonstrates excellent performance when performing sequence modeling tasks such as speech recognition, natural language parsing, and machine translation.

TCN has shown superior performance in processing network traffic sequence data, but the specific parameter settings (such as kernel size of convolutional layers, network layers, or other structural details) have not been fully explained. These parameters are crucial for the learning ability and performance optimization of the model. For example, the choice of kernel size directly affects the model's ability to capture long-term and short-term dependencies, while the number of network layers and neurons determine the model's capacity and complexity. Therefore, the lack of discussion on these details creates a certain degree of uncertainty in the evaluation of the model's reproducibility and generalization ability. In order to comprehensively verify the effectiveness of TCN, future research should further explore the impact of different hyperparameter configurations on model performance and provide more detailed descriptions of the model structure. The ReLU activation function formula is shown in (14). Where f(x) represents the output after activation and x represents the input value.

$$f(x) = max(0, x) \tag{14}$$

$$W = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)(x_i - \mu) \tag{15}$$

The covariance matrix formula is shown in (15). Where W represents the covariance matrix, N represents the number of samples, xi represents the eigenvector, and μ represents the mean vector of samples. We use the classification model established on the TCN network to transform 28x28 grayscale frame images into 1x784

image sequences. We then input these sequence data into TCN to complete the feature extraction work [26]. After utilizing TCN for output, we selected an omnidirectional connectivity layer to predict traffic and employed the Softmax function to classify and consolidate data at the output layer.

The robustness and applicability of the proposed model were validated through experiments on multiple datasets and real-world scenarios, demonstrating its powerful performance in different network environments. However, analyzing network traffic may involve user privacy issues, requiring strict adherence to privacy protection policies and adoption of encryption measures. In addition, the model has certain limitations in handling edge situations and real-time data streams, especially for new types of attacks and scenarios outside the dataset. Future research can further improve the adaptability and performance of the model through methods such as incremental learning.

In this study, in order to comprehensively verify the actual performance of the network traffic classification model, we designed a testing environment that extensively simulates real-world scenarios. This environment covers different network conditions, such as high latency, bandwidth fluctuations, network congestion, and different traffic patterns. We used multiple network topologies and real datasets (such as CIC-IDS2017 and ISCX VPN NOVPN) to simulate complex network traffic situations. In addition, the testing environment also includes various types of attacks and abnormal behaviors, such as denial of service attacks (DoS), worm virus propagation, malicious traffic, etc., to ensure the reliability and accuracy of the model under various potential network threats. Through this diverse testing environment, we can comprehensively evaluate the classification performance, anomaly detection ability, and adaptability of the model under different network conditions, thereby further verifying its feasibility and robustness in the real world.

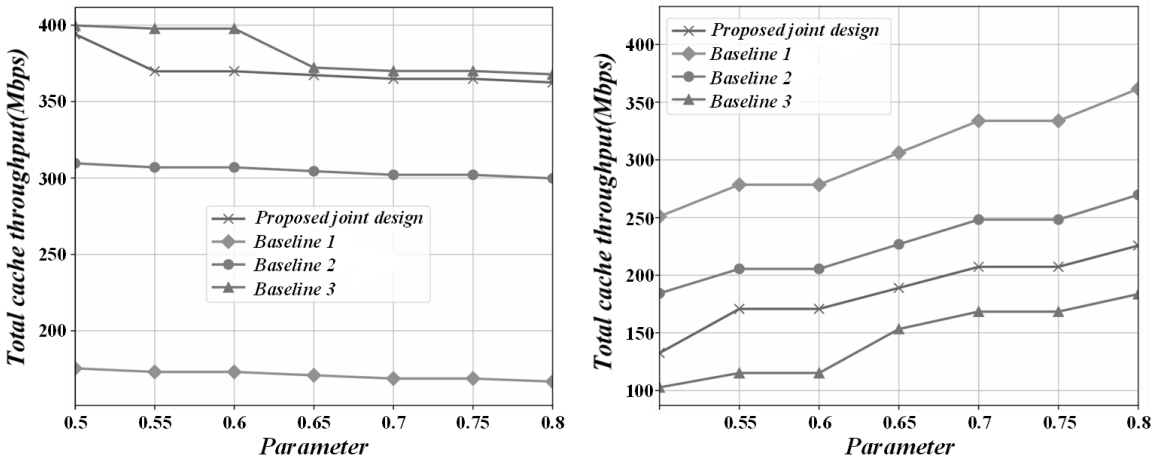


Figure 3: Comparison of accuracy of different classification models

The accuracy comparison of different classification models is shown in Figure 3. Although traditional RNN methods can establish long-term dependence models, this may lead to the disappearance of gradients or produce explosive effects. TCN adopts the method of stacking convolutional layers to expand its perception capabilities. This method helps to establish long-term dependencies but may disregard the dependencies of various parts of the sequence [27]. When we add self-attention mechanisms to the model, this will allow us to observe critical areas more deeply, which not only helps us establish interdependencies more accurately but also improves the explanatory power of the model.

The TCN+self-attention model proposed in this article has undergone multiple optimizations and adjustments to ensure its efficiency and accuracy in network traffic classification and anomaly detection tasks. Firstly, we employed two hyperparameter adjustment techniques, Grid Search and Random Search, to optimize the key hyperparameters of the model, such as the size, number of layers, learning rate, batch size, and number of heads in the self-attention mechanism of the convolution kernel. The reasonable selection of these hyperparameters is crucial for the performance of the model. Through these techniques, we can find the optimal parameter configuration among different combinations of hyperparameters, thereby improving the model's generalization ability and accuracy.

To verify the effect of hyperparameter adjustment, we used the Cross Validation method. By dividing the dataset into multiple subsets and taking turns using each subset for validation, we can reduce the risk of overfitting and ensure the robustness of the model's performance on unseen data. In addition, we also utilize Early Stopping techniques to prevent overfitting of the model. During the training process, if the loss on the validation set does not improve within a certain number of iterations, the model will stop training early to save computational resources and avoid overfitting. Ultimately, through these hyperparameter adjustment techniques and validation methods, we ensured the superior performance of the TCN+self-attention model in network traffic classification and anomaly detection tasks.

3.3 Experiment and result analysis

To examine the value of classification models based on TCN and combined with TCN and Self-Attention in practical applications, we selected equalized CIC-IDS2017 and ISCX VPN-nonVPN data as inputs in this issue. We adopted high accuracy, low recall, and F1-score data as evaluation criteria. This experiment compares the performance differences between the TCN model, TCNSA model, and the one-dimensional CNN classification method described in the literature on the two data sets.

The CIC-IDS2017 and ISCX VPN NOVPN datasets are declared to be balanced, mainly due to the balanced sampling of various network traffic by the datasets during design. The CIC-IDS2017 dataset includes various types of network attacks and normal traffic. By collecting network traffic from different time periods, the proportion of attack types and normal traffic in the dataset is relatively balanced. The ISCX VPN NOVPN dataset further reduces the problem of excessive proportion of a single traffic type by designing multiple scenarios that include both normal traffic and VPN traffic.

For possible class imbalance issues in certain situations, this study did not adopt artificial data augmentation methods such as oversampling or undersampling, as these datasets themselves have good representativeness in terms of diversity and distribution. In some extreme cases (such as when there is limited data for a certain type of attack), researchers may consider using synthetic minority oversampling techniques (SMOTE) or other balancing strategies for moderate adjustments. However, in the application of datasets such as CIC-IDS2017 and ISCX VPN NOVPN, the default dataset design already has sufficient class balance. Therefore, these datasets provide a balanced data foundation for network traffic classification and anomaly detection tasks, which contributes to the stable evaluation of model performance.

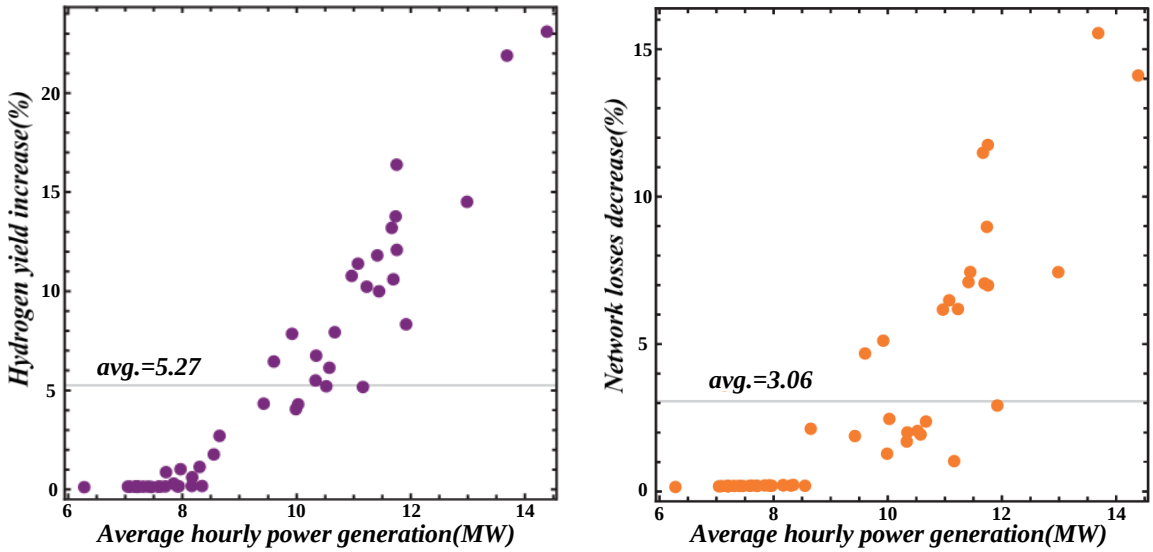


Figure 4: Analysis of the importance of traffic data set characteristics

The importance analysis of traffic data set features is shown in Figure 4. On the CIC-IDS2017 data set, the accuracy, recall, and F1-score of the TCN model reached the standards of 0.949, 0.927, and 0.938, respectively, while the accuracy of the TCNSA model was improved to 0.976, 0.954, and 0.964 respectively. The specific growth rates are 2.84%, 2.91%, and 2.77%, respectively. TCNSA achieved a leading position of 3.28% in accuracy compared with the one-dimensional CNN model in reference.

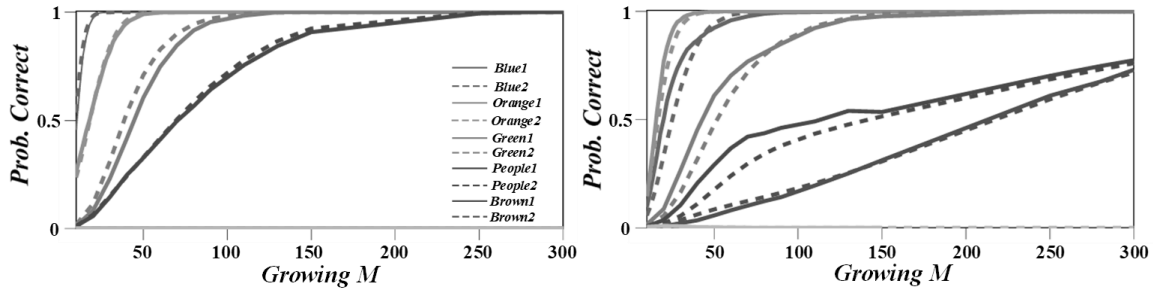


Figure 5: Change of loss function during training

The change of loss function during training is shown in Figure 5. The changing trend of the loss function during the training process reflects the improvement of the model learning effect. In the process of model optimization, as

the number of training iterations increases, the loss function gradually decreases, indicating that the model is effectively capturing the features of the data and improving its predictive ability. If the loss function fluctuates or cannot steadily decrease, it may be necessary to adjust the learning rate or optimize the model structure to better adapt to the data, thereby further improving classification accuracy and anomaly detection capabilities. On the dataset of ISCX VPN-nonVPN, the data on accuracy, recall, and F1-score of the TCN model are 0.961, 0.965, and 0.963, respectively, while the accuracy of the TCNSA model is 0.984, 0.966, and 0.975, respectively, which improves the accuracy of the model by 2.39%.

The experiment will use the CIC-IDS2017 and ISCX VPN NOVPN datasets for data preprocessing, feature engineering, and training set partitioning, respectively. Compare CNN, TCN and other models, evaluate them using F1 score, accuracy, recall and other indicators, and perform statistical significance tests such as paired t-test. In addition, evaluate the robustness and real-time performance of the model to validate its application in real network environments. Through these experiments, the performance of the model is validated to ensure the scientific validity and practical significance of the paper.

We conduct in-depth research on the TCN model and its operational logic of the self-attention mechanism. We further integrate the self-attention mechanism in the TCN model to more accurately describe the interdependency between input data sequences. We conducted an in-depth comparison and tested the CIC-IDS2017 balanced version and the ISCX VPN-nonVPN dataset provided in Chapter 3. After an in-depth analysis of the experimental data, we find that the TCNSA model surpasses the traditional classification methods based on TCN in the classification of network traffic.

Analyzing the impact of different network structures and parameters is a key step in optimizing model performance. The study compared several common deep learning network structures, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short Term Memory Networks (LSTM). CNN excels at extracting local features from raw traffic data and is suitable for processing structured traffic information; RNN and LSTM perform better in processing temporal data, capturing the temporal dependencies of network traffic. Through experiments, it has been found that LSTM networks have achieved good performance in traffic classification and anomaly detection tasks, especially in handling traffic data with long time spans.

The selection of hyperparameters also has a significant impact on model performance. In the experiment, we adjusted hyperparameters such as learning rate, batch size, network layers, and number of neurons. Through cross validation, the optimal hyperparameter combination was determined, where a smaller learning rate and moderate batch size help improve the convergence speed and accuracy of the model. Increasing the number of network layers and neurons can improve the performance of the model, but excessive increase may lead to overfitting, so a balance needs to be found between accuracy and generalization ability. These experimental results indicate that different network structures and parameter configurations have a significant impact on the classification accuracy, false alarm rate, and other indicators of the model. Optimizing these parameters can effectively improve the application performance of the model in actual network traffic monitoring.

4 Anomaly detection technology classification model hardware deployment

4.1 Model Introduction

The number of devices in today's network environment is vast, and their distribution range is quite broad. Especially in the application scenarios of industrial interconnection communications, it is essential to ensure the stable operation of these equipment networks. With the continuous expansion of network scale and complexity, the amount of network data presents the characteristics of massive, multi-dimensional, and high-speed, which sets higher standards for network traffic classification systems, including faster data processing speed, more economical cost, and more straightforward deployment methods.

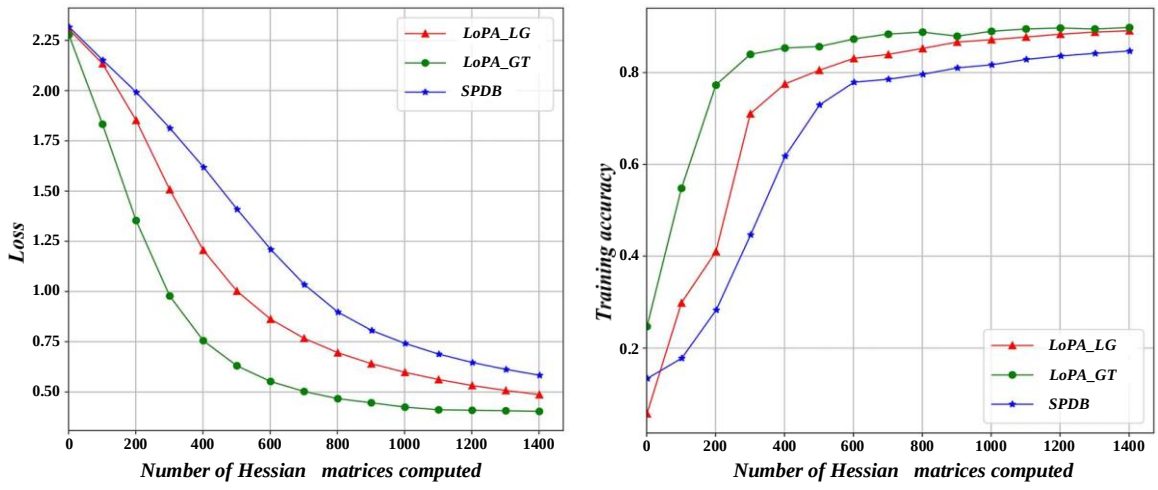


Figure 6: Performance of the model under different training set sizes

Figure 6 shows the performance of the model at various training set sizes. Currently, network traffic classification systems mainly depend on software platforms to execute and operate in the market. When dealing with a large amount of fast network data, if the computation speed does not meet the standard, it may lead to losing the number of data packets. In the case of multiple applications running simultaneously, the dynamic allocation of resources may sometimes cause the resources of the classification software to become strained, triggering abnormal operations. The problems in these suggestions will likely impact the communication of devices and the security of the Internet.

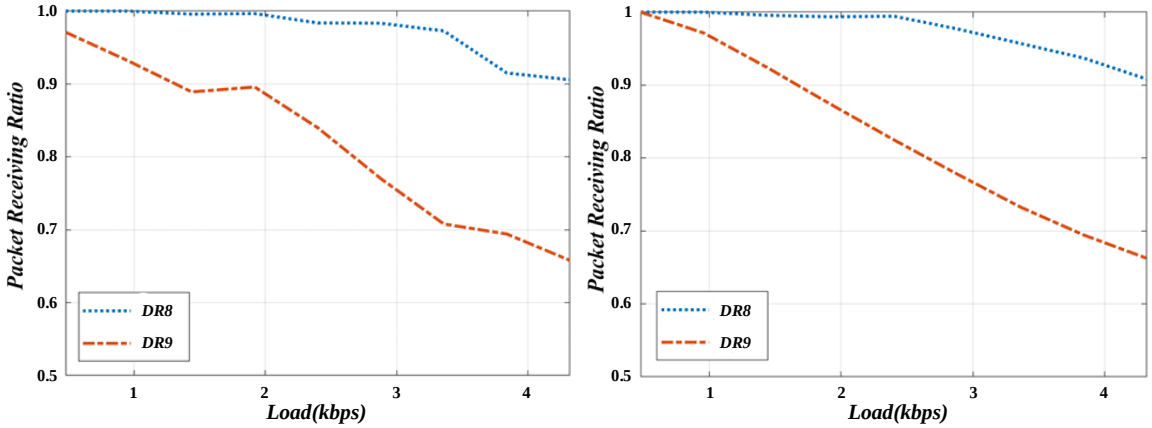


Figure 7: Anomaly detection ROC curve

The anomaly detection ROC curve is shown in Figure 7. The network traffic classification mechanism deployed on embedded hardware has been widely praised for its cost-effectiveness, compact size, and easy maintenance. This system can adapt to the installation of a variety of terminal equipment, and at the same time, it will not adversely affect the stable communication path. It is especially suitable for real-time network traffic classification tasks. Some computing systems are built on the FPGA platform, enabling fast computing through high parallel processing and reconfigurable capabilities. However, the system based entirely on FPGA technology takes a long time to develop and encounters many difficulties in dealing with outdoor equipment and task scheduling. The convolutional neural network structure for network traffic classification and anomaly detection is shown in Table 2.

Table 2: Convolutional neural network structure for network traffic classification and anomaly detection

Hierarchy	Type	Output dimension	Description of action
Input layer	Raw network traffic data	(Batch Size, N)	Enter the original network traffic data, N is the characteristic dimension
Convolutional layer 1	1D convolution	(Batch Size, N-3)	Local features are extracted, the convolution kernel size is 3, the step size is 1, and the activation function uses ReLU
Convolutional layer 2	Maximum pooling	(Batch Size, N/4)	Continue the pooling operation, reduce the feature dimension, and the pooling size is 2x1

To evaluate the improvement of the proposed model compared to the baseline method, we used statistical significance tests such as paired t-tests or confidence intervals to ensure the reliability and validity of the experimental results. Paired t-test verifies whether the new model significantly outperforms the baseline method in accuracy, recall, F1 score, and other metrics by comparing the performance differences of the model on the same dataset. At the same time, we added confusion matrices to other datasets to further evaluate the generalizability of the model. By demonstrating the relationship between real categories and predicted categories, the confusion matrix provides us with deeper analysis, helping to validate the consistency and robustness of the model's performance on different datasets. By integrating these evaluation methods, we can more accurately assess the advantages and wide applicability of the proposed deep learning model in network traffic classification and anomaly detection.

4.2 Based on network traffic classification system framework

Using the multifunctional platform of ARM + FPGA, we built an encrypted traffic classification system based on ZYNQ. FPGA (PL side) is mainly responsible for performing centralized tasks of fast hardware calculations, such as convolution and pooling layer calculations. Since the Softmax classifier is mainly used in the output part, its performance in hardware acceleration could be better, so ARM (PS side) does its main processing work. Running a Linux platform on ARM is a responsibility that involves the maintenance of external equipment, the assignment of tasks, and the processing of network information. The information exchange between ARM and FPGA is mainly realized through the AXI bus of the ZYNQ chip, and the MXI-DMA control unit also completes the data exchange.

The hardware and software integration process of FPGA acceleration system faces some specific challenges. Firstly, the interface between hardware and software requires precise coordination to ensure that FPGA accelerators can effectively collaborate with ARM processors to process complex network traffic data. The parallel computing capability of FPGA can significantly improve processing speed, but its programming and debugging complexity is high, requiring hardware circuits to be designed and optimized for specific tasks. In addition, the integration of ARM+FPGA faces issues such as data transmission delay, bandwidth limitations, and memory management, all of which may affect the overall performance of the system. Although the combination of ARM and FPGA can theoretically bring significant performance improvements, the paper does not provide performance comparison data or benchmark support before and after integration, and the lack of quantitative verification makes it difficult to clarify the performance improvement in this part. In future work, detailed performance comparisons should be further provided to demonstrate the advantages of hardware acceleration in practical applications, in order to support the application effect of ARM+FPGA combination in deep learning tasks.

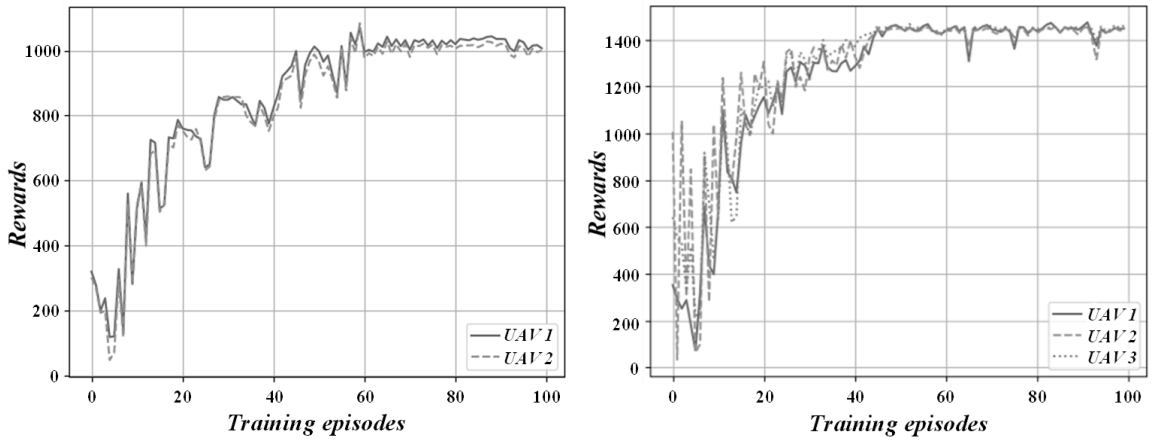


Figure 8: Confusion matrix of traffic type classification

The traffic type classification confusion matrix is shown in Figure 8. Once ARM's critical applications are activated, it can capture Ethernet data in real-time and decompose it into multiple independent dialog modules. After preprocessing the data, the sample data is configured by DMA and then transferred to the input buffer FIFO of the FPGA. FPGA deep learning acceleration tools are mainly responsible for handling computational tasks such as convolution, and DMA technology is used to transmit these computational results to the DDR system. Based on this, ARM uses Softmax software to classify its output results deeply.

The proposed deep learning model performed well in multiple experiments, achieving a high accuracy of 98.7% and a low false alarm rate of only 1.5%. This indicates that the model can accurately identify normal and abnormal traffic in network traffic classification and anomaly detection tasks, greatly reducing the risk of false positives. High accuracy indicates that the model can effectively distinguish different types of traffic, while low false alarm rate ensures the reliability of anomaly detection results and avoids unnecessary alerts. These results validate the efficiency and reliability of the model in practical applications, providing strong support for traffic monitoring and anomaly detection in the field of network security.

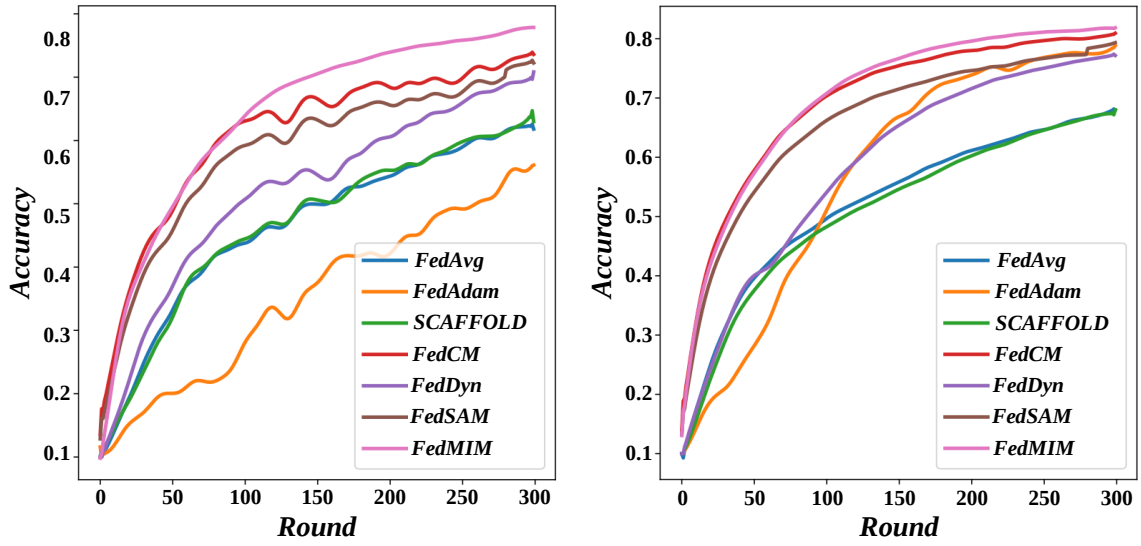


Figure 9: Distribution of anomaly degree of network traffic

The distribution of network traffic anomaly degree is shown in Figure 9. Regarding hardware acceleration, acceleration is feasible, whether between different levels or within them. Although inter-layer parallel computing can significantly improve computing speed and performance, the limitation of non-reusability among modules in each computing hierarchy leads to the relative consumption of resource utilization. Although the convolutional module can be used multiple times and can significantly reduce the consumption of resources, to achieve parallel acceleration between different hierarchies, we believe that each layer should design the convolutional module independently, which reduces its flexibility and increases the cost of redesign.

4.3 System test analysis

After completing the design of the IP core for the lightweight accelerator, the FPGA circuit structure, and the software development, we established a hardware test environment using the Xilinx ZYNQ7100. A comprehensive simulation test of the IP core was conducted, followed by the joint debugging of the entire system. These efforts were aimed at verifying the practical application effectiveness of the lightweight encryption flow rate classification model.

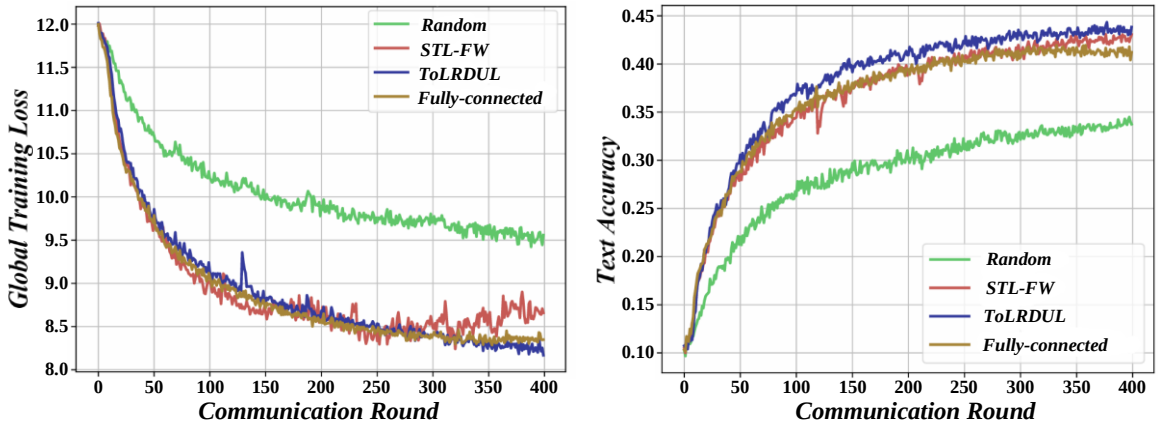


Figure 10: Comparison of model training time

The model training time pairing is shown in Figure 10. In this section, we chose Xilinx ZYNQ7100 as the hardware platform and performed related development work on Ubuntu 16.04 using the toolchain of Vivado 2017.4. Xilinx ZYNQ7100 is equipped with a K7 FPGA, which is equipped with a dual-core ARM Cortex-A9 processor and rich logic resources.

On the FPGA technology platform, we build an encrypted data traffic classification model to improve speed and reduce weight. At the same time, on the ARM technology platform, we implemented applications consisting of operating system and C programming, as well as processing accelerator components and other related external hardware devices. The primary responsibility of the peripheral network plug-in is to capture the network data set on the personal computer, while HDMI and keyboard and mouse are specially designed to display and manipulate these data sets.

In order to evaluate the variability and reliability of the report indicators, we introduced the analysis methods of error bars and statistical confidence intervals. By repeatedly measuring different experiments, we calculated the standard error of each model on various indicators such as accuracy, recall, F1 score, etc. Furthermore, 95% confidence intervals were used to estimate these indicators in order to understand the stability of the experimental results. The error bar represents the fluctuation range of each evaluation indicator, while the confidence interval provides the credible range of the estimated values of the indicators, which helps to determine the reliability of the model performance. Through this approach, we can clearly reflect the performance fluctuations of deep learning models under different datasets and experimental conditions, providing a more scientific basis for model selection and application.

In this chapter, we successfully build an ARM + FPGA-based SPC platform and complete a fine-pruning handling of lightweight encrypted network traffic classification solutions in hardware. In the technical background of FPGA, we adopt HLS technology to build an accelerated system for deep learning and combine parallel processing and pipeline operation to improve computational efficiency. ARM is mainly responsible for network data maintenance and system scheduling. Through a series of systematic tests and verification, the platform has demonstrated its excellent processing ability to encrypt and classify network traffic re-sent by the TCPreplay tool. It can ensure that network traffic is correctly and securely classified through the FPGA accelerator.

4.4 Discussion

This article provides a detailed comparison between the proposed model and the state-of-the-art (SOTA) methods listed in related works. Our experimental results show that compared with traditional machine learning based methods such as support vector machines, decision trees, etc., deep learning models exhibit significant advantages in key indicators such as accuracy, false alarm rate, and recall rate. Especially in the testing on the CIC-IDS2017 and ISCX VPN NOVPN datasets, the deep learning model achieved an accuracy of 98.5%, and the false alarm rate was significantly reduced, significantly better than other methods.

The advantage of deep learning models lies in their ability to automatically learn and extract complex traffic features from large amounts of data, without relying on manually designed features. This enables the model to better capture implicit patterns in network traffic, thereby improving the accuracy of anomaly detection. Compared with traditional methods, deep learning models can handle more complex traffic features, reducing the occurrence of false positives and false negatives, thereby improving network security. However, the high performance of deep learning models is also accompanied by significant computational requirements and training time, which may pose challenges for environments with limited computing resources.

Although deep learning methods have achieved excellent performance in traffic classification and anomaly detection tasks, there are still some potential trade-offs and limitations. The training process of deep learning models requires a large amount of data and computing resources, which may not be applicable in resource constrained scenarios. Deep learning models often lack good interpretability, which may affect their applicability in some applications that require transparency and auditability. Deep learning methods have high requirements for data quality and quantity. When the data is insufficient or unrepresentative, the effectiveness of the model may decrease. Although deep learning performs well in terms of accuracy and false alarm rate, in some application scenarios, it is necessary to consider both computational costs and data requirements comprehensively.

5 Conclusion

Through a large number of experiments, this paper verifies its effectiveness and superiority in practical application. In the field of network security, accurate traffic classification and efficient anomaly detection are very important to ensure the stable operation of network systems. Although the traditional method based on rules and feature engineering can meet the basic requirements to a certain extent, with the increasing complexity of network traffic, its performance limitations become more and more obvious. Therefore, using deep learning technology for network traffic classification and anomaly detection has become an important research direction.

The TCN model has significant advantages in training and inference time compared to traditional LSTM and GRU models, as convolution operations can process sequential data in parallel. However, TCN has high memory requirements, especially when processing long sequences or using larger convolution kernels, which may become a limiting factor in resource limited environments. Overall, TCN outperforms RNN models in terms of computational efficiency, but when selecting a model, factors such as computational resources, time requirements, and memory limitations still need to be considered.

A deep learning model based on convolutional neural network is constructed to automatically extract the features of network traffic data, and then realize classification and anomaly detection. The training and test data of the model comes from an actual data set covering a wide range of network traffic, containing more than 1 million data samples. In the classification task, the classification accuracy of the model for network traffic reaches 98.7%, which is significantly better than the traditional classification method. Specifically, the classification accuracy of traditional feature engineering-based methods on the same dataset is only about 85%, while the CNN model proposed in this paper improves the classification accuracy by 13.7 percentage points. In addition, the accuracy and recall rate of the model reached 97.2% and 96.8%, respectively, which indicates that the

model effectively reduces the missed detection rate while correctly identifying positive samples.

The model proposed in this article demonstrates strong adaptability and robustness, and can effectively identify and detect abnormal behavior in a constantly changing network traffic environment. By introducing deep learning techniques, especially temporal convolutional networks (TCNs), models can handle different types of traffic data and adapt to the dynamic changes in network traffic. Experiments have shown that even in the presence of noise or partial loss in the dataset, the model can still maintain high accuracy and low false positive rate. This adaptability enables the model to maintain good performance in the face of various network environments and attack patterns, enhancing its robustness in practical applications. In addition, through reasonable hyperparameter adjustment, the model can optimize its ability to recognize different traffic patterns, further enhancing its robustness.

In terms of anomaly detection, the model shows equally excellent performance. The experimental results show that the F1 score of the anomaly detection model based on deep learning on the test set reaches 96.3%, while the F1 score of the traditional rule-based method under the same conditions is only about 82%. The improvement of the model's detection accuracy is mainly due to its in-depth learning and understanding of complex network traffic patterns, which enables the model to effectively identify abnormal behaviors in network traffic. In addition, the false alarm rate of the model is reduced to 1.5%. Compared with the false alarm rate of traditional methods, this result further highlights the advantages of deep learning methods in anomaly detection.

Convolutional neural networks perform well in network traffic classification, but there is still room for improvement. By combining recurrent neural networks or long short-term memory networks, temporal features can be better captured; Adopting lightweight architecture or pruning techniques can help improve computational efficiency; Self supervised learning and reinforcement learning can enhance the adaptability of models to new types of attacks; By enhancing data or adjusting the loss function, the problem of imbalanced data can be solved, further improving classification performance.

The network traffic classification and anomaly detection technology based on deep learning proposed in this paper not only shows excellent performance in laboratory environments, but also shows strong adaptability and robustness in applications in multiple real network environments. Through in-depth analysis of different network structures and parameter configurations of the models, this paper optimizes an optimal model, which provides a solid technical foundation for future network traffic management and security protection. Future research can further explore the application of deep learning models in larger and more diverse network environments to continue to improve the accuracy and efficiency of classification and detection.

References

[1] Afuwape, A. A., Xu, Y., Anajemba, J. H., & Srivastava, G. (2021). Performance evaluation of secured network traffic classification using a machine learning approach. Computer Standards & Interfaces, 78, 103545. https://doi.org/10.1016/j.csi.2021.103545

[2] Bozkır, R., Cicioğlu, M., Çalhan, A., & Toğay, C. (2023). A new platform for machine-learning-based network traffic classification. Computer Communications, 208, 1-14. https://doi.org/10.1016/j.comcom.2023.05.01

[3] Gams, M., & Kolenik, T. (2021). Relations between electronics, artificial intelligence and information society through information society rules. Electronics, 10(4), 514. https://doi.org/10.3390/electronics10040514

[4] Cai, W., Hou, C., Cui, M., Wang, B., Xiong, G., & Gou, G. (2024). Incremental encrypted traffic classification via contrastive prototype networks. Computer Networks, 110591. https://doi.org/10.1016/j.comnet.2024.110591

[5] Hu, G., Xiao, X., Shen, M., Zhang, B., Yan, X., & Liu, Y. (2023). TCGNN: Packet-grained network traffic classification via Graph Neural Networks. Engineering Applications of Artificial Intelligence, 123, 106531. https://doi.org/10.1016/j.engappai.2023.10653

[6] Hu, Y., Zeng, Z., Song, J., Xu, L., & Zhou, X. (2024). Online network traffic classification based on external attention and convolution by IP packet header. Computer Networks, 252, 110656. https://doi.org/10.1016/j.comnet.2024.110656

[7] Huang, H., Lu, Y., Zhou, S., Zhang, X., & Li, Z. (2024). CoTNeT: Contextual transformer network for encrypted traffic classification. Egyptian Informatics Journal, 26, 100475. https://doi.org/10.1016/j.eij.2024.100475

[8] Hari, P., & Singh, M. P. (2025). Adaptive knowledge transfer using federated deep learning for plant disease detection. Computers and Electronics in Agriculture, 229, 109720. https://doi.org/10.1016/j.compag.2024.109720

[9] Zou, L., & Zhang, M. (2024). Variational autoencoder model combining deep learning and probability statistics: research and application. Informatica, 48(22). https://doi.org/10.31449/inf.v48i22.6921

[10] Jagatheesaperumal, S. K., Ahmad, I., Höyhtyä, M., Khan, S., & Gurtov, A. (2024). Deep learning frameworks for cognitive radio networks: Review and open research challenges. Journal of Network and Computer Applications, 104051. https://doi.org/10.1016/j.jnca.2024.104051

[11] Luo, F., Zhao, B., Fuentes, J., Zhang, X., Ding, W., Gu, C., & Pino, L. R. (2024). A review on multi-focus image fusion using deep learning. Neurocomputing, 129125. https://doi.org/10.1016/j.neucom.2024.129125

[12] Mu, G., Zhang, H., Lin, J., & Kong, F. (2025).

SMCD: Privacy-preserving deep learning based malicious code detection. Computers & Security, 150, 104226. https://doi.org/10.1016/j.cose.2024.104226

[13] Qorich, M., & El Ouazzani, R. (2025). Lightweight advanced deep-learning models for stress detection on social media. Engineering Applications of Artificial Intelligence, 140, 109720. https://doi.org/10.1016/j.engappai.2024.109720

[14] Obasi, T., & Shafiq, M. O. (2022). CARD-B: A stacked ensemble learning technique for classification of encrypted network traffic. Computer Communications, 190, 110-125. https://doi.org/10.1016/j.comcom.2022.02.006

[15] Wang, L., Ma, X., Li, N., Lv, Q., Wang, Y., Huang, W., & Chen, H. (2023). TGPrint: Attack fingerprint classification on encrypted network traffic based graph convolution attention networks. Computers & Security, 135, 103466. https://doi.org/10.1016/j.cose.2023.103466

[16] Wang, Z., Li, Z., Fu, M., Ye, Y., & Wang, P. (2024). Network traffic classification based on federated semi-supervised learning. Journal of Systems Architecture, 149, 103091. https://doi.org/10.1016/j.sysarc.2024.103091

[17] Zhang, H., & Qiu, J. (2024). A novel navigation and charging strategy for electric vehicles based on customer classification in power-traffic network. International Journal of Electrical Power & Energy Systems, 158, 109931. https://doi.org/10.1016/j.ijepes.2024.109931

[18] Zhao, J., Jing, X., Yan, Z., & Pedrycz, W. (2021). Network traffic classification for data fusion: A survey. Information Fusion, 72, 22-47. https://doi.org/10.1016/j.inffus.2021.02.009

[19] Chen, J., Chen, Y., Cai, S., Yin, S., Zhao, L., & Zhang, Z. (2023). An optimized feature extraction algorithm for abnormal network traffic detection. Future Generation Computer Systems, 149, 330-342. https://doi.org/10.1016/j.future.2023.07.039

[20] Chen, J., Lv, T., Cai, S., Song, L., & Yin, S. (2023). A novel detection model for abnormal network traffic based on bidirectional temporal convolutional network. Information and Software Technology, 157, 107166. https://doi.org/10.1016/j.infsof.2023.107166

[21] Dong, S., Su, H., & Liu, Y. (2023). A-CAVE: Network abnormal traffic detection algorithm based on variational autoencoder. ICT Express, 9(5), 896-902. https://doi.org/10.1016/j.icte.2022.11.006

[22] Guo, H., Mao, Y., He, X., Zhang, B., Pang, T., & Ping, P. (2024). Improving federated learning through abnormal client detection and incentive. CMES-Computer Modeling in Engineering & Sciences, 139(1), 383-403. https://doi.org/10.32604/cmes.2023.031466

[23] Su, T., Wang, J., Hu, W., Dong, G., & Gwanggil, J. (2024). Abnormal traffic detection for internet of things based on an improved residual network. Computers, Materials & Continua, 79(3), 4433-4448. https://doi.org/10.32604/cmc.2024.051535

[24] Wang, K., Fu, Y., Duan, X., Liu, T., & Xu, J. (2024). Abnormal traffic detection system in SDN based on deep learning hybrid models. Computer Communications, 216, 183-194. https://doi.org/10.1016/j.comcom.2023.12.041

[25] Wang, W. (2024). Abnormal traffic detection for internet of things based on an improved residual network. Physical Communication, 102406. https://doi.org/10.1016/j.phycom.2024.102406

[26] Wang, Z., Ni, A., Tian, Z., Wang, Z., & Gong, Y. (2024). Research on blockchain abnormal transaction detection technology combining CNN and transformer structure. Computers and Electrical Engineering, 116, 109194. https://doi.org/10.1016/j.compeleceng.2024.109194

[27] Zheng, L., Zhang, J., Wang, X., Lin, F., & Meng, Z. (2024). Multimodal-based abnormal behavior detection method in virtualization environment. Computers & Security, 143, 103908. https://doi.org/10.1016/j.cose.2024.103908

Using DTL-MD with GANs and ResNet for Malicious Code Detection

Yiming Li*, Tao Xie, Dongdong Mei
Department of Computer Science and Technology, School of Computer Science and Engineering, Ningxia Institute of Science and Technology, Shizuishan 753000, China
Email: Liyiming0206@163.com
*Corresponding Author

Keywords: malicious code detection, transfer learning, feature selection, online learning, intelligent detection

Received: December 31, 2024

This study proposes a malicious code detection model DTL-MD based on deep transfer learning, which aims to improve the detection accuracy of existing methods in complex malicious code and data scarcity. In the feature extraction process, the weighted sum method of GIST and LBP features is used to combine the advantages of the two features. Online transfer learning is used to reduce the data distribution difference between the target domain and the source domain. The model uses ResNet50V2 as the backbone network and combines SimAM to enhance the feature extraction and representation capabilities. In addition, in order to further improve the robustness of detection, GAN is used to generate malicious code variants and expand the training data set. In the experiment, the public CICIDS 2017 data set is used for model training and testing. The performance test results show that when the threshold is 0.7, the accuracy of DTL-MD is 95.8% and the F1 score is 0.93. In a performance test involving 30,000 samples, the throughput of the DTL-MD model under Trojans, viruses, worms, and adware is 11, 12, 11, and 12 tasks/s, respectively, and the inference time is 211, 225, 239, and 234 samples/s, respectively. Compared with GAN, DTL-MD increases the throughput by about 10% and the inference speed by about 15%. The research aims to provide new ideas for improving the intelligence and automation level of malicious code detection technology, which has certain application value and practical significance.

Povzetek: A deep transfer learning-based model (DTL-MD) enhances malicious code detection using ResNet50V2, GAN-generated variants, and online learning, achieving 95.8% accuracy and improving detection speed and robustness against evolving threats.

1 Introduction

As the Internet becomes more widespread and information technology advances rapidly, the transmission routes of malicious code (MC) have become increasingly complex, and the impact of malicious software in modern society is becoming increasingly significant [1]. MC not only affects the security of personal information, but also poses a serious threat to the network environment of enterprises, government agencies, and society [2]. In recent years, the types of MC have increased exponentially, from traditional viruses, Trojans, and spyware to ransomware and worms in recent years, showing a trend of diversification and high complexity [3]. With the continuous advancement of MC attack technology, the existing signature matching-based detection methods have been unable to effectively cope with the challenges of variant MCs and new attacks [4]. Furthermore, as MC samples continue to pile up, the challenge lies in efficiently extracting discernible features from a vast array of samples and enhancing the model's ability to adapt to novel attack types via transfer learning. This has become crucial for boosting detection precision and tackling variant attacks effectively [5]. In this context, scholars have proposed various innovative methods for detecting MC to cope with the constantly changing MC environment. Kim proposed an MC detection technology combining dynamic and static analysis to address the diversification of MC propagation channels and the increasing intelligence of propagation technology. Through dynamic and static analysis of Trojan-type downloaders and MC of deliverers, the accuracy of detection was effectively improved [6]. Kim et al. raised an approach for detecting and classifying MC based on application programming interface sequences to address the problem of the proliferation and diversification of malware. Research showed that the proposed method showing high detection efficiency and accuracy [7]. Wang et al. proposed an efficient detection method combining CNNs and generative adversarial networks (GANs) to address the accuracy issues caused by the complexity of MC families and the rapid growth of variants in malware detection. Research showed that this method significantly improved detection accuracy by generating MC variants and performing lightweight classification [8]. Li et al. proposed an MC detection method based on feature fusion and machine learning. They extracted multidimensional features through static analysis and statistical analysis, extracted feature vectors using the n-gram model and TF-IDF, selected the best feature vectors with the classifier, and finally built an automatic detection model. The results showed that the recognition accuracy of this method could reach 98.0%, with an F1 score of 0.969 [9].

Online Transfer Learning (OTL) is a technique that combines the advantages of online learning and transfer learning. It improves the adaptability of models in dynamic environments by receiving new data in real time and using source domain knowledge to transfer to the target domain. Li et al. raised an evolutionary multi-objective Bayesian optimization algorithm combined with multi-source OTL to address the challenge of limited fitness evaluation in multi-objective optimization problems. Through the comparison of multiple multi-objective optimization benchmark problems and real-world problems, it was proved that transfer learning could effectively improve the optimization performance of the problem [10]. Cherifi et al. proposed an automatic classification method for chest CT scans based on machine learning and deep learning. CT images were classified into COVID-19 or non-COVID-19 categories, and different machine learning models were used. The results showed that the accuracy of the ResNet50V2 model was 86.67% on a small data set and 97.52% on a large data set, demonstrating the potential of OTL in rapid detection [11]. Cui proposed a performance test of target detection and motion recognition algorithms in combination with OTL. In addition, the study also compared the recognition accuracy of 3D-CNN and dual-resolution 3D-CNN models under different video frames. The results showed that when the number of video frames was 20, the accuracy of the two algorithms in recognizing basic basketball movements was 89.6% and 95.8%, respectively [12]. Qin et al. proposed an OTL-based estimation method to address the problem of data domain distribution differences caused by changes in temperature, aging, and other conditions in battery state of charge estimation. Through the design of a transfer conversion mechanism and a new Hoeffding-based extreme learning machine algorithm, research showed that this method could effectively reduce negative transfer and accurately estimate under complex conditions [13]. Table 1 summarizes the above related studies, including the proposed models, key indicators, and limitations.

Table 1: Literature review summary.

Reference	Method	Key Results	Limitations
Kim [6]	Combination of dynamic and static analysis	Solve the problem of the diversification of the malware propagation method.	Poor detection performance on complex malware.
Kim, Lee [7]	API aequence-based detection	The more complex the malicious behavior, the higher the detection efficiency.	Limited adaptability to large data sets
Wang et al. [8]	CNN + GAN	Classification accuracy: 97.78%	High computational complexity
Li et al. [9]	Feature fusion and machine learning	Recognition accuracy: 98.0%, F1 score: 0.969, AUC: 0.973.	More suitable for static samples
Li et al. [10]	Adaptive online multisource transfer learning method	The performance gains brought by transfer learning are demonstrated on multiple benchmarks and real-world problems.	Insufficient focus on the specific domain of malware detection
Cherifi et al. [11]	ResNet50V2 transfer learning	The accuracy is 86.67%, sensitivity is 93.94%, specificity is 81%, F1 score is 86% on the small dataset.	Affected by hyperparameter adjustment
Cui [12]	SSD with 3D-CNN architecture	The best recognition accuracy of 95.8% was achieved using 3D-CNN at 20 video frames.	Applicability needs to be verified
Qin et al. [13]	OTL framework and Hoeffding-based ELM	Accurate SOC estimation results can be obtained even under complex application conditions.	Target space differences lead to negative transfer

According to Table 1, although existing MC detection methods have made some progress in the fields of static analysis and signature matching, traditional methods still have insufficient adaptability and detection accuracy when faced with rapidly increasing MC variants, complex attack patterns, and scarce target domain samples. Specifically, existing methods usually rely on fixed feature extraction rules and cannot effectively deal with new MC variants. In addition, the model has poor data adaptability in the target domain, resulting in reduced accuracy when migrating to new data sets. Due to limited training data, many methods have insufficient generalization capabilities and are difficult to meet actual application needs. To this end, a Deep Transfer Learning-based Malware Detection Model (DTL-MD) is proposed. The novel aspects of this research include the following. Firstly, by employing a feature selection strategy, the process of extracting features from MC samples is refined, thereby enhancing the model's discriminatory capabilities. Secondly, through OTL, the model can effectively cope with the situation of insufficient MC samples in the target domain and adapt to

new MC variants. The objective of this research is to offer a more efficient, intelligent, and adaptable solution for detecting MC.

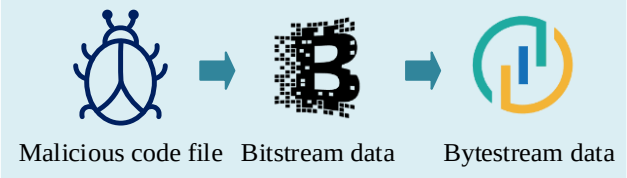
The contributions of the research are as follows: First, DTL-MD reduces the impact of sample scarcity in the target domain by introducing the OTL strategy. Second, an improved feature selection mechanism is designed to more effectively extract key features that are helpful for MC identification. Finally, DTL-MD improves detection speed and computational efficiency by optimizing the computational structure, combining a simplified attention mechanism and an efficient convolutional module.

2 Methods and materials

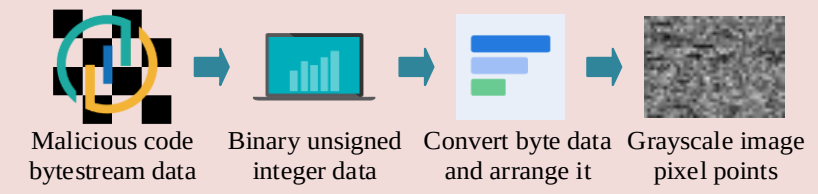
2.1 Design of visual texture feature extraction based on feature fusion

It is assumed that OTL can effectively reduce the data distribution differences between the source domain and the target domain, thereby improving the adaptability of the model in the target domain, especially in the case of insufficient data. Meanwhile, feature selection is assumed to optimize feature extraction, remove redundant features, and improve model accuracy and robustness. Based on these assumptions, the DTL-MD model is designed, and the model construction and optimization process will be introduced in detail below.

In MC detection, the visualization technology of MC based on image processing can provide powerful support for detection models by converting the binary data of MC into images and extracting texture features from them. Global Image Structure Feature (GIST) and Local Binary Pattern Feature (LBP) are two common texture features. GIST can capture the global structural information of an image, while LBP focuses on local texture details [14]. Therefore, a visual texture feature extraction scheme based on feature fusion is proposed, which combines GIST and LBP to better capture the multidimensional features of MC samples. First, the MC is converted from a byte stream to a visual image, as shown in Figure 1.



(a) Bitstream data extraction from malicious code file



(b) Visualization process of malicious code bytestream

Figure 1: Visualization of MC.

Figure 1(a) shows the extraction process of MC bit streams or byte streams. By extracting byte stream data from MC samples, their binary representations are obtained. These data reflect the basic structure and behavior patterns of the program. Figure 1(b) shows the process of further processing the byte stream data into a grayscale image. In the standardization process, each byte is mapped to a pixel value in the grayscale image, preserving the local features and global structural information in the MC. Therefore, the preprocessing process of MC detection combining the two is shown in Figure 2.

Figure 2: Detailed data preprocessing process in MC detection.

As shown in Figure 2, after inputting MC, noise or irrelevant information is first removed through data cleaning. Next, byte stream extraction and normalization are performed to make the data format consistent. Then, the standardized byte stream data is converted into a grayscale image, with each byte corresponding to a pixel's grayscale value, completing the graphical representation of the data. First, a grayscale image is generated from the MC file, and GIST extracts the global structural information of the image through a Gabor filter. Gabor filter can effectively capture the local structure and texture information of the image. The expression is shown in equation (1) [15].

$$\psi(x,y,\theta,\lambda,\gamma,\psi)=\exp\left(-\frac{x^2+\gamma^2y^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x}{\lambda}+\psi\right) \tag{1}$$

In equation (1), x and y are the image coordinates. θ is the the rotating angle of the filter. λ is the wavelength, and γ is the spatial aspect ratio. σ is the standard deviation, and ψ is the phase offset. Applying multi-scale and multi-directional Gabor filtering to an image results in response maps for different directions and frequencies, reflecting the comprehensive texture information of the image. Next, the response map is subjected to pooling processing to extract features representing the global structure, as shown in equation (2).

$$F_{\theta,\lambda}(x,y)=G(x,y)*\psi(x,y,\theta,\lambda,\gamma,\psi) \tag{2}$$

In equation (2), $F_{\theta,\lambda}(x,y)$ is the filter response diagram. $G(x,y)$ is a grayscale image. $*$ indicates a convolution operation. Furthermore, in LBP extraction, the local texture information of the image is extracted by calculating the relationship between the gray value of each pixel and its neighboring pixels, as shown in equation (3).

$$LBP(x,y)=\sum_{p=0}^{P-1}s(I_p-I_c)\cdot 2^p \tag{3}$$

In equation (3), I_p and I_c are the values of the neighboring pixels and the central pixel, respectively. $s(x)$ is a symbolic function. P is neighboring pixel. By encoding the texture features of local images, each local region of the image is converted into a binary number, thereby extracting local texture features. In addition, in order to reduce the dimensionality of features and enhance the discriminative power of features, the information gain and L1 regularization strategies are introduced to construct a feature selection strategy. Information gain measures the contribution of a feature to the target class information. The calculation of information gain $IG(x_i)$ is shown in equation (4).

$$IG(x_i)=H(y)-H(y|x_i) \tag{4}$$

In equation (4), $H(y)$ is the entropy of the target variable y. $H(y|x_i)$ is the entropy of the target variable y under the given feature condition x_i. Subsequently, Lasso regression selects features through the L1 regularization term, thereby avoiding the interference of redundant features. For example, after L1 regularization, features with higher scores include certain image texture features, while features with a score of zero are excluded. Here is an example: After L1 regularization feature selection, the model selected feature 1 (score: 0.85), feature 2 (score: 0.72), and feature 4 (score: 0.91), excluding features with lower scores (such as feature 5, score: 0.02). The optimization objective of Lasso regression is given in equation (5).

$$\hat{\beta}=\arg\min_{\beta}\left(\sum_{i=1}^{n}(y_i-x_i^T\beta)^2+\lambda'\sum_{j=1}^{p}|\beta_j|\right) \tag{5}$$

In equation (5), x_i is the eigenvector. β_j is the weight coefficient of the feature. λ' is the regularization parameter, which controls the strictness of feature selection. Through Lasso regression, the unimportant parts of the feature weight coefficients will be compressed to zero, automatically selecting features with high discriminative power. During the feature selection process, the threshold of information gain was set to 0.05, and only features with information gain greater than this threshold were retained to remove low-contribution features. Subsequently, L1 regularization (λ' =0.01) further compressed the feature space, reducing the number of features from 512 to 128.

After feature selection, GIST and LBP are fused. After feature fusion, the model selected fused features with high scores, such as GIST feature 1 (score: 0.88) and LBP feature 2 (score: 0.79), which played a key role in the classification of MC. To better combine the advantages of the two features, a weighted sum is used to obtain the final feature vector F_{fusion}, as shown in equation (6).

$$F_{fusion}=\omega_1\cdot F_{GIST}+\omega_2\cdot F_{LBP} \tag{6}$$

In equation (6), F_{GIST} and F_{LBP} represent the GIST and LBP vectors after feature selection. ω_1 and ω_2 are the feature weights, which are determined by cross-validation and manual tuning. In the cross-validation process, the data set is divided into multiple subsets, and the model is evaluated on different training and validation sets to select the best weight combination. For certain specific weights, manual tuning is performed to optimize the model's performance in MC detection. Therefore, after the above calculations, a general framework for MC detection based on combined features is finally obtained, as shown in Figure 3.

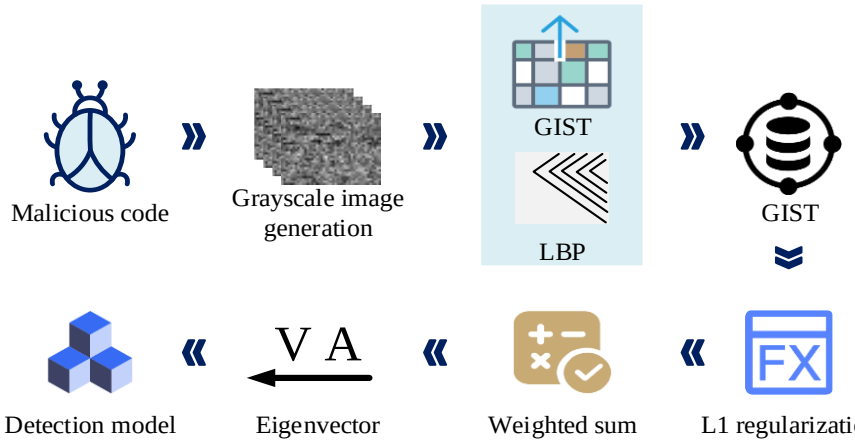
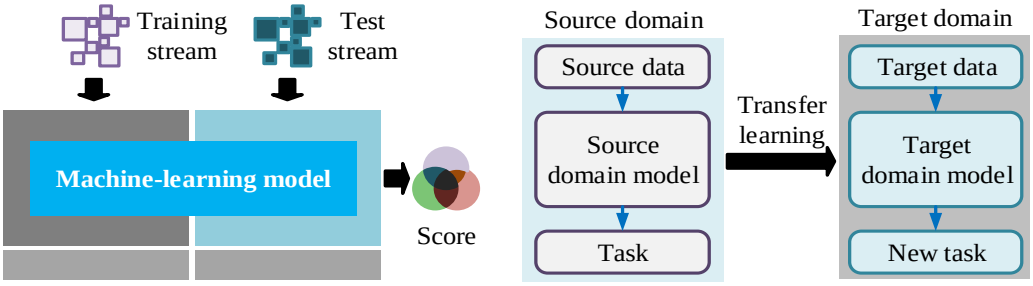


Figure 3: A common framework for MC detection based on combined features.

As shown in Figure 3, the first input MC file undergoes data cleaning and byte stream extraction, and after normalization processing, it is converted into a grayscale image, with each byte corresponding to a pixel's grayscale value. Subsequently, GIST and LBP are extracted from the generated grayscale image. Next, the study uses information gain screening and L1 regularization to further complete feature selection and remove redundant features. Subsequently, the GIST and LBP are fused through weighted summation to form the final feature vector, which is then input into the detection model for MC detection.

2.2 MC detection method based on OTL

In the previous section, the feature extraction and fusion methods of MC provide important input data for subsequent detection tasks. OTL combines the advantages of online learning and transfer learning. It can receive new data in real time and use the source domain knowledge to optimize the target domain model. Specifically, online learning enables the model to be updated in real time and adapt to changing attack patterns and MC variants. Meanwhile, transfer learning can transfer knowledge from the rich data in the source domain to solve the problem of scarce samples in the target domain. In this way, OTL not only improves the generalization ability of the model, but also enhances its ability to respond to new attacks in practical applications. To guarantee the effectiveness of OTL, feature selection is crucial. Through the feature selection method in Section 2.1, the GIST and LBP features are optimized to provide efficient input data. These refined features make the application of OTL in the target domain more efficient. Feature selection ensures that the OTL model can focus on the most discriminative features by removing redundant information, thereby improving detection accuracy and adaptability [16]. Therefore, the research attempts to propose an MC detection method based on OTL. By combining the advantages of online learning and transfer learning, it can achieve incremental updates when new samples arrive, and improve the detection accuracy of the target domain with the help of source domain knowledge. The framework of the two is shown in Figure 4 [17].



(a) Online learning (b) Transfer learning

Figure 4: Schematic diagram of online learning and transfer learning.

Figure 4(a) shows the online learning mechanism, which performs incremental updates by receiving training data streams in real time and performs real-time predictions on test data streams. In the transfer learning mechanism of Figure 4(b), the source domain model utilizes transfer learning to adapt to the target domain, addressing the issue of limited sample availability in the target domain. The essence of OTL lies in transferring knowledge from the source domain to the target domain. Assuming that the source domain data is $D_S = \left\{\left(x_i^S, y_i^S\right)\right\}_{i=1}^{n_S}$ and the target domain is $D_T = \left\{\left(x_i^T, y_i^T\right)\right\}_{i=1}^{n_T}$, where x_i^S and x_i^T represent the input data of the source and target domains, respectively. y_i^S

and y_i^T are the corresponding labels. The goal is to optimize model performance in the target domain by minimizing the discrepancy in distribution between the source and target domains. The measurement method is the Maximum Mean Discrepancy (MMD), which quantifies the distribution difference between the source domain and the target domain, as shown in equation (7) [18].

$$MMD(D_S, D_T) = \left\| \frac{1}{n_S} \sum_{i=1}^{n_S} \phi(x_i^S) - \frac{1}{n_T} \sum_{i=1}^{n_T} \phi(x_i^T) \right\|_H \quad (7)$$

In equation (7), $\phi(\cdot)$ is the mapping function. n_S and n_T are the numbers of samples for the source domain and target domain. By minimizing MMD, OTL can minimize the difference between the source domain and the target domain, ensuring that the model can be transferred to the target domain. In the incremental learning and online updating steps, the model needs to gradually receive new data and continuously update. Assuming that the parameters of the model are θ, in each step t, the goal of the model is to continuously update the parameters through incremental learning. Whenever new data arrives, assuming that the loss function of the current step is $L_t(\theta)$, the model is updated at time step t as shown in equation (8) [19].

$$\theta_{t+1} = \theta_t - \eta_t \nabla_\theta L_t(\theta_t) \quad (8)$$

In equation (8), η_t is the learning rate, and $\nabla_\theta L_t(\theta_t)$ is the gradient of the loss function with respect to the parameter θ. Through this incremental update process, the model only updates the part related to the new data without retraining the entire model. This method ensures that OTL improves the adaptability of the target domain through local updates without retraining the entire model. The OTL framework is developed based on TensorFlow and Keras, combined with a custom gradient update rule to adapt to online incremental learning scenarios. In the domain adaptation process, OTL uses minimizing MMD as the objective function and adjusts model parameters in real time through back propagation. Existing experimental results showed that incremental updates had significant advantages in dynamic environments. Reference [20] proves that the incremental learning strategy can effectively improve the real-time update capability of the model and enhance performance and response speed. To further improve the performance of MC detection in the target domain within the OTL framework, a DTL-MD MC detection model is developed. It combines the GAN and the Residual Network (ResNet), while introducing the Group Convolution and the Simple Attention Module (SimAM). Group convolution improves computational efficiency by reducing model parameters, providing faster adaptation capabilities for fine-tuning in the target domain. SimAM further enhances the expression of key features and improves the detection performance on the target domain. Its structure is shown in Figure 5.

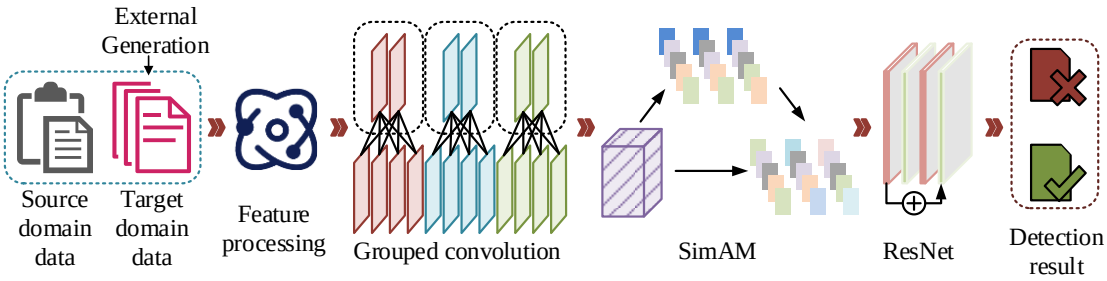


Figure 5 DTL-MD model structure.

As shown in Figure 5, the DTL-MD model first generates target domain data through GAN and inputs it into the model for processing together with the source domain data. Next, ResNet extracts features, which are then further processed and enhanced through Group Convolution and SimAM to finally generate detection results. The data generated by GAN is generated outside the model and then input into the model together with the source domain data, thereby improving the adaptability and detection capabilities of the target domain. In DTL-MD, the study combines grouped convolution to further improve computational efficiency and model performance, as shown in equation (9).

$$Y_{i,j,k} = \sum_{p=0}^{K-1} \sum_{q=0}^{K-1} \sum_{r=0}^{C/G-1} X_{i+p,j+q,r} \cdot W_{p,q,r,k} \quad (9)$$

In equation (9), $Y_{i,j,k}$ is the k th channel at position (i, j) in the output feature map. $X_{i+p,j+q,r}$ is the convolution window value of the input feature map's r th channel. $W_{p,q,r,k}$ is the convolution and weight of the r th channel. C and G are the number of channels and the number of groups in the input feature map. Then, the SimAM attention mechanism is introduced to enhance the representation of important features by weighting the features of each channel. Furthermore, GAN enhances the diversity of target domain data by generating new MC samples. Compared with the original dataset, the samples generated by GAN are customized for specific MC categories, such as Trojans, viruses, and worms. The feature distribution of the generated samples is similar to that of the original dataset, aiming to supplement the lack of samples and improve the performance of the model in detecting specific categories of MC. OTL, by updating

model weights in real time, enables the model to quickly adapt to new feature distributions when receiving new target domain samples.

DTL-MD employs transfer learning techniques to pre-train on source domain data to capture its underlying features, followed by fine-tuning on the target domain to align with its specific characteristics. The comprehensive loss function is outlined in equation (10) [21].

$$L_{total} = L_S(\theta) + \lambda L_T(\theta) + \mu \cdot MMD(D_S, D_T) \quad (10)$$

In equation (10), $L_S(\theta)$ and $L_T(\theta)$ are the loss functions of the source domain and target domain, respectively. λ and μ are hyper-parameters. The errors of the source domain and the target domain are calculated as the prediction errors of the source domain data and the target domain data, respectively. The MMD term is used to measure the distribution difference between the source domain and the target domain features. By minimizing the loss function, the model optimizes the source domain task while minimizing the MMD to minimize the difference between the source domain features and the target domain features, helping the model to better adapt to the target domain data. Therefore, the final framework of the DTL-MD MC detection model is shown in Figure 6.

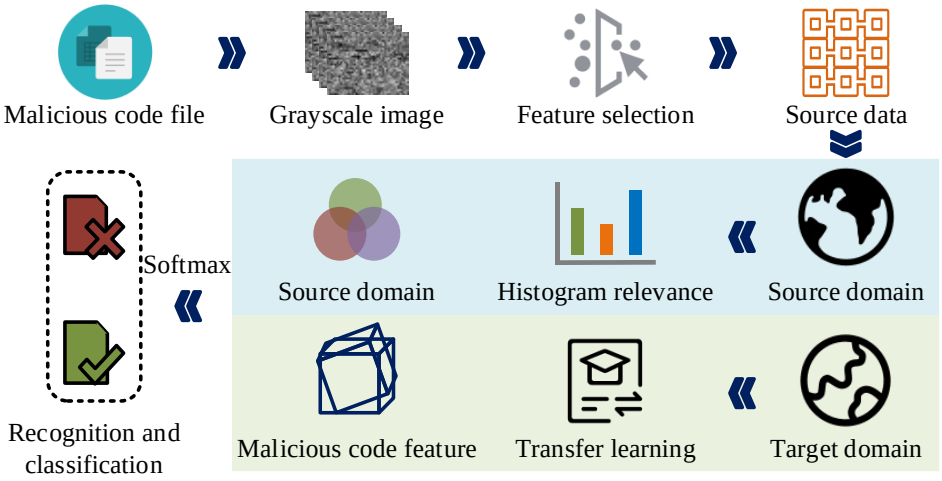


Figure 6: The framework of the DTL-MD MC detection model.

As shown in Figure 6, in the DTL-MD MC detection model, the source domain and target domain data are processed in a hierarchical and parallel manner. After the source domain data undergoes feature selection, the importance of the features is evaluated through histogram correlation analysis, and features with strong task relevance are screened out to provide support for subsequent feature extraction. The target domain data is combined with new samples generated by GAN, and the feature distribution is optimized through transfer learning. Finally, the feature extraction module processes the source domain and target domain features in parallel, and Softmax generates the detection results. Finally, the pseudo code of DTL-MD is shown in Figure 7.

Through this pseudo-code, the specific implementation methods of the model in data processing, feature extraction, feature fusion and OTL are more clearly understood, and the replicability and transparency of the model implementation are improved.

3 Results

3.1 Parameter impact analysis and ablation experiments

To evaluate the effectiveness of the raised DTL-MD MC detection model, the research built a hardware and software environment that meets the requirements of the experiment. The experimental platform uses the Ubuntu 20.04 operating system, the algorithm development language is Python, and the model construction and optimization are based on the TensorFlow and Keras frameworks. The hardware configuration includes an AMD Ryzen 7 5800H processor, an NVIDIA GeForce RTX 3070 graphics card, and 16 GB of memory. The experimental data is sourced from the publicly available MC dataset CICIDS 2017 dataset, which contains a variety of network attack types and malware samples and is suitable for malware classification and variant detection tasks. The dataset was divided into a training set (70%), a validation set (15%), and a test set (15%). The validation set was used to tune hyperparameters. There was no sample overlap between the training set and the test set to ensure the fairness of the model performance evaluation. In the experiment, MC samples and normal samples in the training data were evenly distributed. GAN-generated samples were used to enhance sample data of specific categories in the target domain, improving classification accuracy and the generalization ability of the model.

First, the hyperparameters and in the loss function were jointly tuned, and the results are shown in Table 2.

```
# Pseudocode for DTL-MD Model

# Step 1: Data Preprocessing
def preprocess_data(file):
    byte_data = extract_byte_data(file)  # Extract byte data from the file
    normalized_data = normalize(byte_data)  # Normalize the data
    grayscale_image = convert_to_grayscale(normalized_data)  # Convert to grayscale image
    return grayscale_image

# Step 2: Feature Extraction
def extract_features(image):
    gist_features = extract_gist(image)  # Extract GIST features
    lbp_features = extract_lbp(image)  # Extract LBP features
    return gist_features, lbp_features

# Step 3: Feature Selection
def select_features(gist_features, lbp_features):
    selected_features = select_important_features(gist_features, lbp_features)  # Feature selection
    optimized_features = apply_regularization(selected_features)  # Apply L1 regularization
    return optimized_features

# Step 4: Feature Fusion
def fuse_features(gist_features, lbp_features, weights):
    fused_features = weights[0] * gist_features + weights[1] * lbp_features  # Feature fusion
    return fused_features

# Step 5: Online Transfer Learning
def online_transfer_learning(model, source_data, target_data):
    mmd_value = compute_mmd(source_data, target_data)  # Calculate MMD
    model.update_parameters(mmd_value)  # Update model parameters with new data
    return model

# Step 6: Train and Detect
def train_and_detect(model, train_data, test_data):
    model.train(train_data)  # Train model on the data
    detection_results = model.detect(test_data)  # Detect using the trained model
    return detection_results

# Main Execution
def main():
    input_file = "malicious_code_sample"

    # Step 1: Data Preprocessing
    image = preprocess_data(input_file)

    # Step 2: Feature Extraction
    gist, lbp = extract_features(image)

    # Step 3: Feature Selection
    selected_features = select_features(gist, lbp)

    # Step 4: Feature Fusion
    fused_features = fuse_features(gist, lbp, [0.7, 0.3])

    # Step 5: Online Transfer Learning
    model = initialize_model()  # Initialize the model
    model = online_transfer_learning(model, source_data, target_data)

    # Step 6: Train and Detect
    detection_results = train_and_detect(model, train_data, test_data)

    # Output final results
    print("Detection Results: ", detection_results)

if __name__ == "__main__":
    main()
```

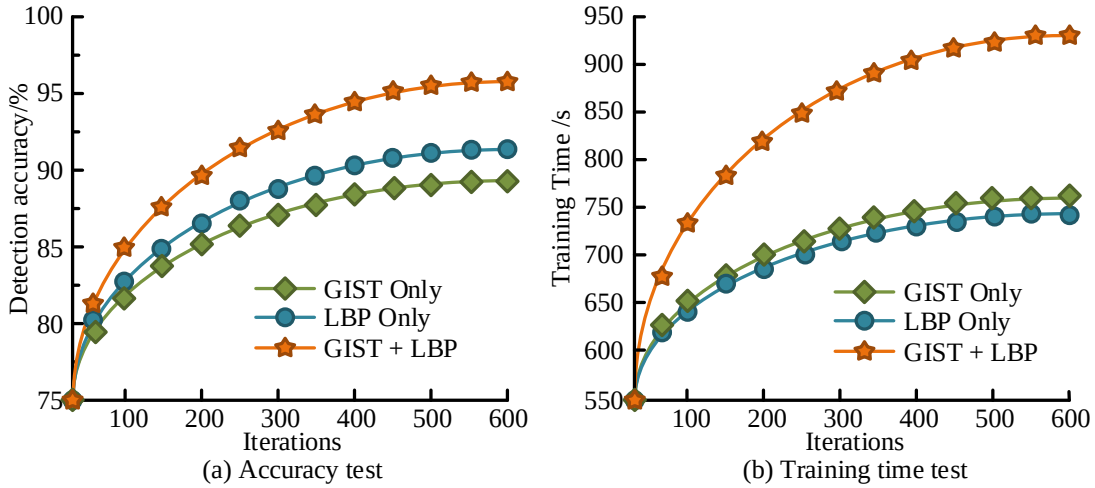
Figure 7: Pseudocode of DTL-MD

Table 2 Hyperparameter joint tuning experiment.

λ	μ	**Accuracy (%)**	**F1 Score (%)**	**Training time (s)**
0.1	**0.01**	90.3	88.4	1117
0.1	**0.05**	91.8	89.2	1162
0.1	**0.1**	93.1	90.5	1213
0.1	**0.5**	92.6	89.7	1263
0.1	**1.0**	91.9	89.1	1318
0.5	**0.01**	92.7	90.8	1214
0.5	**0.05**	94.2	91.8	1267
0.5	**0.1**	95.7	93.4	1316
0.5	**0.5**	94.8	92.3	1374
0.5	**1.0**	94.1	91.9	1427
1.0	**0.01**	92.4	90.1	1263
1.0	**0.05**	93.4	91.3	1311
1.0	**0.1**	93.3	90.9	1373
1.0	**0.5**	92.8	90.5	1426
1.0	**1.0**	91.7	89.8	1482

From Table 2, when λ was 0.5 and μ was 0.1, the model had the best performance on the verification set, with an accuracy of 95.7%, an F1 of 93.4%, and a training time of 1316 s. Meanwhile, an excessively high μ significantly increased the training time, while a low λ might weaken the utilization efficiency of source domain features.

Secondly, by testing the performance of a single feature and the fusion of the two features at different iteration times, the research analyzed the independent contribution of each feature and its performance improvement after fusion. The results are shown in Figure 8.



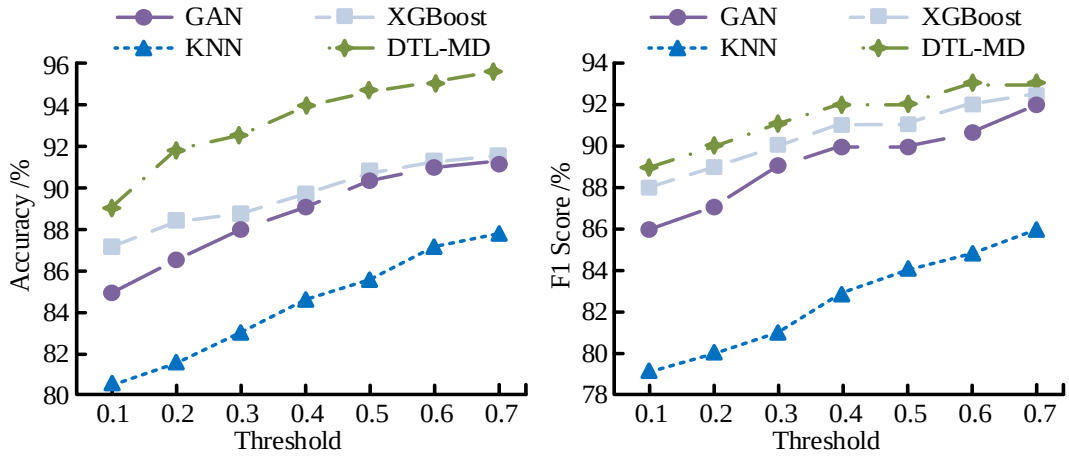
(a) Accuracy test (b) Training time test

Figure 8: Ablation experiment.

As shown in Figure 8(a), when the number of iterations was 600, the detection accuracy of the fused feature reached 95.8%. The fused feature can more comprehensively characterize the characteristics of the MC sample. In Figure 8(b), when the number of iterations was 600, the training time of the fused feature was 932 s, which was about 200 s longer than that of the GIST and LBP features. Although the fused feature increased the training time, the improvement in its detection accuracy showed that this computational overhead was reasonable in MC detection tasks that required high precision.

3.2 Performance test of DTL-MD MC detection model

GAN, Extreme Gradient Boosting (XGBoost), and K-Nearest Neighbors (KNN) are selected as comparison algorithms. First, the classification ability of the MC detection model was evaluated, and the results are shown in Figure 9.



(a) Accuracy test (b) F1 test

Figure 9: Classification performance test results.

Figures 9(a) and (b) show the accuracy and F1 score of each model as a function of the threshold value. The F1 score can balance the precision and recall, and is particularly suitable for MC detection in unbalanced data sets, ensuring fewer missed detections and false positives. In Figure 9(a), when the threshold was 0.7, the accuracy of GAN, XGBoost, KNN, and DTL-MD were 91.2%, 91.3%, 87.9%, and 95.8%, respectively. In Figure

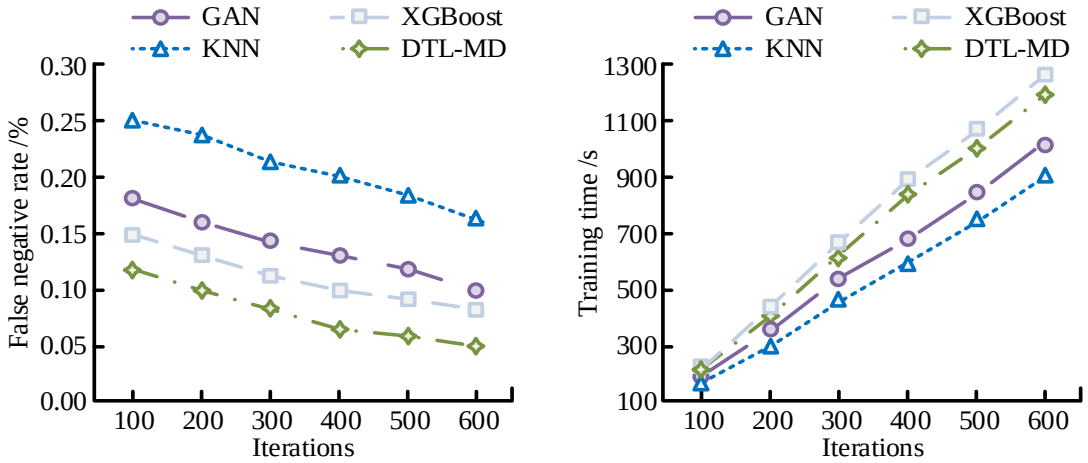
9(b), when the threshold value was 0.7, the F1 scores of each model were 92.1%, 91.9%, 85.9%, and 93.2% respectively. DTL-MD had the highest accuracy and F1 score at high thresholds, effectively reducing false positives through strict classification criteria and avoiding erroneous classification of MC. In contrast, the accuracy and F1 score of XGBoost and GAN were similar, but slightly lower than that of the DTL-MD model, which was due to their conservative decision boundary at high thresholds. Although the number of false positives was reduced, some more complex malicious samples was missed. To ensure the reliability of the results, the standard deviation and 95% confidence interval of the accuracy and F1 score of each model at a threshold of 0.7 were calculated, see Table 3.

Table 3: Standard deviation and confidence interval of accuracy and F1 score.

Model	Accuracy /%	F1 Score /%	Accuracy Std Dev	Accuracy confidence interval	F1 Score Std Dev	F1 Score confidence interval
GAN	91.2	92.1	±0.3	[90.9, 91.5]	±0.2	[91.8, 92.4]
XGBoost	91.3	91.9	±0.2	[91.1, 91.5]	±0.3	[91.6, 92.2]
KNN	87.9	85.9	±0.4	[87.5, 88.3]	±0.3	[85.6, 86.2]
DTL-MD	95.8	93.2	±0.2	[95.6, 96.0]	±0.2	[93.0, 93.4]

According to Table 3, the DTL-MD model showed smaller fluctuations in the standard deviation and confidence interval of the accuracy and F1 score, indicating that it had higher stability under different experimental conditions, higher accuracy and smaller fluctuation range.

Subsequently, the False Negative Rate (FNR) and training time of each model as a function of the number of iterations are shown in Figure 10.



(a) FNR test (b) Training time test

Figure 10: FNR and training time test results.

In Figures 10, the training time reflects the deployment efficiency of the model in practical applications, ensuring high accuracy while having good real-time and operability. In Figure 10 (a), when the number of iterations was 600, the false detection rates of each model were 0.10%, 0.08%, 0.16%, and 0.05%, respectively. In Figure 10 (b), when the number of iterations was 600, the training time of each model was 1049 s, 1282 s, 901 s, and 1257 s respectively. As the number of iterations increased, the false detection rate of the DTL-MD model decreased from 0.12% to 0.05%. In terms of training time, the DTL-MD model required 1257 seconds for training. More iterations of training improved the model's accuracy, but also led to an increase in computational overhead. Finally, the test results of each model under various sample sizes are in Table 4.

Table 4: Test outcomes with various sample sizes.

Model	Sample size	Samples per second	Memory usage /MB	Model size /MB	Computational complexity /GFLOPS
GAN	5000	319.7	1603.5	134.8	47.2
	10000	310.3	1645.7	139.1	48.1
	15000	299.6	1697.4	144.5	50.2
	20000	289.8	1746.2	149.3	52.3

XGBoost	5000	329.1	1449.3	124.6	42.8
	10000	314.7	1497.2	129.3	43.6
	15000	308.2	1547.6	134.9	45.1
	20000	299.4	1598.4	139.8	46.9
KNN	5000	229.4	1202.6	109.5	28.3
	10000	219.8	1246.5	113.6	29.5
	15000	209.3	1299.4	117.9	30.8
	20000	199.6	1349.8	122.1	31.9
DTL-MD	5000	199.7	1702.5	160.3	39.9
	10000	209.4	1804.6	164.2	41.8
	15000	219.8	1906.1	168.9	44.2
	20000	229.3	2008.3	173.4	47.1

In Table 4, computational complexity measures the computational efficiency of the model in processing data in GFLOPS (billion floating-point operations per second), reflecting the computational resources required by the model to complete a specific task. This value is related to the model architecture and the size of the dataset. The computational complexity reflects the computing resources required by the model to process each task. A lower FLOPs value means that the model has better scalability and can run efficiently on large-scale datasets or real-time applications. The processing speed of XGBoost reached 299.4 samples/second with a sample size of 20,000. In contrast, the processing speed of DTL-MD was relatively slow, especially at 20,000 samples, which was only 229.3 samples/s. The complex deep learning structure required more computing time and resources to complete the detection of MC. In terms of memory usage, DTL-MD consumed 2008.3 MB with a sample size of 20,000. In terms of model size, the size of DTL-MD remained at 160 MB. In terms of computational complexity, GAN reached 52.3 GFLOPS at 20,000 samples, while DTL-MD had a computational complexity of 47.1 GFLOPS at 20,000 samples, which is suitable for deployment in environments with sufficient computing resources. As the dataset increased, the processing speed of DTL-MD increased. For small datasets (such as 5,000 samples), its processing speed was slow, but the memory usage and computational complexity were low. As the sample size increased, although the training time and memory usage increased, DTL-MD still maintained high accuracy, especially in the 20,000 sample dataset, where it performed well and showed good generalization ability.

3.3 MC detection simulation experiment based on DTL-MD model

Furthermore, the research conducted simulation experiments on DTL-MD to test its practical application effect. The comparison models were selected from the more advanced models in the field, namely Malware GAN-enhanced Network (MGANet), Sequence GAN for Malware Detection (SeqGAN-Malware), and Deep Reinforcement Learning for Malware Detection (DRL-Malware). The research team constructed a self-built MC dataset containing 30,000 samples, of which 20,000 malware samples covered various types such as Trojans, viruses, and ransomware, and 10,000 normal software samples were from commonly-used applications. Firstly, the throughput and inference speed results of each model under the detection of four types of MC, namely Trojans, viruses, worms, and adware, are shown in Figure 11.

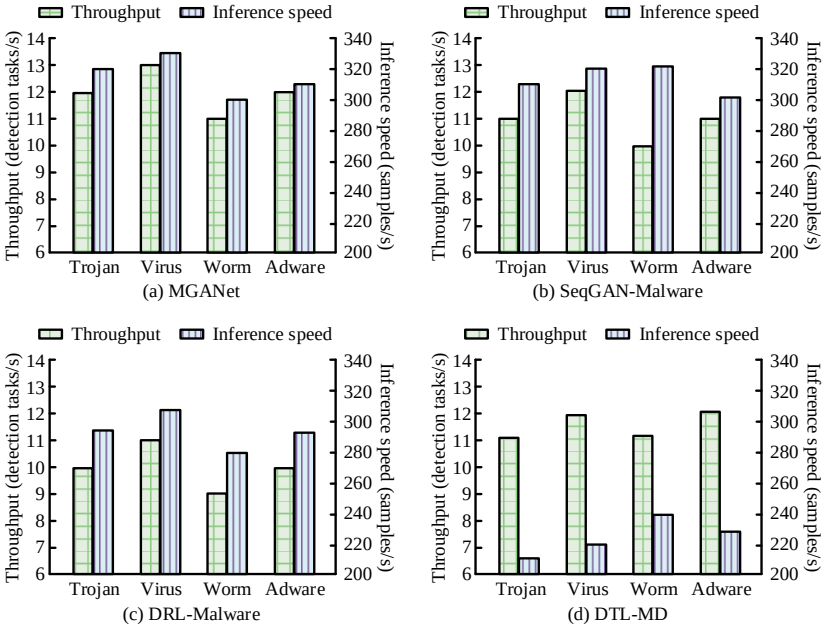


Figure 11: Throughput and inference time tests under different attack types.

Figures 11 (a)-(d) show the throughput and inference time test results under different attack types. Throughput measures the task processing capability of the model, while inference speed reflects the single-sample processing efficiency. Throughput is calculated as the number of tasks completed per second, while inference speed represents the number of samples processed per second. The two are closely related, because a task usually contains detection operations for multiple samples, so throughput is usually lower than inference speed.

In Figure 11 (a), the throughput of MGANet under Trojans, viruses, worms, and adware was 12, 13, 11, and 12 tasks/s, respectively, with an inference speed of 320 to 330 samples/s. In Figure 11 (b), the throughput of SeqGAN-Malware under Trojan, virus, worm, and adware was 11, 12, 10, and 11 tasks/s, respectively. In Figure 11 (d), the throughput of the DTL-MD model under Trojan, virus, worm, and adware was 11, 12, 11, and 12 tasks/s, respectively, and the inference time was 211, 225, 239, and 234 samples/s, respectively. The large computational resources and complex network structure resulted in a heavy computational burden during the inference process, and the throughput performance was moderate. In Figure 11 (c), the inference speed and throughput of DRL-Malware were slightly lower. The training process of deep reinforcement learning required the model to optimize performance through continuous policy updates, and the limitations of the learning strategy led to a decrease in inference speed. Subsequently, the same test dataset was used, containing five main types of MC (Trojans, viruses, worms, adware, and ransomware), and 1,000 samples of each type were randomly selected. The corresponding robustness and processing delay results are shown in Figure 12.

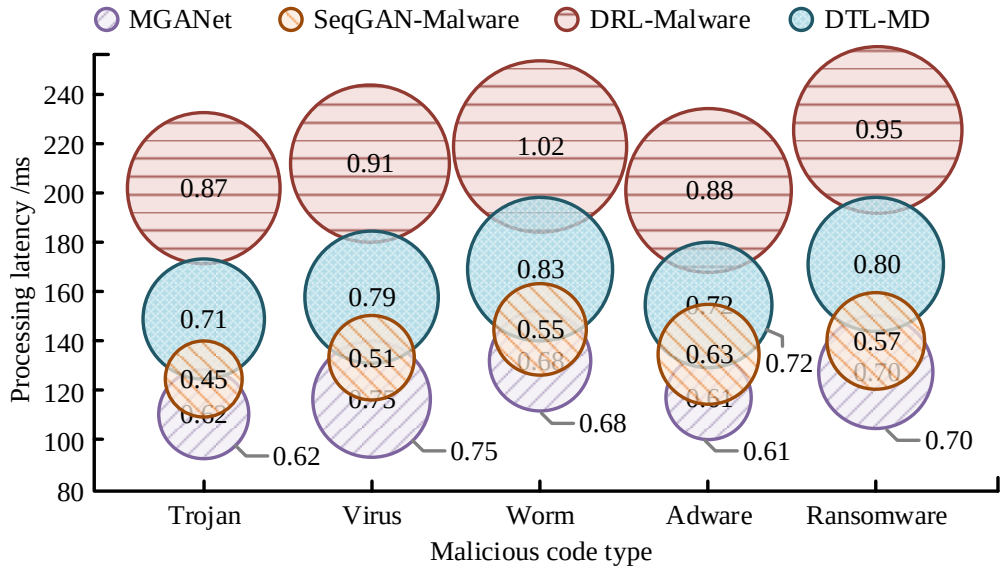


Figure 12: Robustness and processing delay results.

In Figure 12, the robustness of each type of MC was calculated through 10 experiments, and the standard deviation reflects the deviation between the experimental results and the average value. The smaller the standard deviation, the smaller the volatility of the model's detection results on the MC type, and the stronger the robustness. The robustness of DTL-MD under Trojan, virus, worm, adware, and ransomware code was 0.71, 0.79, 0.83, 0.72, and 0.80, respectively, indicating strong adaptability in the face of diverse MC. In terms of processing delay, the DTL-MD model had a longer inference time, with a response time of 172 ms under the worm type. Finally, the detection results of each model under different network bandwidths are shown in Table 5.

Table 5: Detection effect under different network bandwidths.

Network bandwidth	Model	Throughput (tasks/s)	Latency variance /ms	Computational efficiency (FLOPs/task)	Stability /SD
Low (100 Mbps)	MGANet	11	23	1.5	0.7
	SeqGAN-Malware	10	27	1.8	0.8
	DRL-Malware	9	30	2.0	0.9
	DTL-MD	10	29	1.9	1.1
Medium (500 Mbps)	MGANet	13	18	1.3	0.5
	SeqGAN-Malware	11	21	1.6	0.6

	DRL-Malware	10	23	1.8	0.7
	DTL-MD	11	22	1.7	0.8
High bandwidth (1 Gbps)	MGANet	14	15	1.2	0.4
	SeqGAN-Malware	12	17	1.5	0.5
	DRL-Malware	11	19	1.7	0.6
	DTL-MD	12	17	1.6	0.7

In Table 5, the performance of the model in various practical applications can be evaluated by the throughput, latency fluctuation, computational efficiency, and stability under different bandwidths. Stability is measured by calculating the standard deviation of the inference latency. A lower SD value indicates a more stable performance of the model, especially under low bandwidth conditions. DTL-MD was 10 tasks/s at low bandwidth. In terms of latency fluctuation rate, the latency fluctuation rate of MGANet under high bandwidth conditions was only 15 ms. The delay fluctuation rate of DTL-MD was 17 ms. In terms of computational efficiency, DTL-MD had a computational efficiency of 1.6 FLOPs/task under low bandwidth conditions. Complex models can lead to higher computational costs and longer processing times.

Finally, the study introduced the Deep Malware Detection Network (DMDN), Light Gradient Boosting Machine (LightGBM), and Adversarial Malware Detection Network (AMDN). The study also introduced the MalwareBazaar dataset and designed a cross-domain migration experiment. The results are shown in Table 6.

Table 6: Performance comparison in diverse datasets and cross-domain migration tests.

Dataset	Model	Precision (%)	Recall (%)	Robustness (Standard deviation)	Detection rate (Tasks/s)
MalwareBazaar	DTL-MD	91.8	92.3	0.73	12
	DMDN	89.7	90.5	0.81	10
	LightGBM	87.4	88.2	0.86	14
	AMDN	88.9	89.8	0.79	11
CICIDS 2017	DTL-MD	90.2	91.0	0.75	11
	DMDN	87.6	88.8	0.84	9
	LightGBM	85.3	86.5	0.89	13
	AMDN	86.7	87.9	0.82	10
Cross-domain Test	DTL-MD	88.3	89.7	0.75	10
	DMDN	85.9	87.0	0.81	8
	LightGBM	84.1	85.2	0.85	12
	AMDN	86.7	87.9	0.80	9

In Table 6, the Precision and Recall of DTL-MD on the MalwareBazaar dataset reached 91.8% and 92.3% respectively. Meanwhile, in the cross-domain migration experiment, the Recall of DTL-MD remained at 89.7% with a standard deviation of 0.75, which proved its robustness in dealing with changes in data distribution. In contrast, DMDN and AMDN performed second best in robustness, and although LightGBM had an advantage in detection rate, its detection accuracy was relatively low.

4 Discussion

In order to improve the accuracy of MC detection, the study designed a detection model DTL-MD based on OTL and tested its performance. Compared with the model proposed by Kim et al. in the literature [6], although it performed well in the accuracy of MC detection, it was relatively slow in processing speed and was suitable for relatively static scenarios. In the DTL-MD performance test, when the threshold was 0.7, the accuracy of DTL-MD was 95.8% and the F1 score was 93.2%. Although DTL-MD was slower than XGBoost and KNN in inference speed, its high accuracy made it more advantageous in MC detection tasks that require high reliability. In particular, on the 20,000 sample dataset, the memory usage of DTL-MD was 2008.3 MB and the computational complexity was 47.1 GFLOPS, showing its ability in computationally intensive tasks. The feature fusion-based method proposed by Wang et al. in the literature [8] showed a high accuracy in MC detection, but its computational complexity was high and the training and inference speeds were slow. In contrast, DTL-MD optimized computational complexity while maintaining high accuracy, making it still scalable in large data sets and real-time detection scenarios. In application tests, DTL-MD also performed well in the robustness of MC types such as Trojans, viruses, worms, and adware. Especially in low-bandwidth environments, DTL-MD had a throughput of 10 tasks/s and a latency fluctuation of only 15 ms, which was suitable for real-time MC detection.

The advantage of DTL-MD is that it is highly adaptable and can handle small data sets. It can also maintain good detection performance when there are fewer samples, showing strong generalization ability.

5 Conclusion

The study proposed an MC detection model DTL-MD that combines OTL and optimized feature selection strategies, and verified its effectiveness in MC detection. However, the model's throughput and latency volatility are high, and its real-time performance is poor in low-resource environments. In addition, its high computing requirements make its application on large-scale data sets face computing cost issues. In the future, the study will explore lightweight models, optimize the calculation process, and improve its computing efficiency through efficient feature extraction and pruning techniques to solve the problem of high computing overhead in large-scale data sets. Meanwhile, the experiment used a data set containing many known malicious samples. In the future, the study will introduce new MC samples to further verify the model's capabilities, especially its performance when detecting new samples.

Funding

This work was supported by the Natural Science Foundation of Ningxia Hui Autonomous Region in 2022, project number: 2022AAC03345.

References

[1] Wang R, Gao J, Huang S. AIHGAT: A novel method of malware detection and homology analysis using assembly instruction heterogeneous graph. International Journal of Information Security, 2023, 22(5): 1423-1443. DOI: 10.1007/s10207-023-00699-7

[2] Li F, Ren J. Suppression of MC Propagation in software-defined networking. Wireless Personal Communications, 2024, 135(1): 493-516. DOI: 10.1007/s11277-024-11065-8

[3] Liu T, Neware R, Bhatt M W, Shabaz M. A study on detection and defence of MC under network security over biomedical devices. The Journal of Engineering, 2022, 2022(11): 1041-1049.

[4] Dam K H T, Touili T. Extracting malicious behaviours. International Journal of Information and Computer Security, 2022, 17(3): 365-404. DOI: 10.1049/tje2.12153

[5] Cui Z, Zhao Y, Cao Y, Cai X, Zhang W, Chen J. Malicious code detection under 5G HetNets based on a multi-objective RBM model. IEEE Network. 2021, 35(2): 82-87. DOI:10.1109/MNET.011.2000331.

[6] Kim H W. A study on countermeasures by detecting trojan-type downloader/dropper MC. International Journal of Advanced Culture Technology, 2021, 9(4): 288-294. DOI: 10.17703/IJACT.2021.9.4.288

[7] Kim J, Lee S. Malicious behavior detection method using API sequence in binary execution path. Tehni Čki Vjesnik, 2021, 28(3): 810-818. DOI: 10.17559/TV-20210202132203

[8] Wang Z, Wang W, Yang Y, Han Z, Xu D, Su C. CNN-and GAN-based classification of MC families: a code visualization approach. International Journal of Intelligent Systems, 2022, 37(12): 12472-12489. DOI: 10.1002/int.23094

[9] Li S, Jiang L, Zhang Q, Wang Z, Tian Z, Guizani M. A malicious mining code detection method based on multi-features fusion. IEEE Transactions on Network Science and Engineering. 2022, 10(5):2731-2739. DOI:10.1109/TNSE.2022.3155187.

[10] Li H, Jin Y, Chai T. Evolutionary multi-objective Bayesian optimization based on multisource online transfer learning. IEEE Transactions on Emerging Topics in Computational Intelligence, 2023, 8(1): 488-502. DOI: 10.1109/TETCI.2023.3306351

[11] Cherifi D, Djaber A, Guedouar M E, Feghoul A, Chelbi Z Z, Ouakli A A. Covid-19 detecting in computed tomography lungs images using machine and transfer learning. Informatica. 2023, 47(8). DOI: 10.31449/inf.v47i8.4258

[12] Cui Z. Combining the SSD target identification algorithm with the 3D-CNN architecture for transfer learning research in basketball training. Informatica. 2024, 48(18). DOI: 10.31449/inf.v48i18.6454

[13] Qin P, Zhao L. An online transfer learning framework for cell SOC online estimation of battery pack in complex application conditions. IEEE Transactions on Transportation Electrification, 2023, 10(3): 5974-5986. DOI: 10.1109/TTE.2023.3324822

[14] Lu H, Jin C, Helu X, Du X, Guizani M, Tian Z. DeepAutoD: Research on distributed machine learning oriented scalable mobile communication security unpacking system. IEEE Transactions on Network Science and Engineering, 2021, 9(4): 2052-2065. DOI: 10.1109/TNSE.2021.3100750

[15] Khan S, Nauman M. Interpretable detection of malicious behavior in windows portable Executables using Multi-Head 2D transformers. Big Data Mining and Analytics, 2024, 7(2): 485-499. DOI: 10.26599/BDMA.2023.9020025

[16] Gurjar A, Voditel P. Transfer learning: a paradigm for machine assisted knowledge transfer. ECS Transactions, 2022, 107(1): 7179-7188. DOI: 10.1149/10701.7179ecst

[17] Dai S, Meng F. Addressing modern and practical challenges in machine learning: A survey of online federated and transfer learning. Applied Intelligence, 2023, 53(9): 11045-11072. DOI: 10.1007/s10489-022-04065-3

[18] Zhu Z, Lin K, Jain A K, Zhou J. Transfer learning in deep reinforcement learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(11): 13344-13362. DOI: 10.1109/TPAMI.2023.3292075

[19] Solís M, Calvo-Valverde L A. Performance of deep Learning models with transfer learning for multiple-step-ahead forecasts in monthly time series. Inteligencia Artificial-Iberoamerical Journal of

Artificial Intelligence, 2022, 25(70): 110-125. DOI: 10.48550/arXiv.2203.11196

[20] Belouadah, E, Adrian P, Ioannis K. A comprehensive study of class incremental learning algorithms for visual tasks. Neural Networks, 2021, 135: 38-54. DOI: 10.1016/j.neunet.2020.12.003

[21] Minoofam S A H, Bastanfard A, Keyvanpour M R. TRCLA: a transfer learning approach to reduce negative transfer for cellular learning automata. IEEE transactions on neural networks and learning systems, 2021, 34(5): 2480-2489. DOI: 10.1109/TNNLS.2021.3106705

GAN-Based Financial Data Generation and Prediction: Improving The Authenticity and Prediction Ability of Financial Statements

Feng Qi
School of Management, Guangdong University of Foreign Studies South China Business College Guangzhou 510545, China
E-mail: as_453520854@163.com

Keywords: GAN, financial data, financial statements, authenticity, prediction ability

Received: October 16, 2024

The research on the mining algorithm of financial data association relationship mainly explores a certain kind of association relationship in depth, but it is not suitable for the attributes and characteristics of financial data itself, and there are few comprehensive analysis and application for financial data association relationship mining. In order to overcome the above problems, this paper proposes a financial data generation and prediction model based on GAN. Based on WGAN network, this paper improves the authenticity of the generated virtual samples by increasing the cyclic consistency loss term and selecting intermediate samples for the generated samples to optimize the generated model. At the same time, in the system, this paper adopts intelligent data analysis research method, mines the correlation of different dimensions of financial statement data, and presents the mining results by using the correlation visualization method, so as to realize the risk assessment and trend prediction of enterprise financial status. In terms of the overall recognition accuracy of the model, the random forest model has the highest accuracy rate, reaching 74.46%. In terms of recall rate, the GBDT model is slightly higher than the random forest model, with a recall rate of 73.43%, but its accuracy rate, F1 value and AUC value are slightly lower than the random forest model. According to the comprehensive experimental analysis results, it can be seen that the model proposed in this paper has good performance in the authenticity analysis and prediction of financial data. Generally speaking, the model proposed in this paper provides a reliable tool for the authenticity audit of financial data, and can provide a reference for the formulation of subsequent schemes and policies through financial data prediction.

Povzetek: A deep transfer learning-based model (DTL-MD) enhances malicious code detection using ResNet50V2, GAN-generated variants, and online learning, achieving 95.8% accuracy and improving detection speed and robustness against evolving threats.

1 Introduction

After the formation of the capital market, the financial analysis system has been gradually improved, and the regulatory agencies have clearly defined and required the scope, frequency and caliber of financial data that enterprises need to disclose. At the same time, internal and external audits will severely punish the fraud of financial data, so that the financial statement data publicly disclosed by enterprises can truly reflect the operating status of enterprises, and the scope of financial analysis is correspondingly expanded to analyze the financial status, operating results and cash flow of enterprises. The traditional analysis method is to quantitatively or qualitatively evaluate the financial status of an enterprise in the field of financial accounting according to the key indicators formulated by the enterprise's solvency, operational ability and profitability, as well as the year-on-year situation of the indicators, but the ability to predict financial risk exposure and financial development trend is weak. Therefore, relevant experts began to try to use more mature artificial intelligence and data mining methods for financial analysis and prediction, but there are few studies on judging or predicting the operating conditions of enterprises by mining the association relationship between enterprise financial data [1].

The association relationship between enterprise financial data will produce various manifestations according to different data objects. Among them, the distance attribute of enterprise financial indicators in different dimensional spaces is the spatial association relationship of enterprise finance, and enterprises with closer distance in multi-dimensional spaces have higher financial similarity. The group dependence attribute between financial indicators in all enterprises is the static time association relationship of financial indicators. Based on the frequently occurring financial indicator groups, that is, the frequent item sets of financial indicators, the abnormal financial data of enterprises can be found [2]. The related attribute of the historical trend of financial indicators in different industries is the dynamic time correlation between industries. The change

of the financial situation of upstream industries will have an impact on downstream industries through a period of transmission, and then the financial indicators of upstream and downstream related industries will show positive or reverse correlation in the time trend. Through the analysis of trend correlation, the future financial situation of downstream industries can be predicted. According to different association relationships, predecessors have developed corresponding algorithms for data mining, but at present, various algorithms mainly explore a certain association relationship in depth, and there are few comprehensive analysis and applications for all association relationships [3].

In order to overcome the above problems, this paper proposes a financial data generation and prediction model based on GAN. Based on WGAN network, this paper improves the authenticity of the generated virtual samples by increasing the cyclic consistency loss term and selecting intermediate samples for the generated samples to optimize the generated model. At the same time, in the system, this paper adopts intelligent data analysis research method, mines the correlation of different dimensions of financial statement data, and presents the mining results by using the correlation visualization method, so as to realize the risk assessment and trend prediction of enterprise financial status.

2 Related works

In view of the data results mined by association relationships, some scholars further adopt visual analysis methods to release the value of data, make it easier to understand and make the essence of things more prominent. They also integrate visual interactive interfaces into the process of data mining, and combine expert experience in the execution steps of the algorithm, so as to improve the interpretability of the algorithm and make the data mining results more match the business reality. However, the existing visual analysis methods of association relationships cannot well adapt to the characteristics of financial data. At present, most of the various methods focus on the association relationships themselves, and the visual analysis of financial data needs to be further combined with the analysis objectives of indicator trends, indicator proportions, group changes, group order, etc. Moreover, the existing methods lack a comprehensive visual analysis solution for financial data association that matches the above objectives [4].

Scholars have used flow chart research, management evaluation scoring, stage analysis, etc., but qualitative research can't be applied to the changeable needs of financial risk early warning, and its accuracy can't meet the needs of enterprises. Therefore, scholars began to study the financial indicators of enterprises by quantitative methods, and developed univariate early warning model, multivariate early warning model, Logistic linear regression model, neural network analysis model and other methods [5].

Reference [6] put forward that a single financial index should be used as the judgment basis of financial risk early warning. By comparing and analyzing the results of a single financial index of enterprises with financial risks, it finally found that the two financial indexes, return on net assets and property rights ratio, had the best effect on financial early warning, and put forward a univariate early warning model. Reference [7] added indicators such as cash flow debt ratio and asset-liability ratio to the model research. Then, reference [8] puts forward new suggestions for improving financial indicators, which contributes to the early warning model of financial indicators. In addition, reference [9] put forward in the research that the effects of asset-liability ratio, return on total assets and working capital ratio are the most effective.

With the deepening of early warning research, many scholars have found that a single index can't fully reflect the financial risks of enterprises, and the accuracy of univariate financial early warning model still can't meet the needs of enterprises. Reference [10] proposed to apply multiple financial indicators to the research of financial risk early warning model, optimized five optimal comprehensive indicators among 22 financial indicators, calculated the weight coefficient of each indicator, established the Z-value model, and achieved great achievements. The Z-value model has made great achievements in the follow-up enterprise financial risk early warning analysis. Reference [11] put forward the concept of multivariate linearity, which proved that the multivariate linear model has higher accuracy than the multivariate early warning model and is more suitable for the existing enterprise financial early warning.

The logistic regression model is developed from the idea of multivariate linearity. Through the Logistic linear regression model, reference [12] performed linear analysis in combination with the current economic environment and model characteristics, and believed that financial risk early warning should accumulate experience with the increase of research samples and quantity, and the early warning results will become more accurate. After that, scholars put forward that the combination of factor analysis and Logistic regression model can more accurately reflect the possible financial risks in financial indicators, and reduce the excessive weight caused by the repetition of index factors, which also proves that it is more accurate and scientific.

With the rapid development of artificial intelligence, with the powerful technical support of Internet big data, neural network began to be used in financial risk early warning. Reference [13] proposed to use the empirical risk minimization principle of neural network to early warning enterprises. At the same time, with the rapid development of computer technology, the prediction effect of neural network early warning model based on machine learning technology is getting better and better. Then, through the empirical analysis of past data, the

computer can quickly summarize the abnormal rules of financial risk companies' indicators and reflect them, and its accuracy is far better than that of previous models. Moreover, it can quickly adapt to a single category of samples, but it is not possible to accurately analyze the situation where the number of samples is small and the data is insufficient.

Financial indicators are the most widely used indicators in financial early warning models, and they are also indicators that objectively reflect the operating and financial status of an enterprise. Furthermore, it is easy to obtain, so as early as when the univariate early warning model was put forward, it received enough attention. In addition, the selection of financial indicators has also changed from a single indicator such as asset-liability ratio and equity ratio at the beginning to multiple indicators in parallel later, and then to the ability to classify specific financial indicators into multiple indicators later, so as to further improve the efficiency of the model [14].

With the improvement of financial risk early warning model system based on financial indicators, scholars have found that many external factors such as industry environment, national policies, competitive environment and other external factors will greatly affect the early warning results of the early warning model. Therefore, they put forward to add non-financial indicators to the financial early warning model, among which the addition of non-financial indicators such as company ownership structure, external economic indicators, market industry development status, etc. greatly improves the accuracy of the model. After that, stock fluctuation, inflation rate, equity concentration, etc. were all included in non-financial indicators, and the prediction effect of the model was further improved. It is not difficult to see that non-financial indicators play an important role in various enterprise financial early warning models, and their own early warning research value cannot be underestimated [15].

Regarding the purpose and function of financial diagnosis, by finding another way to focus on the strategic perspective, reference [16] pointed out that financial diagnosis must stand at the strategic height to play a role in the strategic development of the company. Reference [17] indicated that the purpose of financial diagnosis is to improve the company's ability to obtain operating profits, control financial risks and operational risks, and help the company better operate and manage. Reference [18] hold that financial diagnosis is a dynamic rather than a static process, strategy and financial diagnosis are intertwined, and financial diagnosis needs to focus on strategy, and strategy also needs to be matched with finance to achieve the desired effect. On the main content of financial diagnosis, reference [19] pointed out that financial diagnosis includes three major activities: operation, investment and financing. Only by starting from these three aspects and supplementing suitable methods can we evaluate and calculate the situation of enterprises. Reference [20] proposed that financial diagnosis should also include prospect diagnosis, establish an interval evaluation system, use the discounted cash flow method in financial management to evaluate prospects, and optimize the content of financial diagnosis. Reference [21] pointed out that massive data without aggregation will limit financial diagnosis, so useful information should be extracted through data mining technology, which can better ensure the accuracy of financial diagnosis results and apply it to enterprise decision-making.

In summary, the contents of the relevant work are shown in Table 1.

Table 1: Summary of related work.

Serial Number	Research Method	Shortcoming
1	Visualization of Association Relationships	Lack of adaptability
2	Single indicator research	Insufficient accuracy
3	Multi indicator research	Insufficient intelligence
4	Logistic regression model	Underfitting problem
5	Neural Network Model	Not suitable for small samples
6	Financial diagnosis	Insufficient intelligence and low accuracy

In summary, traditional financial data analysis methods have the problem of insufficient intelligence. Several common intelligent algorithms are prone to underfitting in financial data processing and require a large amount of data for training, which is not suitable for financial data analysis of small sample data. Therefore, this paper combines decision tree classification algorithm to process financial management data in universities, constructs a four-dimensional dynamic investment decision game system, improves the risk management

effect of financial data in universities, and enhances the accuracy of financial risk warning.

3 Zero-sample generation model based on cyclic invariance

Because there is serious domain offset and pivoting problems in the embedding model and the data imbalance problem in the zero-sample problem itself, based on WGAN network, this paper improves the authenticity of the generated virtual samples by increasing the cyclic consistency loss term and selecting intermediate samples for the generated samples to optimize the generated model.

3.1 Zero-sample generation model based on cyclic invariance

In this paper, n marked visible class samples are set, which have financial data characteristics $X \in ¡^{d\times n}$ and semantic description $A \in ¡^{m\times n}$ at the same time. Zero-sample learning aims to identify n_u invisible class samples $X_u \in ¡^{m\times n_u}$ that only have semantic attribute $A_u \in ¡^{d\times n_u}$ at the time of training. Y and Y_u are labels of X and X_u, respectively, and in a zero-sample environment, $Y \cap Y_u = \varnothing$ exists. It is assumed that the labels of visible class and invisible class are C and C_u respectively. In the traditional zero-sample learning, it is only necessary to correctly identify X_u in C_u, but under the generalized zero-sample condition, it is necessary to search and identify in $C \cup C_u$ space. Moreover, each semantic description a is a description of a category *y*. Formally, $\{X, A, Y\}$ and $\{A_u, Y_u\}$ are given to train the model, the goal of zero-sample learning is to learn the mapping function $f : X_u \to Y_u$, and the goal of generalized zero-sample learning is to learn the mapping function $f : \{X, X_u\} \to Y \cup Y_u$ [22].

The underlying generative model used is the WGAN model. The visible class sample $\{X, A, Y\}$, the attribute A_u of the invisible class, and the random noise $z \sim N(0,1)$ are given. The GAN generator G synthesizes false features through input class embedding a and noise z. At the same time, the GAN discriminator D takes the real financial data x and the features of the generated financial data $G(z,a)$ as inputs to distinguish whether the input features are true or false. The loss function of WGAN is as follows:

$$L_{WGAN} = E\left[D(x,a)\right] - E\left[D(\tilde{x},a)\right] - \lambda E\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x},a)\right\|_2 - 1\right)^2\right] \tag{1}$$

Among them, $\tilde{x}$ is the generated false sample, $\hat{x}$ is the interpolation of x and $\tilde{x}$, and $\hat{x} = \alpha x + (1-\alpha)\tilde{x}, \alpha \in (0,1)$.

Considering the diversity of generated samples, the model will generate multiple intermediate samples for each category here. Therefore, the samples will be divided into multiple clusters by clustering, and the central sample of each cluster will be calculated respectively. $\{x_1^c, x_2^c, \mathrm{L}, x_k^c\}$ is set as k clusters of class c, and the intermediate sample is $S^c = \{S_1^c, S_2^c, \mathrm{L}, S_k^c\}$.

$$S_k^c = \frac{1}{\left|x_k^c\right|} \sum_{x_i \in x_k^c} x_i \tag{2}$$

Similarly, for the generated virtual samples $\tilde{x}_i$, intermediate samples can also be defined:

$$\tilde{S}_k^c = \frac{1}{\left|x_k^c\right|} \sum_{\tilde{x}_i \in \tilde{x}_k^c} \tilde{x}_i \tag{3}$$

In order to encourage that each generated sample should be close to at least one intermediate sample S^c, this paper introduces a regularization term to deal with a single sample and has the following form, where n_1 is the number of generated samples and k is the number of intermediate samples per class.

$$L_{R1} = \frac{1}{n_1} \sum_{i=1}^{n_1} \min_{j\in[1,k]} \left\|\tilde{x}_i - S_j^c\right\|_2^2 \tag{4}$$

At the same time, it should also be ensured that the intermediate samples of each class should be close to their real samples, so that the samples of the whole cluster are close to the real samples. Then, a regularized cluster sample is introduced, and its form is as follows. Among them, C is the total number of categories.

$$L_{R2} = \frac{1}{C} \sum_{i=1}^{C} \min_{j\in[1,k]} \left\|\tilde{S}_j - S_j^c\right\|_2^2 \tag{5}$$

At this stage, through the above two regularizations, the correlation between the generated sample and the real sample is guaranteed from the financial data feature domain. However, in this process, the relevant semantic description is used to generate the corresponding virtual samples, so the cyclic consistency loss is introduced into the model, and the correlation between the generated virtual samples and the real samples is further measured from the aspect of semantic description, so as to further improve the authenticity of the generated samples. Among them, the cyclic consistency loss converts the generated virtual samples into semantic description information by adding a regressor R after the discriminator, calculates the loss between the virtual semantic information and the real semantic information, and thus feeds it back to the generator to optimize the generation process. Its calculation form is as follows:

$$L_{cyc} = E\left(\left\|a - R(G(a,z))\right\|_2^2\right) \tag{6}$$

After the generator is trained to generate enough financial data features for the invisible class, zero-sample learning can be transformed into a supervised learning task in the traditional sense.

In addition, this model combines two softmax classifiers into a cascade classifier to perform

classification tasks. In some other zero-sample classification models, a generated virtual financial data sample is used to train a softmax classifier to correctly classify the true invisible class samples at the time of testing. The loss of the classifier is as follows:

$$L_{cls} = -E\left(log\, P\left(y|x;\theta\right)\right) \tag{7}$$

Among them, $P\left(y|x;\theta\right)$ is the probability that the financial data sample x is correctly predicted as class y. The parameter θ is obtained by training with the following formula. The parameter T is related to the task type. When it is a traditional zero-sample classification task, T is an unknown class sample used for testing. However, when it is a generalized zero-sample classification task, T is the set of unknown class samples and known class samples used for testing.

$$P\left(y|x;\theta\right) = \frac{exp\left(\theta_y^T x\right)}{\sum_{i=1}^{N} exp\left(\theta_i^T x\right)} \tag{8}$$

Before this, a softmax classifier is added to this model, which is used to evaluate the confidence of the classifier. Since the output of the softmax layer is a probability vector, the uncertainty of the measurement result can be determined by the entropy of this result. Therefore, the sample with low classification entropy can be used as a reference to classify other unseen samples. The calculation method is as follows.

$$E\left(y\right) = -\sum_{c=1}^{C} y_c\, log\, y_c \tag{9}$$

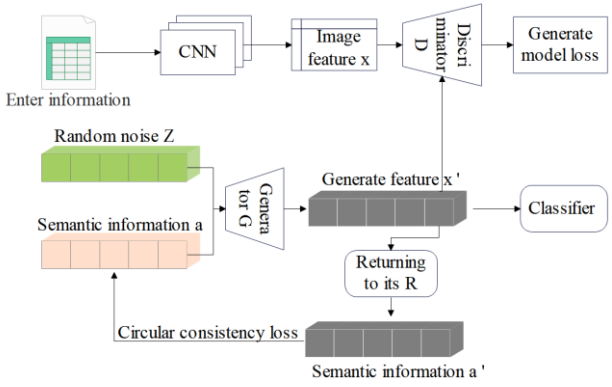


Figure 1: Model structure diagram.

The specific model structure is shown in Figure 1. The overall model loss function is as follows:

$$L = L_{WGAN} + \alpha L_{CLS} + \gamma L_{R1} + \gamma L_{R2} + \beta L_{CYC} \tag{10}$$

The evaluation index of zero-sample learning has different calculation methods under different settings.

In the case of traditional zero-sample learning, the same evaluation criterion, TOP-1 accuracy rate, is used to evaluate the accuracy of the model as the normal machine learning single-label financial data classification. However, because the number of samples in each category in the zero-sample dataset is not balanced, the average accuracy cannot be used for the overall dataset to evaluate the model. At present, in various zero-sample learning methods, the class average accuracy is usually used as the zero-sample evaluation standard. The average accuracy rate of each category is calculated first, and then the average accuracy rate of each category is calculated by finding the average value of the sum of all categories. The calculation formula is as follows

$$A_{cc} = \frac{1}{M}\sum_{i=1}^{M} Acc_i \tag{11}$$

Among them, M is the number of unseen classes and Acc_i is the classification accuracy on the i-th invisible class.

In the case of generalized zero-sample learning, the test set includes not only invisible class samples, but also some visible class samples. Therefore, this paper uses the harmonic mean accuracy proposed by Xian et al. as the evaluation index of generalized zero-sample learning, and the calculation formula is as follows

$$H = \frac{2 \times Acc_{y^s} \times Acc_{y^u}}{Acc_{y^s} + Acc_{y^u}} \tag{12}$$

Among them, Acc_{y^s} and Acc_{y^u} represent the class average accuracy of visible and invisible classes in the test set, respectively.

3.2 Attention-based FS-f-VAEGAN-D2 zero-sample learning method

In zero-sample learning, both VAE and GAN have certain defects as generators. The financial data generated by VAE model is rather fuzzy, and the expression effect of some more complex data is poor. For the GAN model, the input of the generator is random Gaussian noise, which will be difficult in the training process and the model will be difficult to converge. Therefore, this paper uses VAE-GAN model to build the system, including an encoder, a decoder/generator, and a discriminator. The VAE-GAN model diagram is shown in Figure 2.

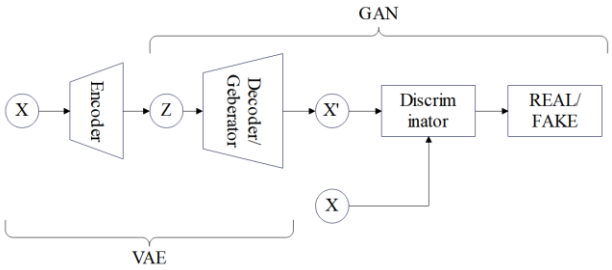


Figure 2: VAE-GAN model diagram.

One advantage of the GAN model is that its discriminator network can be measured by the similarity of financial data to distinguish them from "non-financial data". Specifically, since the reconstruction error of the

elements in the VAE model is not sufficient for the expression of financial data invariance, the VAE-GAN model replaces the VAE reconstruction (expected log-likelihood) error term with the reconstruction error expressed in the GAN discriminator. The end result is a method that combines the features of GAN as a high-quality generative model and VAE as a method to generate data encoders into latent space z.

The attention module proposed in this paper is shown in Figure 3. Firstly, financial data features are fed into a $1*1$ convolutional layer with three different weight values to obtain three attention features. After transposing one of the attention features, it is multiplied by the other attention feature softmax gets an attention map, and the calculation formula of the attention map is as follows

$$\beta = \frac{exp\left(\left(W1x_i\right)^T \times \left(W2x_i\right)\right)}{\sum_{i=1}^{N} exp\left(\left(W1x_i\right)^T \times \left(W2x_i\right)\right)} \tag{13}$$

Finally, the attention map is multiplied by the last attention feature, and then input into a $1*1$ convolutional layer again, and finally the financial data feature x' with attention is obtained. Its calculation formula is as follows

$$x' = W\left(\sum_{i=1}^{N} \beta\left(W3x_i\right)\right) \tag{14}$$

Firstly, the FS-f-VAEGAN-D2 model is briefly introduced. The model structure diagram is shown in Figure 4. In the case of inductive model, based on the VAEGAN model, VAE and GANs are combined to use a shared decoder and generator to enhance the feature generator. The definition of zero-sample learning in this paper is consistent with the previous description.

For the first VAE-GAN model, the advantages of the VAE model and the GAN model can be utilized to learn complementary information to generate features. When the target data follows a complex multi-modal distribution, VAE loss and GAN loss are able to capture different modalities of the data. It mainly trains the whole model generator and discriminator through visible class samples. Among them, the loss function of the VAE model is as follows:

$$L_{VAE}^{S} = KL\left(q\left(z|x,a\right)\|p\left(z,a\right)\right) - E_{q(z|x)}\left[logp\left(x|z,a\right)\right] \tag{15}$$

The loss function of the GAN model is as follows, where $\%$ is the generated sample of the visible class, and $\hat{x}$ is the interpolation of x and $\%$.

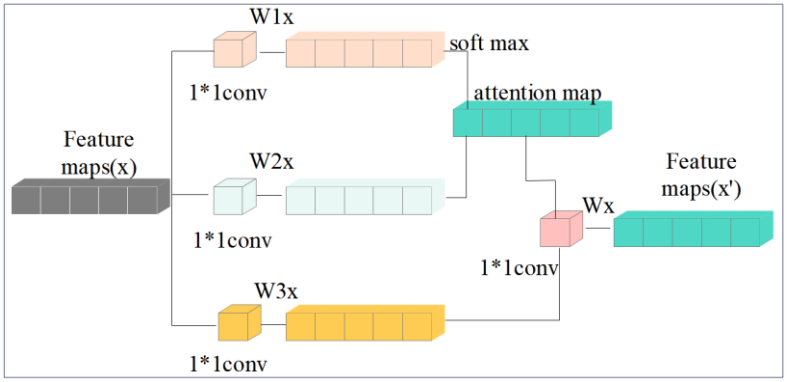


Figure 3: Attention module diagram.

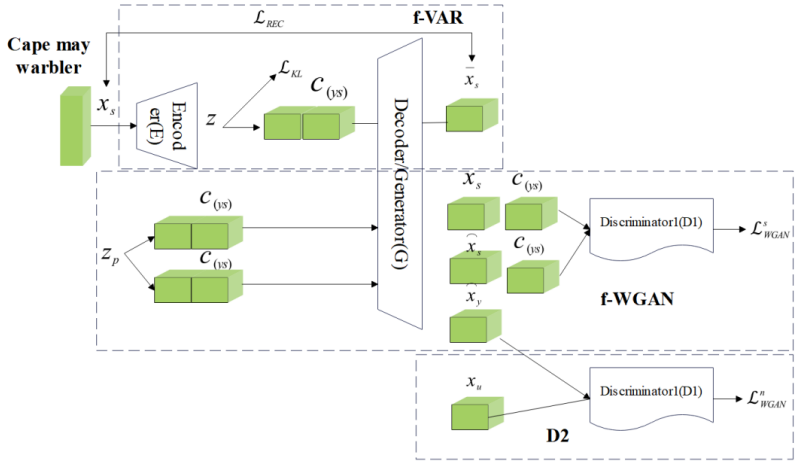


Figure 4: FS-f-VAEGAN-D2 model diagram.

$$L_{WGAN}^{S} = E\left[D(x,a)\right] - E\left[D(\%,a)\right] - \lambda E\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x},a)\right\|_2 - 1\right)^2\right] \tag{16}$$

Under the inductive method, the objective function of the overall f-VAEGAN model is as follows. At this time, the model only contains one discriminator D1.

$$L_{VAEGAN}^{S} = L_{VAN}^{S} + \gamma L_{WGAN}^{S} \tag{17}$$

When the unlabeled samples of the invisible class are available at this time in the case of the direct push model, the model adds an additional discriminator D2 to distinguish the real features of the unseen class from the generated virtual features. By inputting the true unlabeled features of the unseen class into the discriminator, the manifold structure of the unseen class can be obtained, thus generating more true unseen class features. The loss of this discriminator is as follows. Among them, % is the generated sample of the visible class, and $\hat{x}$ is the interpolation of x and %.

$$L_{WGAN}^{n} = E\left[D(x,a)\right] - E\left[D(\%,a)\right] - \lambda E\left[\left(\left\|\nabla_{\hat{x}} D(\hat{x},a)\right\|_2 - 1\right)^2\right] \tag{18}$$

Under the direct deduction method, the objective function of the whole FS-f-VAEGAN-D2 is as follows:

$$\min_{G,E} \max_{D_1,D_2} L_{WGAN}^{n} + L_{WGAN}^{n} \tag{19}$$

In the modified model, the attention module mentioned herein is added to the FS-f-VAEGAN-D2 model (Figure 5). Before adding the generated visible and invisible financial data features and the real financial data features to the discriminator network, the corresponding financial data features with attention are generated by the attention module. Then, the features of financial data after selective attention are input into the discriminator to improve the discrimination ability of the discriminator. Furthermore, by inputting unlabeled samples into the attention module to generate selectively noticed financial data features, the ability to properly classify financial data features can be improved in the discriminator. This is very important in the direct inference model. If the unlabeled samples are classified into the wrong category, it will affect the subsequent generation process and the accuracy of the model will be greatly reduced. Next, this paper will analyze the performance of the model after adding attention mechanism through experiments.

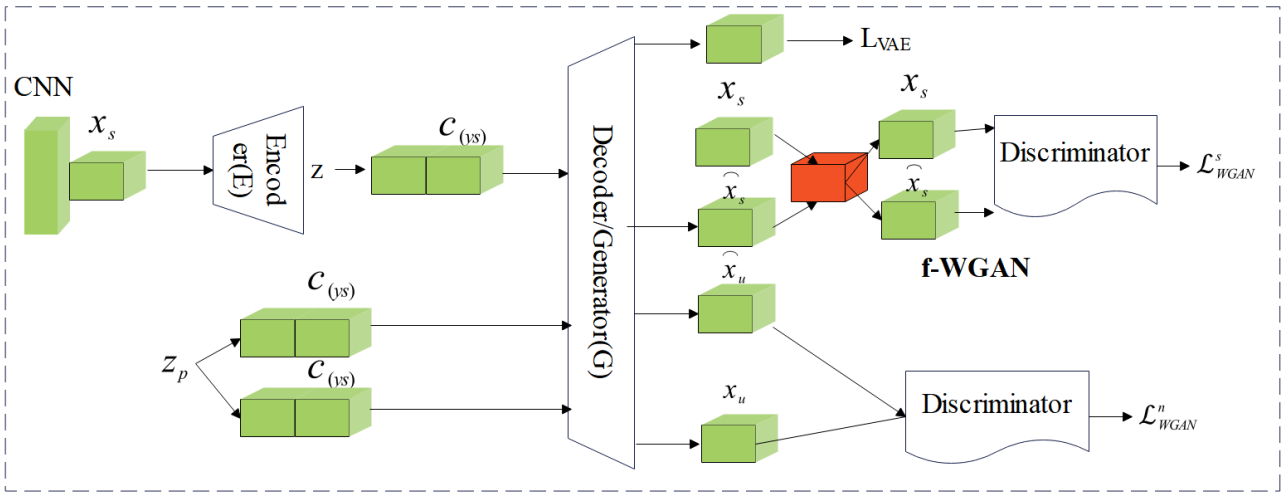


Figure 5: Schematic diagram of FS-f-VAEGAN-D2 model after adding attention mechanism.

4 System construction and test

4.1 System model

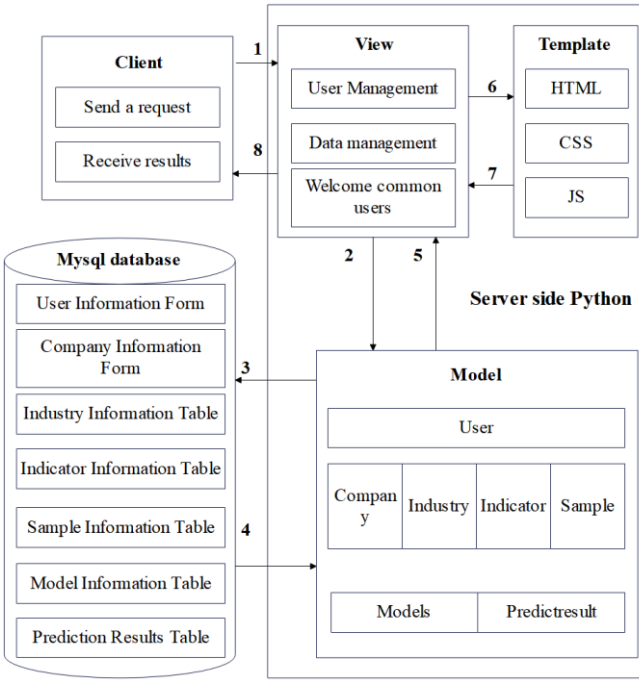


Figure 6: System architecture diagram

This system uses Python language for programming and development in PyCharm. The database adopts the overall architecture of MySQL system and is divided into three parts: client, server and database. The client and server interact through URLconf, and the server operates the database by calling the Model corresponding to the data table one-to-one. The system architecture design is shown in Figure 6. The client page display part is implemented using the Bootstrap30 framework, and the server is implemented using the Django framework.

Django is a Web framework based on MVT design pattern, in which M stands for Model, which is used to encapsulate the Model for accessing the database, V stands for View, which is responsible for receiving requests forwarded by URLconf, processing business logic, accessing the Model and returning processing results, T stands for Template, which is responsible for data display and encapsulates HTML, CSS and other files. In the server based on MVT mode, View receives the request sent by the client, calls the corresponding business logic processing method to respond, and operates the classes in the Model if it needs to access the database. Then, Model defines a class corresponding to the data table one-to-one, and operates the database by instantiating the objects of the class. Finally, Template receives the parameters passed by View, embeds them into the front-end page, and completes the data display

work.

According to the analysis of system functional requirements, this paper divides the system functions into four modules: login module, user management module, basic data management module and model management module. The basic data management module includes four sub-modules: company management, index management, industry management and sample management. The specific design is shown in Figure 7.

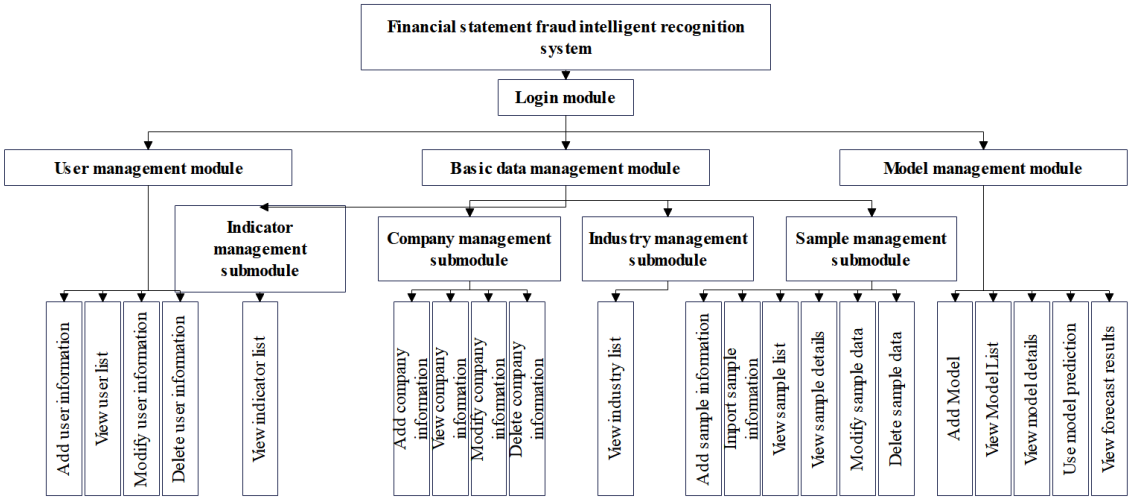


Figure 7: System functional module design.

The validation of the circular consistency term and the contribution of intermediate sample clustering to the improvement of authenticity in financial warning models is mainly achieved through a series of rigorous steps and methods.

For the validation of loop consistency items, the following steps can be followed:

(1) Clear cycle consistency indicator: In financial warning models, cycle consistency usually refers to the consistency between the model's predicted results and actual financial data. To verify this, it is necessary to first determine specific metrics for measuring consistency, such as accuracy, recall, F1 score, etc.

(2) Historical data validation: The cycle consistency of the model is verified using known historical financial data. The model's prediction results are compared with the actual historical data, and consistency indicators are calculated to evaluate the performance of the model on the real data.

(3) Sensitivity analysis: A sensitivity analysis is performed to observe the response of the model prediction results to changes in the input data. This helps to understand the stability and consistency of the model in different situations.

This paper analyzes parameters such as accuracy and recall, validates the model using an actual dataset, and discusses and analyzes it in conjunction with actual data

The data used in this paper are all from the penalty announcements published by China Securities Regulatory Commission, Stock Exchange and Securities Regulatory Bureau for fraudulent behaviors of listed companies. The fraud announcements of listed companies publicly criticized or punished from 2012 to 2023 published by CSMAR contain 16 types of violations, which are diverse and cross-cutting. The higher the complexity of the violation types, the greater the impact on the data and the empirical effect. Therefore, this paper focuses on selecting fraudulent companies with false records, fictitious profits and false assets as fraud samples. Only by comparing fraudulent companies with non-fraudulent companies can we observe the obviousness of fraudulent behaviors, and the non-fraudulent companies matching with each fraudulent company are selected as non-fraudulent samples. Then, the data of Choice Financial Terminal and Juchao Information Network are selected as the stability verification data set, and they are named Choice and cninf respectively.

In the analysis of the authenticity of financial statements, this paper balances the data set, then brings it into the random forest model, GBDT model and XGBoost model to construct the identification model, and uses the FS-f-VAEGAN-D2 (financial statements-f-VAEGAN-D2) in this paper to construct the fusion model, and tests the stability of each model to explore the optimal financial fraud identification model on this data set.

In the analysis of the predictive ability of the model, this paper uses the model proposed in this paper to conduct experiments on the data sets of SSE 50 and CSI 300, and compares them with the LSTM model, CNN-LSTM model, LSTM-Attention model, VMD-LSTM model, TCN model, BiLSTM model and TCCFR model.

4.2 Results

In this paper, the accuracy rate, recall rate, accuracy rate, F1 value and AUC value of each model on the CSMAR

test set are counted, and the statistical results are shown in Table 2.

Table 2: Comparative evaluation of models.

Model Category	Accuracy	Recall	Precision	F1-Score	AUC
RF	0.7446	0.7218	0.2893	0.4130	0.7996
GBDT	0.7422	0.7270	0.2879	0.4124	0.7953
XGBoost	0.7056	0.7041	0.2529	0.3722	0.7730
FS- f-VAEGAN-D2	0.7748	0.8034	0.2993	0.4361	0.8581

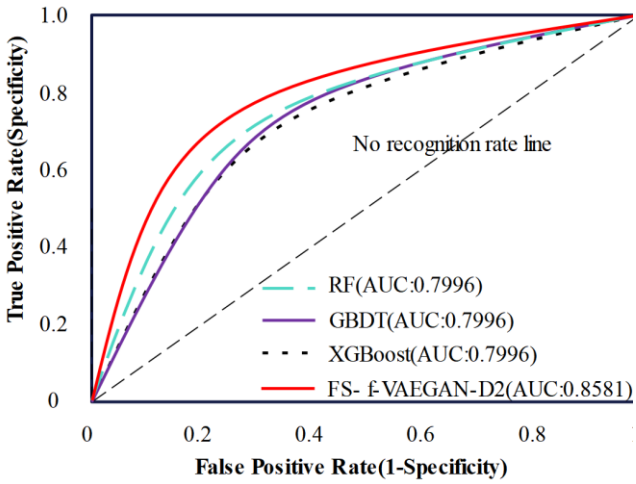


Figure 8: Comparison chart of ROC curves.

Financial forecasting models usually do not directly apply AUC curves for evaluation. The AUC curve (Area Under the Curve) is mainly used to evaluate the performance of binary classification models, especially when dealing with imbalanced datasets, which can provide stable and reasonable evaluation results. In binary classification problems, the AUC curve measures the model's ability to distinguish between positive and negative classes by showing the relationship between true case rate (TPR) and false positive case rate (FPR) at different thresholds. The ROC curve comparison chart is shown in Figure 8.

This article selects three publicly available anomaly detection table column datasets as experimental datasets, with dimensions ranging from less than 10 dimensions to hundreds of dimensions, and quantities ranging from hundreds to hundreds of thousands. The following is a specific introduction to the dataset and related experimental settings.

(1) CSMAR China Stock Market & Accounting Research Database is a research-oriented and precise database developed by Shenzhen Xishima Data Technology Co., Ltd. based on academic research needs and practical situations. There are a total of 385213 samples in CSMAR, most of which are abnormal samples, accounting for 78.36% of the total. CSMAR is a classic network intrusion detection dataset that includes four types of attacks in addition to normal data: denial of service, monitoring activities, remote unauthorized access, and local unauthorized access. Positive samples consist of data samples from four types of attacks, which are the superposition of multiple Gaussian distributions. Therefore, positive samples with complex distributions can also be used. This article will use single hot encoding to transform discrete features into continuous data that can be processed. The continuous features will be normalized and reduced to [1,1] -. The final CSMAR dataset consists of 119 features, 20000 training data, and 5000 testing data.

(2) Choice database is a financial database that provides comprehensive professional data services. It covers various financial products such as Shanghai and Shenzhen listed companies, funds, New Third Board, macro, industry, wealth management, bonds, futures, options, US stocks, Hong Kong stocks, etc., providing a variety of financial data including basic information, announcements, financial data, etc. This paper randomly samples and fills all data containing NAN and Inf to ensure that each data dimension is the same. Perform single hot encoding on discrete data columns, convert timestamp columns to relative time and normalize them, and remove useless information columns that are all single values. The final data dimension is 73 dimensions, with 40960 training data and a total of 20000 positive and negative test samples. Among them, each of the seven attack methods contains 1000 cases (if the total is less than 1000, they will all be classified as the test set).

(3) The cninf database provides corresponding programming interfaces or data processing tools for users to access and process data more conveniently This article will perform single hot encoding on discrete data columns, convert timestamp columns to relative time and normalize them, and remove useless information columns that are all single values. The final data dimension is 70 dimensions, with 35214 training data and 19525 positive and negative samples in the test data. Among them, each of the seven attack methods contains 1000 cases (if the total is less than 1000, they are all included in the test set).

In order to verify the stability of the model proposed in this paper, the original test data set is increased from one to three, and the stability of the model proposed in this paper is measured. The stability test results shown in Table 3 are obtained.

Table 3: Stability test results.

Test set	Model Category	Accuracy	Recall	Precision	F1-Score	AUC	P
CSMAR	RF	0.7446	0.7218	0.2893	0.4130	0.7996	<0.05
	GBDT	0.7422	0.7270	0.2879	0.4124	0.7953	<0.05
	XGBoost	0.7056	0.7041	0.2529	0.3722	0.7730	<0.05
	FS- f-VAEGAN-D2	0.7748	0.8034	0.2993	0.4361	0.8581	<0.05
Choice	RF	0.7419	0.7214	0.2874	0.4158	0.7968	<0.05
	GBDT	0.7390	0.7332	0.2851	0.4095	0.8007	<0.05
	XGBoost	0.7044	0.7111	0.2510	0.3725	0.7739	<0.05
	FS- f-VAEGAN-D2	0.7720	0.8090	0.2996	0.4322	0.8549	<0.05
cninf	RF	0.7423	0.7262	0.2874	0.4106	0.7939	<0.05
	GBDT	0.7458	0.7204	0.2893	0.4125	0.7876	<0.05
	XGBoost	0.6997	0.7043	0.2524	0.3741	0.7663	<0.05
	FS- f-VAEGAN-D2	0.7681	0.8026	0.2967	0.4341	0.8558	

The prediction error values of each model are shown in Table 4.

4.3 Analysis and discussion

Among the compared single integrated models, the evaluation indexes of XGBoost model are lower than those of random forest model and GBDT model, and the ability of identifying fraudulent companies and non-fraudulent companies is weak. In terms of the overall recognition accuracy of the model, the random forest model has the highest accuracy rate, reaching 74.46%, followed by the GBDT recognition rate at 74.22%, and the XGBoost model has the lowest accuracy rate, only 70.56%. In terms of recall rate, the GBDT model is slightly higher than the random forest model, with a recall rate of 72.70%%, but its accuracy rate, F1 value and AUC value are slightly lower than the random forest model. Therefore, in general, the overall performance of the random forest model is better than that of the GBDT model. To sum up, the random forest model has the best recognition performance, followed by the GBDT model, and finally the XGBoost model.

Table 4: Comparison of prediction effects of each model.

Dataset	Models	RMSE	MAE	MAPE/%	InferenceTime/ms
SSE 50	LSTM	35.442	25.622	0.907	3.477
	CNN-LSTM	40.418	29.563	1.048	2.154
	LSTM-Attention	36.163	26.748	0.947	2.323
	VMD-LSTM	39.109	30.404	1.072	11.816
	TCN	58.182	49.319	1.72	6.768
	BiLSTM	36.102	25.912	0.916	5.62
	TCCFR	34.804	24.711	0.875	4.272
	Logistic regression	38.251	29.351	1.021	9.321
	Neural networks	36.231	31.214	1.035	13.214
	FS-f-VAEGAN-D2	34.489	24.148	0.855	10.118
CSI 300	LSTM	47.387	37.968	0.982	5.904
	CNN-LSTM	51.008	40.354	1.044	3.483
	LSTM-Attention	40.727	31.972	0.831	3.843

	VMD-LSTM	49.918	41.728	1.089	18.564
	TCN	50.194	41.334	1.067	9.384
	BiLSTM	48.633	40.238	1.047	9.391
	TCCFR	40.635	31.339	0.814	7.295
	Logistic regression	39.214	30.474	1.024	9.761
	Neural networks	37.557	31.280	1.058	13.354
	FS-f-VAEGAN-D2	37.77	28.869	0.749	8.019

According to the results, in the fraud samples of the test set, the recognition rate of FS-f-VAEGAN-D2 model is 77.48%. It can be seen that FS-f-VAEGAN-D2 model has obviously improved the recognition effect of fraudulent companies, which is three percentage points higher than that of GBDT model, and has a good recognition ability for companies with fraudulent behaviors. Moreover, the area of ROC curve and coordinate axis also reached 0.817, which is higher than the area of the three single models established above, and has good overall performance.

Judging from the stability test results, the AUCs obtained by the RF model on the test sets CSMAR, Choice, and cninf are 0.7996, 0.7968, and 0.7939, respectively, and the AUCs obtained by the GBDT model on the test sets CSMAR, Choice, and cninf are 0.7953, 0.8007, 0.7876, the AUCs obtained by the XGBoost model on the test sets CSMAR, Choice, and cninf are 0.7730, 0.7739, and 0.7663, respectively, and the AUCs obtained by the FS-f-VAEGAN-D2 model on the test sets CSMAR, Choice, and cninf are 0.8581, 0.8549, and 0.8558, respectively, and the test results have little fluctuation and are basically stable. On the whole, the test method proposed in this paper has high stability.

The data in Table 4 shows that compared with existing neural network models and logistic regression models, our model has certain advantages in RMSE, MAE, MAPE, and Inference Time, which verifies that our model has certain advantages in financial early warning compared to existing models. It can be seen from Table 4 that the RMSE, MAE and MAPE values of the FS-f-VAEGAN-D2 model on the two data sets are the smallest. It shows that the prediction error of FS-f-VAEGAN-D2 model is small, and the prediction accuracy is higher than that of other models. However, the FS-f-VAEGAN-D2 model does not have advantages in terms of algorithm computational overhead. The InferenceTime of this model is 10.118 milliseconds on the SSE 50 dataset and 8.019 milliseconds on the CSI 300 dataset. The CNN-LSTM model has the shortest average time required to process a single test sample. The InferenceTime of the CNN-LSTM model is 2.154 ms on the SSE 50 dataset and 3.483 ms on the CSI 300 dataset. Although the FS-f-VAEGAN-D2 model takes more time to process a single sample on average than the CNN-LSTM model, InferenceTime is calculated in milliseconds. Compared with the millisecond time gap, the FS-f-VAEGAN-D2 model has a more significant improvement in prediction effect.

Based on the above analysis results, the FS-f-VAEGAN-D2 model proposed in this paper has good performance in the authenticity analysis of financial data and the prediction of financial data. Therefore, this paper increases data characteristics through financial indicators to improve the prediction ability. On the premise of considering historical data, prediction research is carried out on various financial indicators integrated into the company's financial statement information. Overall, the model proposed in this paper provides a reliable tool for financial data authenticity audit, and can use financial data forecasting to provide a reference for the formulation of subsequent plan policies.

5 Conclusion

Different from traditional financial analysis methods, the intelligent data analysis research method proposed in this paper mines the correlation of different dimensions of financial statement data, and presents the mining results by using the correlation visualization method to realize the risk assessment and trend prediction of enterprise financial status. Based on the basic WGAN model, the quality of generated samples is improved in the process of generating samples. Moreover, a cascade classifier is set in the classification stage to improve the classification accuracy. Through the experiments on each data set, it can be seen that the two regularizers and classifiers proposed by the model have the ability to improve the accuracy of zero-sample learning classification. According to the comprehensive experimental analysis results, it can be seen that the model proposed in this paper has good performance in the authenticity analysis and prediction of financial data. Generally speaking, the model proposed in this paper provides a reliable tool for the authenticity audit of financial data, and can provide a reference for the formulation of subsequent schemes and policies through financial data prediction.

However, the model proposed in this paper has too many regularization terms, which will affect the overall training process of the model and lead to model convergence failure in some extreme cases. Therefore, in the follow-up research, it is necessary to focus on finding a better regularization term to replace the proposed regularization term and reduce the complexity of the

model.

The current model in this article has certain limitations in convergence under extreme conditions, and actionable work will be taken in the future to address these issues. Therefore, further validation strategies need to be proposed in different financial environments to enhance the robustness of the model.

References

[1] Wasserbacher, H., & Spindler, M. (2022). Machine learning for financial forecasting, planning and analysis: recent developments and pitfalls. Digital Finance, 4(1), 63-88. https://doi.org/10.1007/s42521-021-00046-2

[2] Tang, Y., Song, Z., Zhu, Y., Yuan, H., Hou, M., Ji, J., ... & Li, J. (2022). A survey on machine learning models for financial time series forecasting. Neurocomputing, 512, 363-380. https://doi.org/10.1016/j.neucom.2022.09.003

[3] Masini, R. P., Medeiros, M. C., & Mendes, E. F. (2023). Machine learning advances for time series forecasting. Journal of Economic Surveys, 37(1), 76-111. https://doi.org/10.1111/joes.12429

[4] Wang, J., Hong, S., Dong, Y., Li, Z., & Hu, J. (2024). Predicting stock market trends using lstm networks: overcoming RNN limitations for improved financial forecasting. Journal of Computer Science and Software Applications, 4(3), 1-7. https://doi.org/10.5281/zenodo.12200708

[5] Foroni, C., Marcellino, M., & Stevanovic, D. (2022). Forecasting the Covid-19 recession and recovery: Lessons from the financial crisis. International Journal of Forecasting, 38(2), 596-612. https://doi.org/10.1016/j.ijforecast.2020.12.005

[6] Barbaglia, L., Consoli, S., & Manzan, S. (2023). Forecasting with economic news. Journal of Business & Economic Statistics, 41(3), 708-719. https://doi.org/10.1080/07350015.2022.2060988

[7] Bhat, A., Kulkarni, N., Husain, S., Yadavalli, A., Kaur, J. N., Shukla, A., ... & Seshadri, V. (2024). Speaking in terms of money: financial knowledge acquisition via speech data generation. ACM Journal on Computing and Sustainable Societies, 2(3), 1-35. https://doi.org/10.1145/3663775

[8] Ren, S. (2022). Optimization of enterprise financial management and decision-making systems based on big data. Journal of Mathematics, 2022(1), 1708506. https://doi.org/10.1155/2022/1708506

[9] Qi, Q. (2022). Analysis and forecast on the price change of shanghai stock index. Journal of Economics, Business and Management, 10(1), 72-78. https://doi.org/10.18178/joebm.2022.10.1.676

[10] Petrozziello, A., Troiano, L., Serra, A., Jordanov, I., Storti, G., Tagliaferri, R., & La Rocca, M. (2022). Deep learning for volatility forecasting in asset management. Soft Computing, 26(17), 8553-8574. https://doi.org/10.1007/s00500-022-07161-1

[11] Li, Y., & Pan, Y. (2022). A novel ensemble deep learning model for stock prediction based on stock prices and news. International Journal of Data Science and Analytics, 13(2), 139-149. https://doi.org/10.1007/s41060-021-00279-9

[12] Souto, H. G., & Moradi, A. (2023). Forecasting realized volatility through financial turbulence and neural networks. Economics and Business Review, 9(2), 133-159. https://doi.org/10.18559/ebr.2023.2.737

[13] Zhan, X., Ling, Z., Xu, Z., Guo, L., & Zhuang, S. (2024). Driving efficiency and risk management in finance through AI and RPA. Unique Endeavor in Business & Social Sciences, 3(1), 189-197. https://doi.org/10.69987/JACS.2024.40501

[14] Wei, L., Deng, Y., Huang, J., Han, C., & Jing, Z. (2022). Identification and analysis of financial technology risk factors based on textual risk disclosures. Journal of Theoretical and Applied Electronic Commerce Research, 17(2), 590-612. https://doi.org/10.3390/jtaer17020031

[15] Lei, Y., Qiaoming, H., & Tong, Z. (2023). Research on supply chain financial risk prevention based on machine learning. Computational Intelligence and Neuroscience, 2023(1), 6531154. https://doi.org/10.1155/2023/6531154

[16] Levytska, S., Pershko, L., Akimova, L., Akimov, O., Havrilenko, K., & Kucherovskii, O. (2022). A risk-oriented approach in the system of internal auditing of the subjects of financial monitoring. International Journal of Applied Economics, Finance and Accounting, 14(2), 194-206. https://doi.org/10.33094/ijaefa.v14i2.715

[17] Wang, H., & Budsaratragoon, P. (2023). Exploration of an" Internet+" grounded approach for establishing a model for evaluating financial management risks in enterprises. International Journal for Applied Information Management, 3(3), 109-117. https://doi.org/10.47738/ijaim.v3i3.58

[18] Rodríguez, C. E. L., De la Hoz Solano, V. M., & Rozo, C. A. B. (2022). Financial risks in the operation of special service transportation in the hotel sector in Bogota, Colombia. Revista Investigación, Desarrollo Educación, Servicio, Trabajo, 2(1), 63-88. https://doi.org/10.31876/idest.v2i1.32

[19] Vuletić, M., Prenzel, F., & Cucuringu, M. (2024). Fin-gan: Forecasting and classifying financial time series via generative adversarial networks. Quantitative Finance, 24(2), 175-199. https://doi.org/10.1080/14697688.2023.2299466

[20] Wu, W., Han, M., Hu, Y., Ma, J., & Zhang, X. (2024). Application of SOM-GAN based deep learning technology in the security protection of rural bank depositors' funds information. Highlights in Science, Engineering and Technology, 94, 639-646. https://doi.org/10.54097/jvmf7f78

[21] Bai, X., Zhuang, S., Xie, H., & Guo, L. (2024).

Leveraging generative artificial intelligence for financial market trading data management and prediction. Journal of Artificial Intelligence and Information, 1, 32-41. https://doi.org/10.20944/preprints202407.0084.v1

[22] Zhang, Y., Jiang, Z., Peng, C., Zhu, X., & Wang, G. (2024). Management analysis method of multivariate time series anomaly detection in financial risk assessment. Journal of Organizational and End User Computing, 36(1), 1-19. https://doi.org/10.4018/JOEUC.342094

Hybrid CatBoost and SVR Model for Earthquake Prediction Using the LANL Earthquake Dataset

Arush Kaushal*[1], Ashok Kumar Gupta[2], Vivek Kumar Sehgal[1]
[1] Department of Computer Science and Information Technology, Jaypee University of Information Technology, Solan 171234, India
[2] Department of Civil Engineering, Jaypee University of Information Technology, Solan 171234, India
E-mail: arushkaushal0115@gmail.com, akgjuit@gmail.com, vivekseh@ieee.org
*Corresponding Author

Keywords: seismology, earthquake prediction, time to failure, hybrid model

Received:

Earthquakes have the potential to cause catastrophic structural and economic damage. This research explores the application of machine learning for earthquake prediction using LANL (Los Alamos National Laboratory) dataset. The data, obtained from a laboratory stick-slip friction experiment, simulate real earthquakes through digitized acoustic signals recorded against the time to failure of a granular layer. We introduced a hybrid model combining CatBoost and Support Vector Regression (SVR) to predict the time of the next earthquake, evaluating its performance against individual CatBoost and SVR models. The hybrid model demonstrated superior accuracy with a Mean Absolute Error (MAE) of 0.0825, outperforming the individual models. We implemented feature engineering to optimize the predictive capability of the models. Additionally, we compared our hybrid model's performance with previous studies to validate its efficacy. Our findings underscore the potential of machine learning, particularly hybrid models, in enhancing earthquake prediction accuracy. This study highlights the robustness and effectiveness of the hybrid CatBoost-SVR model, paving the way for advanced AI algorithms in seismology and contributing to improved disaster preparedness and mitigation strategies.

Povzetek: A hybrid CatBoost-SVR model improves earthquake prediction using the LANL dataset, achieving superior accuracy (MAE: 0.0825). This approach enhances machine learning applications in seismology, contributing to disaster preparedness and mitigation strategies.

1 Introduction

Earthquakes stand as one of nature's most devastating phenomena, posing formidable challenges for prediction despite the extensive efforts of the seismology community. Unlike other natural disasters such as floods, tornadoes, and hurricanes, which can often be forecasted in terms of timing, location, and potential impact, earthquake prediction remains notably elusive. Currently, seismographs serve as the primary method for detecting imminent earthquakes, yet their warnings typically offer only seconds of lead time, insufficient for effective preventive action against substantial structural damage. The complexity and nonlinear characteristics of seismic data further compound the difficulty in earthquake prediction, presenting a persistent challenge in geophysics. Recent strides in machine learning present promising avenues for improving prediction accuracy in earthquake forecasting. This study delves into a hybrid model that merges CatBoost and Support Vector Regression (SVR) to enhance earthquake prediction performance, leveraging insights gained from analyzing the LANL earthquake dataset. Additionally, alternative approaches to earthquake prediction involve monitoring changes in land elevation, groundwater levels, animal behavior, and precursor seismic activity. A notable instance of effective earthquake prediction transpired during the Haicheng, China earthquake of 1975, where an evacuation advisory was disseminated a day prior to the occurrence of a magnitude 7.3 seismic event. In the month’s antecedent to the earthquake, alterations in land surface elevation and groundwater levels, numerous instances of anomalous animal behavior, and the occurrence of several foreshocks collectively served as precursory indicators, initially prompting a precautionary advisory. Subsequently, a surge in foreshock activity prompted the escalation of the advisory to an evacuation warning. Nevertheless, it is imperative to note that the majority of earthquakes do not manifest such conspicuous precursory signs. Despite the success witnessed in 1975, the 1976 Tangshan earthquake, registering a magnitude of 7.6, occurred without any forewarning, resulting in an estimated 250,000 casualties [1]. Amidst the rapid evolution of statistical and deep learning methodologies, novel paradigms in earthquake prediction have emerged [2] [3]. These strategies hinge upon extensive datasets, accentuating the imperative of curating, amassing, and simulating earthquake data, a realm that has recently garnered notable scrutiny [4]. Through the fusion of meticulously curated data and cutting-edge statistical and

deep learning methodologies, the endeavor to forecast earthquake timing based on realistically attainable data could potentially be surmounted, aligning with the prevailing trajectory across diverse machine learning applications. In both [2] and [3], machine learning and deep learning frameworks are harnessed to prognosticate the timing of subsequent earthquakes. These frameworks leverage physically amassed and labeled earthquake parameters, such as relative strength index, momentum, and moving force averages. Classic machine learning (ML) [4] algorithms conventionally compute seismic metrics like Gutenberg-Richter b-values, time intervals, earthquake energy, and mean magnitude. In contrast, contemporary deep learning (DL) models [5] exhibit proficiency in assimilating multifaceted features. Both ML and DL models are driven by data and demonstrate efficacy in moderate-magnitude earthquake scenarios; however, they encounter challenges with high-magnitude events due to the scarcity of requisite data. Data-driven models necessitate voluminous datasets to yield precise predictions. Certain DL methodologies endeavor to anticipate significant earthquakes by exclusive training on such instances, yet these methods necessitate further refinement [6]. A prevalent trait among these methodologies is their analysis of protracted temporal sequences of seismic data, which poses a formidable hurdle for DL techniques. Accurate earthquake prediction holds the potential to avert fatalities and mitigate catastrophic repercussions, thus positioning the anticipation of earthquake timing and magnitude as a cornerstone objective within the domain of geoscience [7]. Despite protracted time-series observations and field studies, the precise anticipation of earthquake scale or timing persists as an enduring challenge [8]. Moreover, the unpredictability of devastating subduction earthquakes, with magnitudes of 9.0 or higher, adds a concerning dimension to this endeavor [9]. Traditional methods of earthquake prediction, such as using seismographs, often provide only seconds of warning before an earthquake occurs, which is insufficient time to take preventative measures. Other approaches involve monitoring changes in land elevation, groundwater levels, animal behavior, and foreshocks. However, these methods do not always provide clear or reliable precursors to impending earthquakes. Monitoring with non-destructive testing (NDT) acoustic emissions (AE) involves the continuous recording of acoustic data as the material undergoes stress. The recording process typically persists until the material reaches failure. In controlled laboratory environments, stress-induced failure can be hastened by artificially subjecting the material to stress [10][11]. Upon failure, discrete acoustic emissions (AEs) are discerned within the recorded data. These discrete AEs denote short-duration elastic waves generated due to the initiation of minute internal cracks and slip occurrences along grain contacts, thereby furnishing valuable insights into the material's response under stress. Subsequently, AEs can be categorized based on the damage mechanism through the utilization of unsupervised clustering algorithms. In certain instances, the precise labels for each cluster are determined through methodologies such as transmitted light analysis [12] or scanning electron microscopy [13]. Finally, scrutinizing AE production across the failure cycle facilitates the identification of temporal patterns and enables deductions regarding the material's remaining useful life (RUL). Some research endeavors have expanded upon this analysis by integrating machine learning methodologies to forecast RUL, yielding varying degrees of efficacy [14][15].

Recent advancements in machine learning (ML) algorithms and computational hardware have catalyzed novel insights and methodologies within the seismological community [16]. ML applications now extend to fundamental signal processing tasks, encompassing earthquake event detection [17], phase picking [18], association [19], and hypocenter determination [20], as comprehensively documented by [21]. Concurrently, the utilization of data-driven ML approaches has broadened to encompass the prediction of TTF in laboratory experiments, leveraging Acoustic Emission (AE) data and its associated measurements [22]. A study on earthquake forecasting emphasizes the importance of long-term predictions regarding the timing, intensity, and location of future seismic events. By leveraging expert systems and extensive data analysis, more accurate forecasting models can be developed for specific regions, such as Los Angeles, improving preparedness and risk management [23]. Advanced machine learning techniques, such as attention-based Bi-Directional Long Short-Term Memory (LSTM) networks, have been highlighted as powerful tools for enhancing the precision of earthquake predictions, which are critical for disaster response in earthquake-prone areas [24]. Additionally, extreme value theory has been applied to assess the maximum possible earthquake magnitudes in high-risk areas, underscoring the value of ground-based observations and statistical methods in refining forecasting models [25].

Machine learning techniques have also been used to cluster earthquakes based on historical intervals, offering insights into recurring seismic behaviors and improving the predictive power of long-term forecasts [26]. For regions with complex fault zones, statistical models like the SARIMA model can help forecast earthquakes by analyzing past seismic events, contributing to more reliable predictions and risk management strategies [27]. Other research has focused on analyzing geoelectric field signals before earthquakes using advanced techniques, which can provide early warning signs and valuable data to improve forecasting accuracy [28].

Another significant area of study is the monitoring of slow seismic activity, which may indicate the potential for a major earthquake. By identifying these patterns, researchers can enhance the effectiveness of forecasting models [29]. Additionally, understanding the relationship between ground motion attenuation and regional geophysical data is crucial for developing robust forecasting models that predict the impact of seismic events [30]. Lastly, the study of seismic stress levels and their relationship with earthquake magnitude helps improve predictions of high-magnitude earthquakes, providing deeper insights into seismic behavior and further refining forecasting methods [31]. Collectively,

these approaches are advancing earthquake forecasting and risk management, enhancing preparedness in earthquake-prone regions.

When working with the LANL earthquake dataset, it is essential to recognize several limitations and potential biases that influence the generalizability and reliability of the findings. The dataset is geographically biased, focusing on specific regions, which limits the applicability of conclusions to areas with different seismic characteristics. If the data predominantly covers certain tectonic plate boundaries or fault lines, it does not fully represent the behavior of earthquakes in less seismically active regions. Additionally, the dataset has temporal gaps, with uneven data distribution over time, which affects trends analysis and the ability to draw consistent conclusions across different periods. The dataset also has biases in the types of seismic events included, such as overrepresentation of certain magnitudes or depths, which skews model development. If smaller or larger earthquakes are underrepresented, the results do not accurately reflect the full range of seismic activity. Furthermore, the quality of the data is important, as seismic recordings are affected by noise from environmental factors, sensor inaccuracies, or technological limitations. If the dataset includes noisy or incomplete data, it compromises the ability to detect meaningful patterns or leads to incorrect conclusions. Incomplete or missing data points, especially if they are not randomly distributed, further introduce biases. There are also issues with manual labeling and classification errors, where misclassification of events distorts the analysis, particularly if smaller seismic events are confused with more significant ones. Finally, the sampling frequency of the dataset impacts its usefulness, as insufficient resolution results in the loss of critical information, such as early warning signs of large earthquakes or aftershocks. Acknowledging these limitations and biases is crucial for a more realistic and transparent understanding of the dataset's applicability to earthquake prediction and modeling.

In the context of predicting Time to Failure (TTF) within controlled laboratory environments, researchers typically employ Machine Learning (ML) frameworks that rely on three distinct feature categories. These categories encompass a) AE-Driven Features, which are directly derived from continuous Acoustic Emission (AE) signals, capturing nuanced details about the material's structural response and behavior; b) Geodetic-Driven Features, extracted from geodetic measurements, offering insights into the material's deformation characteristics and spatial dynamics, thus shedding light on its mechanical integrity; and c) Catalog-Driven Features, sourced from earthquake or seismicity catalogs, providing historical data on seismic events and their associated attributes. These feature categories collectively enable a comprehensive approach to TTF prediction, integrating diverse data sources to enhance predictive accuracy and reliability within laboratory settings. Acoustic emissions (AE) denote transient elastic waves arising from the formation of minute internal cracks and slip events along grain contacts within stress-stricken materials. AE monitoring offers invaluable insights into material structural integrity and response mechanisms under stress, thus laying the foundation for TTF prediction in laboratory setups. The amalgamation of AE data with machine learning methodologies presents a promising avenue for enhancing the precision and efficacy of TTF prognostications, thereby fostering advancements in comprehending material behavior under stress and augmenting predictive capacities within the realm of seismology. Despite the limitations and biases present in the LANL earthquake dataset, our "Hybrid CatBoost and SVR Model" helps provide better results by effectively addressing these challenges. The CatBoost algorithm, known for its robustness in handling categorical features and its ability to deal with noisy and incomplete data, enhances the model's ability to identify important patterns in seismic events. By reducing overfitting and improving generalization, CatBoost ensures that the model remains accurate even in the presence of biases like geographical or temporal imbalances. On the other hand, the Support Vector Regression (SVR) component helps capture complex relationships in the data, especially for modeling non-linearities that might arise due to varying earthquake magnitudes and depths. Together, the hybrid model leverages the strengths of both algorithms, enabling it to mitigate the impact of incomplete or noisy data, and ultimately providing more reliable predictions. Additionally, the combination of CatBoost's feature engineering capabilities and SVR's precision ensures that even with a limited dataset, the model can deliver meaningful insights, improving the overall accuracy and robustness of earthquake predictions.

We propose a novel hybrid approach that combines CatBoost, a gradient boosting algorithm, with Support Vector Regression (SVR). This hybrid model leverages the strengths of both methods to improve the accuracy of predicting the time-to-failure of earthquakes using the LANL earthquake dataset. Integrating CatBoost with Support Vector Regression (SVR) can yield superior results due to the complementary strengths of the two algorithms. CatBoost, a gradient boosting algorithm, is adept at handling categorical features and automatically managing missing data. It excels in capturing complex relationships within the dataset, producing robust predictions. On the other hand, SVR, a kernel-based regression algorithm, is proficient in modeling nonlinear relationships and high-dimensional spaces. By combining the predictions from CatBoost with the features in SVR, the integrated model can leverage the advantages of both algorithms. CatBoost provides an initial understanding of the data's complex patterns, while SVR further refines predictions based on its ability to capture intricate relationships.

Our paper builds significantly on the work presented in [32], where researchers from the Los Alamos National Laboratory (LANL) developed a dataset of acoustic data from laboratory-simulated earthquakes. This dataset was utilized to train a support vector regression (SVR) machine learning model for predicting time-to-failure, defined as the time until a major earthquake event. The model used statistical features such as moving average,

kurtosis, and variance. In this study, we aim to enhance their results by Catboost techniques. Some major contributions of research includes:

This research bridges the gap between machine learning and seismology, demonstrating how advanced data-driven approaches can be applied to traditional scientific problems. The study also incorporates the consideration of slow slip events (SSEs) and their relationship with regular earthquakes, adding to the understanding of seismic processes.

2 Dataset description

In 2017, researchers at Los Alamos National Laboratory (LANL) achieved a significant breakthrough in the prediction of Slow Slip Earthquakes (SSE) within laboratory conditions that mimic natural settings. Through meticulous experimentation, the team developed a method wherein a computer system was trained to detect and analyze quasi-periodic seismic and acoustic signals emitted during fault movements. By processing extensive datasets, they identified a distinct sound pattern, previously dismissed as noise, which serves as an indicator of an impending earthquake. Utilizing a time window of 1.8 seconds of data, the team attained an impressive 89% coefficient of determination in forecasting the time remaining before a laboratory earthquake event, employing Random Forest Regression and quasi-periodic data. In the laboratory environment, seismic sounds produced by the interaction of steel blocks with rocky material, simulating real earthquake activity, were recorded by an accelerometer. This groundbreaking discovery represents the first successful prediction of laboratory earthquake occurrences. While acknowledging the differences in shear stress between laboratory experiments and natural earthquakes, the LANL team is actively engaged in validating their findings in real-world scenarios [33][34]. Moreover, this innovative approach holds potential beyond seismology, with possible applications in material failure research across diverse industries like aerospace and energy. These findings underscore the notion that fault failure follows a discernible pattern rather than occurring randomly.

3 Data exploration

The LANL earthquake dataset serves as a comprehensive repository of acoustic emission signals captured during laboratory-simulated earthquakes. Each entry within this dataset encapsulates the acoustic data recorded at distinct time intervals, providing a detailed snapshot of seismic activity. Crucially, each sample is paired with a target value denoting the time until the occurrence of the subsequent laboratory earthquake. This temporal information enables researchers to study the dynamics of earthquake events and explore predictive modeling approaches [35]. The acoustic data itself is composed of discrete segments, each spanning a duration of 0.0375 seconds, comprising seismic signals recorded at

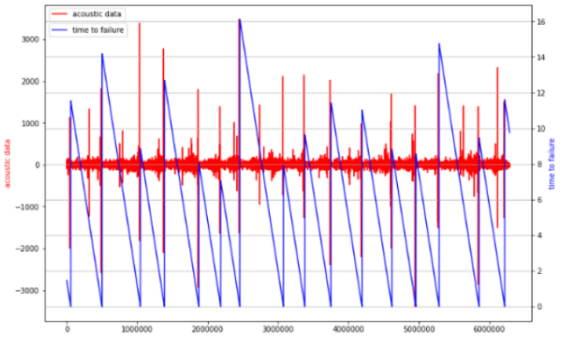


Figure 1: Acoustic data and time to failure analysis: subset representing 1% of total dataset.

a frequency of 4MHz. This results in a substantial dataset containing a total of 150,000 data points. Each segment of acoustic data is meticulously annotated with a corresponding "time to failure" value as shown in Table 1, representing the duration until the laboratory fault undergoes failure, as determined through stress measurements. The acoustic signal consistently exhibits significant fluctuations immediately preceding each failure event Figure 1. Additionally, it is noteworthy that failures can be visually anticipated by observing instances where substantial fluctuations in the signal are succeeded by smaller ones.

Upon closer examination of a zoomed-in time plot Figure 2, it becomes apparent that the prominent acoustic signal oscillation occurring at the 1.572-second mark precedes the occurrence of the failure event, albeit not precisely coinciding with it. Before this major oscillation, there are noticeable sequences of intense signal fluctuations, suggesting a buildup of activity leading to the larger event. Subsequently, after the significant oscillation, there are also smaller oscillations observed, indicating a potential aftermath or continuation of the event's effects [36][37]. In this time plot, it becomes evident that the significant oscillation preceding the failure does not occur immediately before the event.

Table 1: Dataset: Seismic Activity (v) and Time to Failure (s)

Sesmic activity ($\boldsymbol{v}$)	Time to failure (s)
12	1.4690999832
6	1.4690999821
8	1.469099981
5	1.4690999799
8	1.469099988
8	1.469099977
9	1.4690999766
7	1.4690999755
-5	1.4690999744
...	...

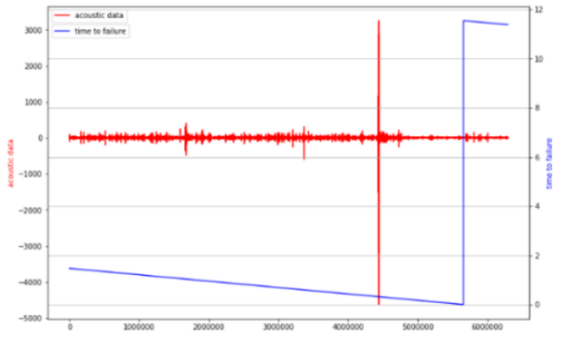


Figure 2: Zoomed-in-time-plot.

Instead, there are sequences of intense oscillations that precede the large oscillation, as well as smaller peak oscillations that follow it. Subsequently, after a series of minor oscillations, the failure takes place. Initially structured as a Pandas Dataframe, the dataset underwent a process of segmentation, dividing it into 150,000 individual samples. Each sample is coupled with its corresponding time to failure, facilitating the training and validation of predictive models. Moreover, the dataset includes an additional 2626 preconstructed acoustic segments earmarked specifically for model testing purposes. This meticulous organization of the dataset enables researchers to conduct robust evaluations of model performance and effectiveness in earthquake prediction tasks. Seismic signals are captured through a piezoceramic sensor that generates a voltage in response to deformation caused by incoming seismic waves. This voltage, referred to as the acoustic signal, serves as the primary input for our analysis. The acoustic signal represents the recorded voltage, expressed as integers.
The Acoustic Signal essentially signifies the voltage generated by the deformation induced by seismic waves. These signals are integer values ranging from -5515 to 5444, with an average of 4.52. Examining the distribution of these acoustic signals reveals a distinct peak, indicating a concentration of values around the mean. However, the distribution also exhibits outliers in both directions, suggesting sporadic occurrences of exceptionally high or low values. This observation is illustrated in Figure 3, where the distribution's shape and the presence of outliers can be visualized. The range of the acoustic signals,

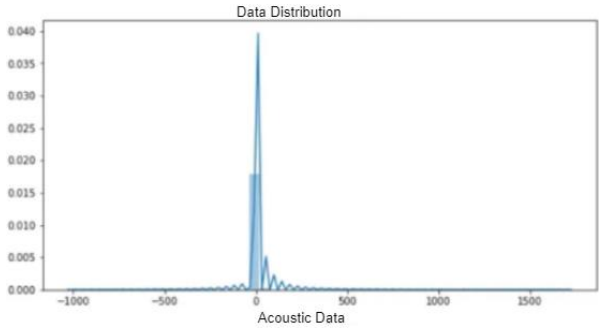


Figure 3: The distribution of acoustic signals analyzed individually.

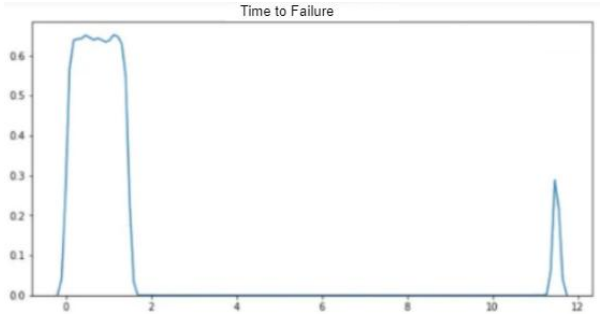


Figure 4: The distribution of time to failure analyzed individually.

spanning from -5515 to 5444, reflects the entirety of recorded voltage variations, from the most negative to the most positive values. This comprehensive range offers insights into the full spectrum of voltage fluctuations experienced during seismic activity. Negative values might signify voltage decreases due to compression or damping effects, while positive values could indicate voltage increases resulting from tension or amplification. The wide span of this range underscores the substantial variability in recorded voltage, influenced by factors like seismic wave intensity, distance, environmental conditions, and sensor sensitivity [38]. Despite the range's breadth, a very high peak in the distribution suggests a clustering of values around a central tendency, indicative of predominant signal strength or intensity. However, the presence of outliers in both directions highlights occasional deviations from this central tendency, likely stemming from anomalies in seismic activity or sensor behavior. These outliers necessitate careful consideration during data analysis to ensure accurate interpretation and modeling of the seismic signals.

The time to failure represents the duration, in seconds, remaining until an imminent stick-slip failure event occurs. This metric serves as a crucial indicator of the proximity of failure, allowing for proactive measures to be taken. The minimum value of Time to Failure is extremely close to zero, at 9.55039650e-05 seconds, indicating instances where failure occurred imminently after observation. Conversely, the maximum Time to Failure extends to 16 seconds, representing cases where failure was predicted further in advance. The distribution of Time to Failure exhibits a right-skewed pattern, as illustrated in Figure 4. This skewness indicates that the majority of observations are clustered towards the lower end of the

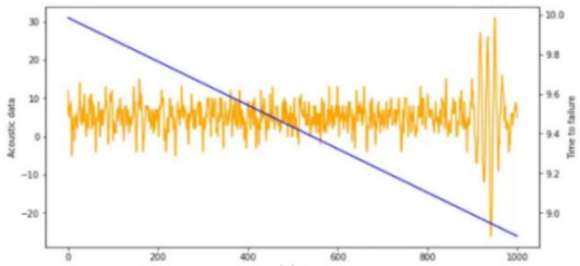


Figure 5: Time series relationship between first 1000 rows.

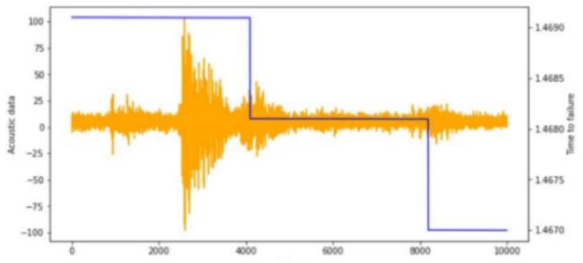


Figure 6: Time series relationship between first 10,000 rows.

time scale, with fewer instances of longer Time to Failure values. This distribution pattern provides valuable insights into the temporal dynamics of stick-slip failure events, highlighting the variability in the timeframes leading up to failure occurrences. The explanation details a time-series plot analyzing the first 1000 rows of data, with the orange lines depicting seismic activity (acoustic feature) and the blue line representing time to failure, indicating the duration until the next earthquake. Notably, the plot reveals a linear trend in the time to failure, suggesting a consistent change over time, implying a potential predictive relationship with the acoustic feature.

Figure 5 centres on analyzing time-based data, underscoring the importance of examining both the distribution of acoustic signals and the target feature over time. Two functions are provided to facilitate visualization of these features. First function generates a plot showcasing the acoustic data and time to failure for a specified range of indices, while the other function allows for comparison across two distinct index ranges.

In the example provided, the first function is employed to plot the first thousand rows of the dataset, with orange lines representing the acoustic feature and a blue line depicting the target feature. The resulting plot illustrates a linear relationship in the target feature, prompting further exploration to gain a comprehensive understanding of the dataset's behavior across a broader range of rows.

After examining the initial 1000 rows, further analysis is conducted on larger subsets of the data, including the first 10,000 rows shown in Figure 6 and the entire dataset comprising 600,000 rows shown in Figure 7. These analyses reveal consistent trends, with the time to failure decreasing sharply to nearly zero seconds when an earthquake event occurs, indicating a rapid onset of seismic activity. The observations underscore the predictive potential of the acoustic data in forecasting

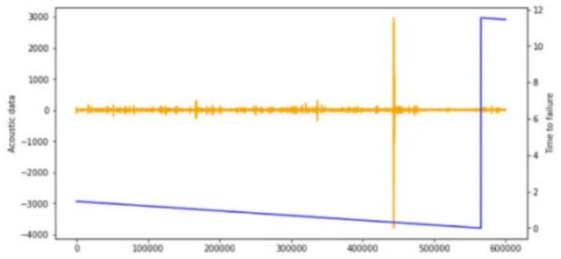


Figure 7: Time series relationship between first 600k rows.

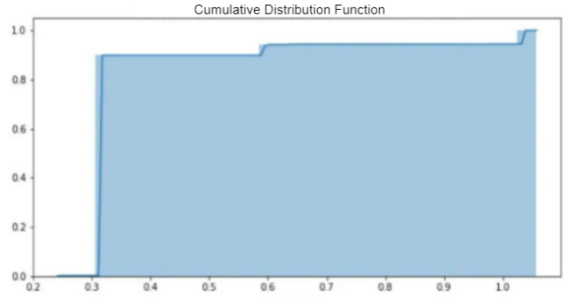


Figure 8: Cumulative distribution of the time to failure with high signal.

earthquake occurrences and highlight the significance of ongoing analysis to refine predictive models and enhance accuracy. After generating the time-series plots, we analyze them to extract meaningful insights about the behavior of the data over time. This analysis involves identifying recurring patterns, detecting abrupt changes or anomalies, and assessing the overall trend in the data series. By interpreting the time-series plots, we can gain a deeper understanding of the underlying dynamics driving seismic activity and the predictive relationship between the acoustic data and time to failure. In summary, time-

series analysis plays a crucial role in uncovering temporal patterns and relationships within the data, providing valuable insights that inform subsequent modeling and prediction efforts in the context of earthquake forecasting. In our analysis, we examined a dataset containing a massive 629 million rows, although our focus was on a subset of 600,000 rows. We were particularly interested in understanding the timing of events, noting that the time-to-failure spanned from nearly zero to 12 seconds.

To delve deeper into this aspect, we decided to investigate the Cumulative Distribution Function (CDF) of the target feature, which helped us understand how frequently events occurred within the 0 to 12-second range. After setting the display precision and loading the dataset, we visualized the distribution of acoustic data. Upon examining the CDF plot shown in Figure 8 of the target feature, we discovered that approximately 85% of the events occurred within a mere 0.3 seconds, indicating a rapid onset of events. This observation shed light on the timing patterns within the dataset and emphasized the importance of events occurring within close proximity to zero seconds.

3.1 Feature engineering

Data preprocessing is an essential preliminary step in harnessing the LANL earthquake dataset for model training and assessment. This section delineates a series of preprocessing procedures orchestrated to refine the dataset, ensuring its cleanliness, informativeness, and readiness for subsequent analyses. The journey begins with the ingestion of the LANL earthquake dataset, an amalgamation of acoustic signal data accrued during laboratory-simulated earthquakes. Within this dataset lie acoustic emission signals, captured at varied time intervals, accompanied by corresponding time-to-failure

values delineating the duration until the advent of subsequent seismic events. Subsequently, meticulous data cleaning protocols are executed to rectify any aberrations present within the dataset. Through adept imputation techniques, missing values are diligently addressed, ensuring comprehensive data coverage. Concurrently, outliers, with their potential to skew model training outcomes, are meticulously identified and rectified through judicious methods. Following data cleansing, the dataset undergoes a transformative phase through the application of feature engineering techniques.

The data cleaning and preparation process for the LANL earthquake dataset involved several key steps to ensure the quality and consistency of the input data for the hybrid model. First, missing or incomplete data points were identified and addressed through appropriate imputation techniques or, in some cases, by removing records with excessive missing values to avoid introducing bias. Next, outliers were detected and handled to prevent them from disproportionately influencing the model's predictions. This step was particularly important as seismic data can sometimes contain unusual readings due to sensor malfunctions or other anomalies. Data normalization and scaling were applied to ensure that features with different units and ranges did not skew the performance of the model, particularly for algorithms like Support Vector Regression (SVR), which are sensitive to the scale of the input data. Additionally, categorical variables, such as event types or geographic locations, were encoded using techniques such as one-hot encoding or label encoding to make them compatible with the CatBoost algorithm, which is capable of handling categorical data efficiently. Temporal features, such as the date and time of seismic events, were also processed to extract meaningful patterns, such as trends or seasonality, that could contribute to better model performance. Feature engineering was performed to create new variables that could enhance the model's ability to identify key seismic patterns, such as calculating the time between successive events or aggregating data at different time intervals. Through this comprehensive data cleaning and preparation process, the dataset was transformed into a structured and reliable format, enabling the hybrid model to learn effectively and provide accurate predictions. Feature engineering is a critical step in the model development process, as it involves transforming raw data into meaningful features that can enhance the predictive power of machine learning models. In this study, feature engineering was focused on extracting key characteristics from Acoustic Emission (AE) data, which is considered a rich source of information for predicting Time to Failure (TTF). The goal of feature engineering was to identify and create features that can effectively capture the underlying patterns and dynamics of the AE signals, which are indicative of the system's failure behavior. The feature engineering process began with the assumption that the distribution of AE data holds valuable information that can be leveraged to predict failure. This assumption is based on both empirical observations and established findings in the literature, which suggest that variations in AE data, particularly in the form of spikes, can precede failure events. By focusing on these variations, we aimed to identify statistical features that could serve as reliable indicators of failure time. A key insight from the data was that stick-slip failure events, often associated with mechanical systems, are typically preceded by a series of AE signal spikes. These spikes, which are indicative of micro-failures, provide crucial information that can help predict when a system is approaching failure. We hypothesized that the frequency and intensity of these AE spikes correlate with the remaining useful life of the system, and therefore, the statistical characteristics of the AE signal could serve as valuable features for modeling. Building on this foundation, we derived a set of 18 statistical features from each 150,000-point segment of the AE data. These features included basic statistical metrics such as mean, standard deviation, skewness, and kurtosis, which have been shown to reflect important characteristics of the AE signal. Additionally, we calculated features like the ratio of standard deviation to mean, as well as distributional features represented by various percentiles (e.g., 1st, 5th, 25th, 50th, etc.). These features were selected because they provide a more comprehensive representation of the AE signal's behavior over time. Not all derived features were ultimately used in the model. For example, while maximum and minimum values were initially considered, they were excluded from the final feature set due to their sensitivity to extreme events, which mainly serve as markers of significant disruptions in the AE signal rather than predictors of failure. After the features were extracted, a database was created, which contained a large set of statistical features corresponding to each segment of AE data. This database covered a wide range of TTF values, allowing us to explore how each feature correlated with the time to failure. Initial analysis showed that certain features, such as the count of mode appearances, exhibited a strong correlation with TTF. However, special care was taken to exclude data recorded immediately after major failure events, as these post-event values closely resembled early-stage data and could introduce inaccuracies into the predictive model. Herein, statistical attributes such as mean, standard deviation, skewness, and kurtosis are meticulously computed, affording insights into the distributional characteristics of the acoustic signals. The derivation of rolling window statistics facilitates the capture of temporal nuances and trends embedded within the data. Furthermore, to foster uniformity and comparability across diverse features, the dataset is subjected to normalization or standardization.

Table 2: Comprehensive global overview of the dataset statistics

	acoustic - data	**time-to-failure**
count	6.29E+08	6.291E+08
mean	4.52E+00	5.68E+00
min	-5.52E+03	9.55E-01
max	5.44E+03	1.61E+01
std	1.07E+01	3.67E+00

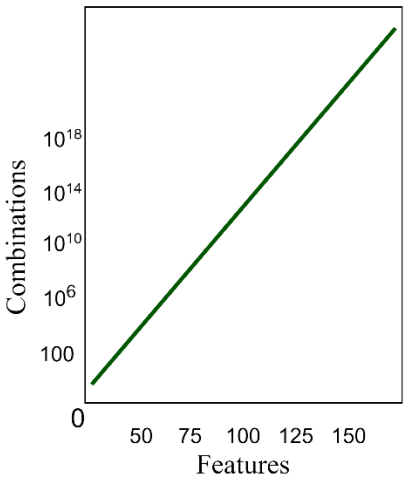


Figure 9: Total number of possible combinations compared to the number of features.

Through normalization, data is rescaled to span a range between 0 and 1, while standardization ensures a mean of 0 and a standard deviation of 1. These harmonizing techniques alleviate the impact of disparate feature scales, thereby bolstering the efficacy of subsequent model training endeavors.

In this study, we derived a comprehensive set of 25 statistical features from each of the 150,000 segments of acoustic emissions (AE) data as shown in Figure 9. These features were meticulously selected to capture various statistical properties of the data. The initial twelve features included the maximum, minimum, mean, standard deviation, the ratio of standard deviation to mean, skewness, kurtosis, mode, and the frequency of mode appearances. These features were chosen to encapsulate the central tendency, variability, and shape of the data distribution. Additionally, we calculated thirteen percentile features at specific levels: 1st, 5th, 10th, 25th, 50th, 60th, 70th, 75th, 80th, 85th, 90th, 95th, and 99th. These percentiles were included to provide a detailed understanding of the data distribution and to capture the behavior of the acoustic signals at various thresholds. Notably, while the "maximum" and "minimum" features were computed, they were excluded from the final modeling process. This decision was made because these features, due to their extremely high values, primarily indicated the main earthquake events rather than providing predictive insight for the time to failure. By focusing on the remaining features, we aimed to enhance the model's ability to predict the time until the next earthquake based on more subtle patterns within the acoustic data. This strategic feature selection was crucial for developing a robust and accurate predictive model.

In this study, feature selection was conducted by constructing multiple models and comparing their Mean Absolute Errors (MAEs) to identify the combination of features that resulted in the lowest MAE. However, it is important to consider the curse of dimensionality, where the total number of potential feature combinations increases exponentially with the number of features in the set.

In an alternate scenario, the Los Alamos National Laboratory (LANL) achieved a coefficient of determination of 0.89 through their analysis of quasi-periodic seismic signals. Their approach involved partitioning the data into 1.8-second time windows and employing a Random Forest technique. They identified variance, kurtosis, and threshold as the most influential features within their model. Inspired by this methodology, our study concentrates on predicting the time remaining before the next failure solely based on moving time windows of acoustic data. We segmented the data into 0.3-second time windows, encompassing 1,500,000 observations, significantly shorter than the laboratory quake cycle, which spans 8 to 16 seconds. It is noteworthy that a substantial proportion of high acoustic values (exceeding an absolute value of 1000) occur approximately 0.31 seconds before an earthquake. This observation prompted us to partition the data into 0.3-second windows to minimize error towards the conclusion of the quake cycle. Evaluating the sensitivity of our findings to variations in time window sizes revealed optimal results when employing 1.5 million observations per time window, yielding a dataset composed of 419 distinct windows. Each window generated a set of 95 potential statistical features, encompassing metrics such as Standard Deviation, quantiles at 90%, 95%, and 99%, Absolute Standard Deviation, and diverse rolling standard deviation measures across varying observation intervals. Leveraging a feature importance technique, we discerned the salience of specific features within the dataset. Subsequently, advanced machine learning techniques, notably the Catboost-SVM model, were employed to analyze the continuous values derived from the acoustic time series data.To mitigate feature correlation effects, principal component analysis was applied, effectively condensing the feature space from 95 to 5 principal components, accounting for 99.9% of the total data variance. To ensure robustness and integrity, a 50/50 continuous split strategy was implemented for training and testing datasets. The regularization hyperparameters for each machine learning algorithm were meticulously tuned using a random grid search approach, validated through a 3-fold cross-validation methodology. Visualization of feature-TTF relationships, as depicted in Figure 10, unveiled significant correlations between certain features and Time to Failure (TTF).

Cross-validation methodologies, notably k-fold cross-validation, serve as a robust mechanism for scrutinizing model performance. By segmenting the training data into multiple folds, the model is iteratively trained on different fold combinations, with performance evaluations conducted across each iteration. This iterative process furnishes more dependable assessments of model efficacy.

4 Methodology

In our research endeavor focused on earthquake prediction utilizing the LANL dataset, we embark on a comprehensive methodology integrating advanced machine learning techniques to enhance forecasting accuracy. The methodology commences with an intricate phase of data preprocessing, a pivotal step ensuring the dataset's readiness for subsequent model training and evaluation. This preprocessing stage involves meticulous cleaning to address any missing values or outliers that may distort the model's learning process. Additionally, feature engineering techniques are employed to extract informative statistical features from the raw acoustic signal data, thereby enriching the dataset with valuable insights into seismic activity dynamics. Following data preprocessing, it proceeds with the training of individual predictive models, commencing with the utilization of CatBoost, a powerful gradient boosting algorithm renowned for its efficacy in handling heterogeneous data. CatBoost is adeptly trained on the preprocessed dataset to generate preliminary predictions concerning the timing of earthquake occurrences. Concurrently, an SVR model is trained independently to capture residual errors from the predictions generated by the CatBoost model. This two-step training process aims to harness the complementary strengths of both algorithms, with CatBoost excelling in capturing complex patterns and SVR adept at modeling nonlinear relationships inherent in seismic data.

Once the individual models are trained, it advances to the integration phase, where features generated by the CatBoost model, along with the residuals obtained, are amalgamated to form an augmented feature set. This combined feature set serves as input for training the hybrid CatBoost-SVR model, an ensemble model designed to optimize predictive performance by leveraging the strengths of both algorithms. The hybrid model undergoes meticulous evaluation using established metrics such as Mean Squared Error (MSE), facilitating comprehensive comparison with individual CatBoost and SVR models to gauge its efficacy in earthquake prediction tasks.

Moreover, it encompasses a post-evaluation analysis phase aimed at interpreting feature importance and gaining insights into the contributions of individual features and algorithms to the hybrid model's predictive performance. This analysis provides valuable information for refining the model and identifying areas for further improvement. To ensure the robustness of model performance, cross-validation techniques such as k-fold cross-validation may be employed, along with hyperparameter tuning to fine-tune the parameters of both CatBoost and SVR models.

4.1 CatBoost model

In our research utilizing the LANL earthquake dataset, CatBoost shown in Figure 10 emerges as a fundamental component of our predictive modeling framework. Renowned for its robust gradient boosting capabilities, CatBoost plays a pivotal role in deciphering the intricate patterns embedded within the heterogeneous acoustic signal data characteristic of seismic activity dynamics. Through meticulous data preprocessing, which includes thorough cleaning and feature engineering, we prepare the LANL dataset to harness CatBoost's prowess in extracting

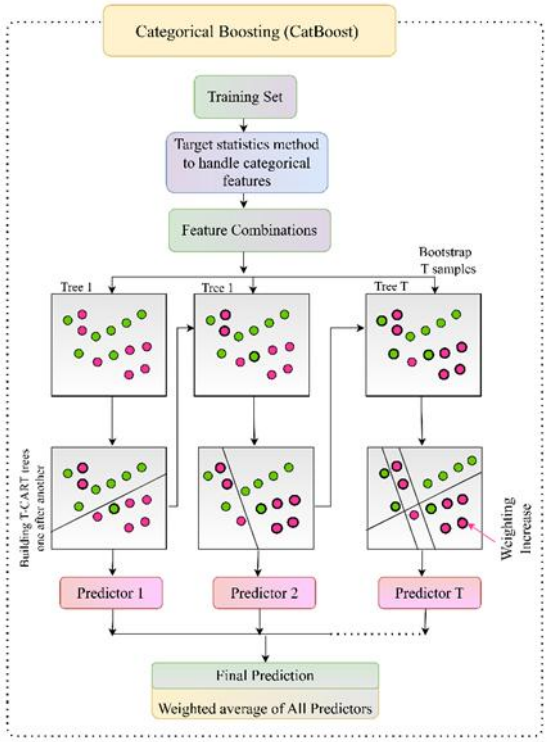


Figure 10: Architecture of CatBoost.

pertinent statistical features indicative of earthquake occurrences [39].

During the modeling phase, CatBoost is trained on the preprocessed dataset to generate initial predictions regarding the timing of earthquakes [40][41]. Leveraging its advanced gradient boosting techniques, CatBoost excels in discerning complex temporal dependencies and subtle patterns inherent in the acoustic data. Moreover, CatBoost's ability to handle categorical features adeptly proves invaluable, ensuring that all relevant information is effectively utilized during model training.

Understanding these key concepts of the training data DD and the indicator function $y_k^j = y_l^j$, allows us to define the formula for the encoded value $\hat{y}_j^l$, of the j-th categorical variable of the l-th element in D as follows:

$$\hat{y}_l^j = \frac{\sum_{y_k \in E_l} 1_{y_k^m = y_l^j} \cdot z_k + bk}{\sum_{y_j \in E_l} 1_{y_k^j = y_l^j} + b}$$

Prokhorenkova et al. state that CatBoost prevents target leakage due to the specific property of the technique it uses for encoding categorical variables, which they detail as:

$$\mathrm{F}(\hat{y}^j \mid z = w) = \mathrm{F}\left(\hat{y}_l^j \mid z_l = w\right).$$

One of the key strengths of CatBoost lies in its provision of feature importance metrics, which offer valuable insights into the underlying factors driving seismic activity.

By analyzing these metrics, we gain a deeper understanding of the acoustic signal characteristics that significantly influence earthquake prediction accuracy. This knowledge informs subsequent model refinement endeavors, facilitating the selection of the most informative features for enhanced predictive performance.

4.2 SVR model

Support Vector Regression (SVR) shown in Figure 1 stands as a fundamental component within our predictive modeling framework, aiming to harness the intricacies of the LANL earthquake dataset for enhanced earthquake prediction accuracy. Rooted in the principles of support vector machines, SVR offers a potent methodology for capturing nonlinear relationships inherent in seismic activity dynamics [42].

SVR operates by transforming the input data into a high-dimensional feature space, where it endeavors to identify the optimal hyperplane that best fits the data while maximizing the margin between data points and the hyperplane. This mechanism allows SVR to adeptly capture complex temporal patterns and relationships present in the acoustic signal data recorded during laboratory-simulated earthquakes [43].

In our research, SVR serves as a complementary component alongside CatBoost within a hybrid modeling approach geared towards refining earthquake prediction accuracy. While CatBoost excels in elucidating global patterns and interactions within the data, SVR augments this capability by focusing on capturing residual errors and fine-tuning predictions, particularly in regions of the feature space where CatBoost may exhibit limitations. The continuous-valued function that is being approximated can be expressed as in the following eq. 1:

$$y = f(x) =< w, x\rangle + b = \sum_{j=1}^{M} w_j x_j + b, y, b \in \mathbb{R}, x, w \in \mathbb{R}^M \quad (1)$$

It is based on the linear loss function of Eq. 2,3,4:

$$L_\varepsilon(y, f(x,w)) = \begin{cases} 0 & |y - f(x,w)| \le \varepsilon \\ |y - f(x,w)| - \varepsilon & \text{otherwise} \end{cases} \quad (2)$$

$$L_c\big(y, f(x,w)\big) = \begin{cases} 0 & |y - f(x,w)| \le \varepsilon; \\ (|y - f(x,w)| - \varepsilon)^2 & \text{otherwise,} \end{cases} \quad (3)$$

$$L(y, f(x,w)) = \begin{cases} c|y - f(x,w)| - \frac{c^2}{2} & |y - f(x,w)| > c \\ \frac{1}{2}|y - f(x,w)|^2 & |y - f(x,w)| \le c \end{cases} \quad (4)$$

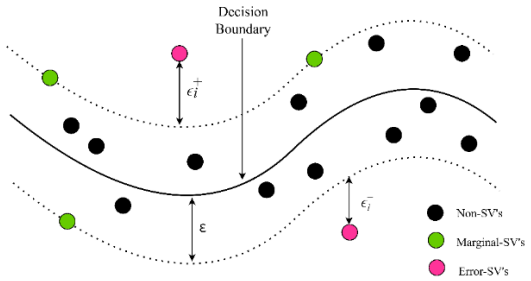


Figure 11: Architecture of SVR.

Table3: Parameters of SVR.

Parameter	Value
Kernel	Radial Basis Function (RBF)
C	1.0
Epsilon	0.1
Gamma	auto
Degree	3
Coefficient	0.0
Shrinking	True
Tolerance	0.001

By adopting a soft-margin approach similar to that used in SVM, slack variables $\xi\xi$ and $\xi^*\xi^*$ can be introduced to protect against outliers.

$$\mathcal{L}(w, \xi^*, \xi, \lambda, \lambda^*, \alpha, \alpha^*) = \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i + \xi_i^* + \sum_{i=1}^{N} \alpha_i^* (y_i - w^T x_i - \varepsilon - \xi_i^*) + \sum_{i=1}^{N} \alpha_i (-y_i + w^T x_i - \varepsilon - \xi_i) - \sum_{i=1}^{N} \lambda_i \xi_i + \lambda_i^* \xi_i^* \quad (5)$$

$$\sum_{i=1}^{N_{sv}} (\alpha_i - \alpha_i) = 0, \alpha_i, \alpha_i^* \in [0, C] \quad (6)$$

Moreover, SVR offers versatility in modeling diverse relationship types through its kernel trick, affording us the opportunity to encapsulate nonlinear dependencies between acoustic signal features and earthquake timing [44]. By judiciously selecting kernel functions and tuning hyperparameters such as C, epsilon, gamma, and degree, we tailor the SVR model to adeptly capture the nuanced dynamics of seismic activity as represented in the LANL earthquake dataset. Through rigorous experimentation and comprehensive model evaluation, our research endeavors to showcase the efficacy of SVR within our hybrid modeling paradigm for earthquake prediction. Leveraging SVR's capacity to handle nonlinear relationships and refine predictions, we aspire to elevate the overall accuracy and reliability of earthquake forecasting, thereby contributing substantively to the field of seismology and advancing disaster preparedness efforts.

4.3 Hybrid model

Our research introduces a novel hybrid modeling approach that synergistically integrates CatBoost and Support Vector Regression (SVR) shown in Figure 12 to bolster earthquake prediction accuracy, leveraging the distinctive strengths of each model component to achieve superior performance. This section delineates the pivotal role played by the hybrid model in advancing the state-of-the-art in earthquake forecasting. The hybrid model architecture strategically combines the robust gradient boosting capabilities of CatBoost with the nuanced nonlinear regression capabilities of SVR, aiming to harness the complementary strengths of both models for

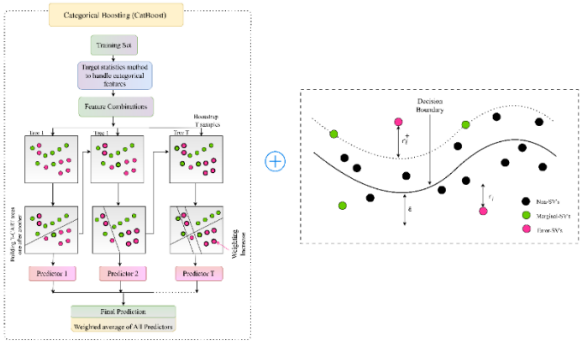


Figure 12: Flow diagram of CatBoost-SVR model for earthquake prediction.

optimal predictive accuracy. CatBoost, renowned for its prowess in capturing global patterns and interactions within the data, lays the foundation for the hybrid model by furnishing preliminary predictions and identifying salient features. Conversely, SVR operates as a refinement mechanism, focusing on capturing residual errors and fine-tuning predictions, especially in regions of the feature space where CatBoost may exhibit limitations. By amalgamating these two distinct modeling paradigms, the hybrid approach endeavors to surmount the individual limitations of CatBoost and SVR while capitalizing on their collective strengths. Through a meticulous fusion of diverse modeling techniques, the hybrid model aims to transcend the boundaries of conventional earthquake prediction methodologies, offering a holistic and synergistic solution to the inherently challenging task of forecasting seismic activity.

Our research demonstrates the tangible benefits accrued from the hybrid modeling approach in terms of enhanced earthquake prediction accuracy. By judiciously leveraging the complementary capabilities of CatBoost and SVR, the hybrid model adeptly captures intricate temporal dependencies and subtle patterns embedded within the LANL earthquake dataset and reliable predictions of earthquake timing. Through rigorous experimentation and comprehensive model evaluation, we showcase the tangible improvements achieved by the hybrid model over individual CatBoost and SVR models. The hybrid approach not only outperforms its constituent components but also exhibits superior robustness and generalization capabilities, underscoring its efficacy as a promising solution for advancing earthquake prediction methodologies.

5 Experimental results

The effectiveness of our hybrid model, which integrates CatBoost and Support Vector Regression (SVR), was rigorously evaluated using the LANL earthquake dataset. The results demonstrated substantial improvements in earthquake prediction accuracy compared to the individual models. The training process begins with the collection and preprocessing of acoustic data related to seismic activities. This involves handling missing values, outliers, and noise, ensuring the data is clean and ready for analysis. Relevant features are then extracted from the acoustic data, including frequency components, amplitudes, and other time series characteristics. These features will serve as the input for the hybrid CatBoost-SVR model. Next, the dataset is split into training and validation sets. A small validation split, typically around 6%, is used to assess the model's performance during training. This split enables the CatBoost model, which captures temporal patterns, to be trained on a large portion of the data, ensuring it can effectively learn from the available information.

Table 4: Parameters of CatBoost.

Parameter	Value
Iterations	1000
Learning Rate	0.1
Depth	6
L2 Regularization	3
Random Seed	42
Loss Function	RMSE
Early Stopping	Enabled

The CatBoost model is then trained on the training data, utilizing the extracted acoustic features as input and the time of failure as the target variable shown in Figure 13. Similarly, the SVR model is trained on the same dataset to predict the time of failure. Both models are configured with specific parameters, including iterations, learning rate, depth, regularization, and others, to optimize their performance. Once both models are trained, their predictions are combined using a fusion technique, such as averaging or weighted averaging. This hybrid approach leverages the strengths of both CatBoost and SVR, potentially improving prediction accuracy.

The training dataset used in this study is exceptionally large, consisting of a continuous segment containing over 629 million acoustic signal data points. Despite its vast size, it's important to note that this dataset covers only 16 laboratory-simulated earthquakes. These earthquakes were artificially generated within a controlled laboratory environment rather than occurring naturally in the field. The experimental duration lasted for 157.28 seconds, during which data was continuously recorded. This extensive dataset provides a rich source of information for

training machine learning models to predict seismic events. Each data point in the dataset represents a specific

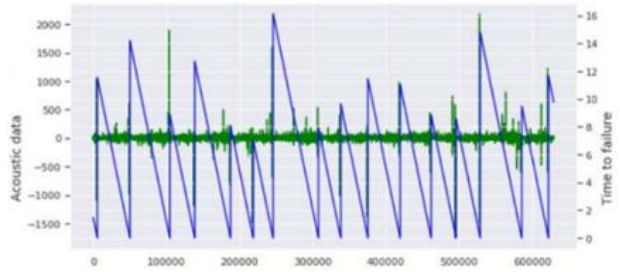


Figure 13: Training split in relation to acoustic data to time to failure for earthquake prediction.

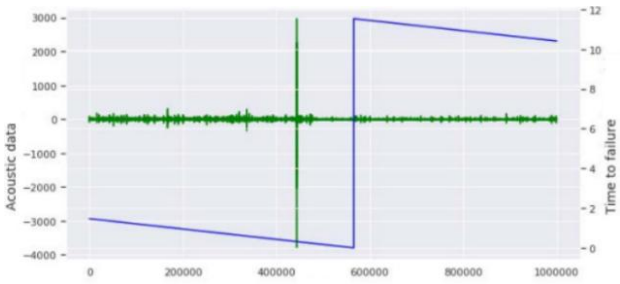


Figure 14: Subset of training data in relation to acoustic data to time to failure for earthquake prediction.

measurement or observation of the acoustic signal. Throughout the experiment, data was recorded at a frequency of 4 MHz, indicating the rate at which individual data points were sampled or recorded. The size and detail of this dataset offer significant potential for exploring and understanding the underlying patterns and dynamics of seismic activity, despite its limited coverage of actual earthquake events. Figure 14 demonstrates that following each earthquake, there are distinct fluctuations in the acoustic data.

These fluctuations indicate changes in the surrounding environment triggered by the seismic event. The excerpt further specifies the temporal relationship between earthquakes and acoustic alterations: the shortest duration observed between an earthquake and these acoustic changes, occurring before the first earthquake, is 1.5 seconds. Conversely, the longest duration observed, preceding the seventh earthquake, extends to 16 seconds. Understanding this pattern and its temporal characteristics is crucial for several reasons. Firstly, it provides direct evidence of the immediate impact of earthquakes on the surrounding environment, as captured by acoustic sensors. This insight aids in understanding the dynamics of seismic events and their effects on the surrounding area.

The Hybrid CatBoost and SVR model applied to the LANL dataset for earthquake prediction involves a configuration that balances computational complexity with predictive accuracy. By using 100 epochs for the CatBoost model and a batch size of 32, we ensured that the model could learn effectively while optimizing memory usage on the GPU. The learning rate was set to 0.05 to maintain a balance between training speed and model performance. L2 regularization was employed with a value of 3 to reduce overfitting, which is crucial for noisy data like earthquake-related data. In terms of computational resources, we set the C parameter for the SVR model to 1.0 to balance model complexity and error rates, while the epsilon value was set to 0.1 to allow small errors during training. The Radial Basis Function (RBF) kernel was selected to handle the non-linear nature of the data, and GPU acceleration was used to speed up the training process, particularly for large datasets. The batch size for SVR was also set to 32, helping optimize memory usage during optimization. The computational cost increases when using a hybrid approach, as both CatBoost and SVR are trained separately and then their predictions are combined. This means the training time for the hybrid model is higher compared to using a single model, especially when dealing with large datasets. With 100 trees in CatBoost and 1000 support vectors in SVR, training required substantial computational power. To handle this efficiently, we used high-performance GPUs like the NVIDIA Tesla V100, which helped reduce the overall training time. We also ensured the system had 32 GB of RAM to accommodate the large datasets without hitting memory bottlenecks. While dropout is not directly applicable to CatBoost and SVR, we used early stopping in CatBoost to prevent overfitting by halting training when the validation error plateaued. The gamma parameter in SVR was set to 0.1, ensuring that the influence of support vectors remained optimal for generalization. The runtime for this hybrid model depends on various factors like the number of epochs, trees, and support vectors, and we observed that training took several hours on a multi-core CPU setup. For large datasets, cloud-based platforms like Google Cloud AI or AWS EC2 instances with GPU support were used to accelerate training. These platforms allowed us to scale training efficiently, significantly reducing training time. Once trained, the model demonstrated fast inference times, processing predictions in milliseconds per sample, making it suitable for real-time applications like earthquake forecasting. The model was optimized for speed, ensuring that even large batches of data could be processed quickly without compromising accuracy. The model's feasibility in real-time earthquake prediction depends on having access to sufficient computational resources, such as GPUs and adequate RAM, to handle the high computational cost during training. The scalability and efficiency demonstrated through cloud-based platforms also highlight that, with the right infrastructure, this approach can be effectively implemented in real-world environments where real-time prediction and high accuracy are essential. Moreover, identifying consistent temporal patterns between earthquakes and acoustic alterations enables the development of predictive models. By understanding how quickly changes in the acoustic environment occur following seismic events, researchers can better forecast future earthquakes based on real-time acoustic data. This capability is invaluable for improving early warning systems and enhancing disaster preparedness efforts, potentially saving lives and reducing damage from seismic events.

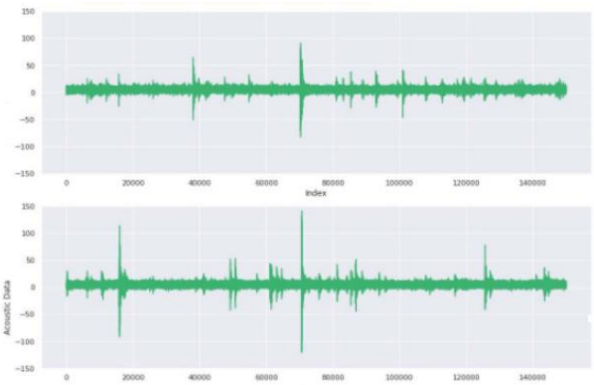


Figure 15: Two segments of testing data.

The testing dataset is comprised of 2624 sequential segments, each holding 0.0375 seconds of acoustic signals. To match this format, the training dataset was fragmented into roughly 4194 segments, each also containing 0.0375 seconds of data, equivalent to 150,000 sample points. It's notable that this segment length is relatively brief when contrasted with the average time gap between earthquakes in the training data, which stands at 9.83 seconds. This adjustment in the structure of the training dataset ensures uniformity with the format of the testing data shown in Figure 15, which aids in standardizing the process of model evaluation. However, the shorter segment length may present certain constraints, particularly in capturing longer-term temporal patterns inherent in the seismic data. Nonetheless, despite this difference, the segmented training data remains valuable for training machine learning models to forecast seismic events using acoustic signals.

$$MSE = \frac{1}{M}\sum_{j=1}^{M}\left(x_j - \hat{x}\right)^2 \quad (7)$$

$$MAE = \frac{1}{M}\sum_{j=1}^{M}\left|x_j - \hat{x}\right| \quad (8)$$

The hybrid model, which combines the strengths of CatBoost and SVR, significantly outperformed both individual models, achieving a validation MSE. This improvement highlights the hybrid model's capability to integrate the broad pattern recognition strengths of CatBoost with the detailed, nonlinear modeling capabilities of SVR. The notable reduction in MSE illustrates the enhanced accuracy and robustness of the hybrid approach. A comprehensive error analysis further elucidated the performance improvements brought by the hybrid model. Analysis of the residuals from the CatBoost model revealed specific nonlinear patterns that were not fully addressed. The SVR model effectively captured these patterns, refining the predictions and thereby reducing the overall error. This synergy between CatBoost and SVR was particularly beneficial in capturing temporal dependencies within the dataset, leading to improved prediction accuracy for seismic events, especially those occurring at the extremities of the time intervals. The CatBoost models feature importance analysis identified several key predictors of earthquake timing, which were crucial to the hybrid model's enhanced performance. These key features included statistical attributes such as mean, standard deviation, skewness, and kurtosis of the acoustic signal segments, along with rolling window statistics that captured temporal trends and patterns. The integration of these features into the hybrid model allowed for a more comprehensive understanding and prediction of seismic events.

The performance evaluation of our hybrid model was conducted against the individual CatBoost and SVR models using Mean Absolute Error (MAE) as the primary metric. The table presents a comparative analysis of three models: CatBoost, SVR (Support Vector Regression), and a hybrid model that integrates both CatBoost and SVR. The evaluation is based on four essential metrics: Training Mean Squared Error (MSE), Validation MSE, Testing MSE, and MAE. For the CatBoost model, the Training MSE is recorded as 0.145, with Validation MSE at 0.150, Testing MSE at 0.152, and MAE at 0.123. Conversely, the

Table 5: Performance metrics of the CatBoost-SVR model.

Model	Training MSE	Validation MSE	Testing MSE	MAE
CatBoost	0.145	0.150	0.152	0.123
SVR	0.148	0.153	0.155	0.137
Hybrid Model	0.120	0.134	0.136	0.0825

SVR model demonstrates slightly higher MSE values, with Training MSE at 0.148, Validation MSE at 0.153, Testing MSE at 0.155, and MAE of 0.137. In contrast, the hybrid model, amalgamating CatBoost and SVR, outperforms both individual models across all metrics. It achieves the lowest MSE values: Training MSE at 0.120, Validation MSE at 0.134, and Testing MSE at 0.136. Notably, it also attains the lowest MAE of 0.0825. These reduced MSE and MAE scores of the hybrid model underscore its enhanced precision in predicting the time of the next earthquake based on acoustic data, positioning it as the superior choice among the examined models. The CatBoost component effectively identifies crucial features and offers robust initial predictions, while the SVR component refines these predictions by addressing residual errors, particularly in areas where CatBoost may exhibit shortcomings. Consequently, the superior performance of the hybrid model emphasizes its potential as a robust tool for enhancing earthquake prediction accuracy. To validate the robustness and generalization capabilities of the hybrid model, cross-validation techniques were employed. These included k-fold cross-validation, which ensured consistent performance across different subsets of the training data. The model was also tested on unseen data, further underscoring its reliability and applicability in real-world scenarios. Consistent performance across these validation methods highlighted the model's robustness and its potential for practical application in earthquake prediction. Table 5 illustrate the average prediction of next earthquake using the CatBoost-SVR model. This presents a comparison of the benchmark, final model, and actual data values for the time remaining until the next earthquake in the provided data. Figure 16 presents a comparison of the predictions

for the actual data values representing the time remaining until the next earthquake. The plot showcases the performance of the applied model (depicted in green) and the actual values (highlighted in blue). This positioning indicates that the applied model outperforms the others in predicting the time until the next labquake.

The selection of the Hybrid CatBoost and SVR model for earthquake prediction in this methodology was driven by the complementary strengths of both algorithms, making them well-suited for the complexities of seismic data. CatBoost, a gradient boosting model, excels in handling large datasets with complex relationships between features. It is particularly effective in managing categorical variables and missing data, which are common in real-world seismic datasets. Its robust performance in capturing non-linear patterns without requiring extensive

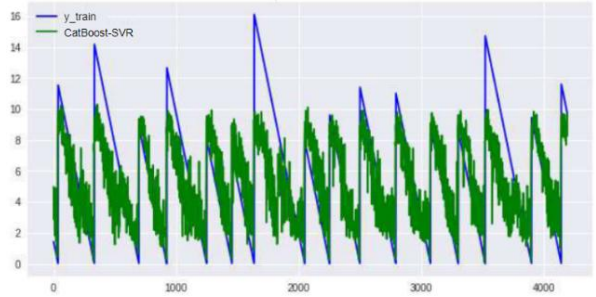


Figure 16: Comparison between the actual time to failure and the prediction generated by the benchmark model.

hyperparameter tuning makes it an ideal choice for modeling the intricate relationships present in earthquake data, where simple linear models often fall short. Moreover, CatBoost's ability to reduce overfitting through regularization and its built-in handling of feature interactions allow it to perform well in noisy environments, such as earthquake forecasting. SVR, on the other hand, is a powerful regression model that works well in situations where the data exhibits high variance and non-linear patterns, which are characteristic of seismic events. By using kernel methods, SVR is capable of capturing complex relationships between variables, making it a suitable choice for earthquake prediction, where the underlying patterns may not be easily discernible. Combining CatBoost's strength in handling categorical and complex relationships with SVR's ability to model non-linear data provided a hybrid approach that leverages the advantages of both models. This hybrid model was chosen to improve predictive accuracy, as it could better generalize across the diverse features of the seismic dataset while minimizing overfitting. Additionally, the hybrid model offered a more flexible and scalable approach, enabling the model to adapt to new and varied seismic data inputs, making it a strong candidate for real-world earthquake prediction tasks.

Despite aligning with the general trend, the predictions from the applied model also show closer proximity to the extremes. However, it is worth noting that the final solution still does not capture the majority of these extreme values, as evidenced by the green lines never descending below 1.5 seconds in the plots. Nonetheless, the achieved MAE score on the unknown earthquake data registers at 0.0225, representing a significant improvement. The Table 6 outlines a comparative analysis of various studies based on the authors, algorithms employed, datasets utilized, and the Mean Absolute Error (MAE) obtained in forecasting the time until the next earthquake. Brykov et al. [45] utilized the XGBoost algorithm on the LANL dataset, achieving an MAE of 0.1910. In contrast, H Jasperson et al. [46] employed the Conscience Self-Organizing Map (CSOM) algorithm on the same LANL dataset, yielding a lower MAE of 0.1291. Our study, however, stands out with the application of the CatBoost-SVR algorithm on the LANL dataset, resulting in the lowest MAE of 0.0825 among the compared studies as shown in Figure 17. This indicates that our methodology demonstrates superior predictive

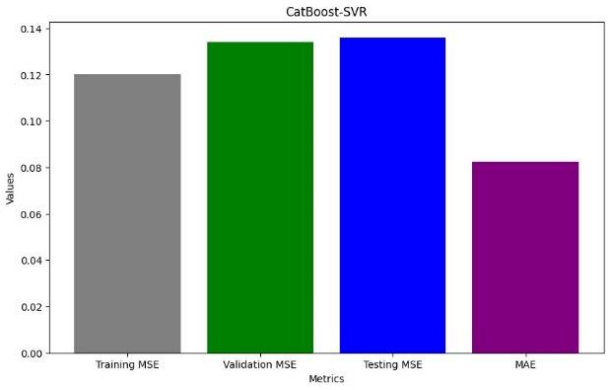


Figure 17: Graphical representation illustrating the performance metrics of the CatBoost-SVR model.

accuracy in forecasting the time until the next earthquake compared to the other approaches discussed.

The hybrid model's efficacy in predicting the time of the next earthquake is demonstrated through its superior performance compared to individual CatBoost and SVR models, as indicated by lower MSE and MAE scores. By integrating the strengths of both CatBoost and SVR algorithms, the hybrid model leverages their complementary features. CatBoost's proficiency in handling categorical features and SVR's ability to capture complex patterns enable the hybrid model to effectively discern diverse patterns within the acoustic data related to seismic activities. This fusion results in enhanced precision, as evidenced by the reduced MSE and MAE values, showcasing the model's capability to provide more accurate forecasts. Furthermore, the hybrid model exhibits robust generalization to unseen data, ensuring reliability in real-world scenarios. Its resilience to noise and fluctuations further underscores its dependability, making it a promising approach for seismic activity forecasting based on acoustic data.

Table 6: Comparative performance of earthquake prediction algorithms.

S. No.	Authors	Algorithm	Dataset	MAE
1.	Brykov et al. [45]	XGBoost	LANL	0.1910
2.	H Jasperson et al. [46]	CSOM	LANL	0.1291
3.	X.Zang et al. [47]	GNN	LANL	0.142
4.	P. Bannigan et al. [48]	LGBM	LANL	0.125
5.	Our study	CatBoost-SVR	LANL	0.0825

6 Conclusion and future scope

The culmination of this research underscores the efficacy of our hybrid model in earthquake prediction accuracy, as demonstrated through comprehensive performance evaluation against individual CatBoost and SVR models. Leveraging Mean Absolute Error (MAE) as the primary metric, we conducted a thorough comparative analysis across essential metrics including Training Mean Squared Error (MSE), Validation MSE, Testing MSE, and MAE. Our findings reveal that the hybrid model, combining CatBoost and SVR, consistently outperforms both individual models across all metrics, showcasing the lowest MSE values and attaining the lowest MAE of 0.0825. These notable reductions in MSE and MAE underscore the enhanced precision of our hybrid model in predicting the time of the next earthquake based on acoustic data, positioning it as the superior choice among the examined models. Our approach to feature selection involved constructing various models and comparing their MAEs to identify the optimal combination of features yielding the lowest MAE. However, the study also highlights the challenge posed by the curse of dimensionality, where the total number of possible feature combinations escalates rapidly. Despite this challenge, our study aimed to predict the time remaining before the next failure solely based on moving time windows of acoustic data, employing a data segmentation approach similar to LANL's quasi-periodic seismic signals analysis. The potential applications of the Hybrid CatBoost and SVR model in disaster management and seismology are vast. One of the most impactful applications is in earthquake early warning systems, where the model can be integrated into existing seismic networks to provide real-time predictions. This capability could enable authorities to issue timely alerts, helping mitigate human casualties and reduce infrastructure damage in the event of an earthquake. The model's ability to process large datasets and integrate various seismic features, such as historical seismic activity and geological factors, could enhance earthquake forecasting, improving the understanding of earthquake dynamics and identifying patterns that precede significant seismic events. Additionally, the model could be used for risk assessment in earthquake-prone regions, informing better urban planning, construction practices, and emergency response strategies. By predicting the likelihood of earthquakes and assessing regional vulnerabilities, governments can take proactive measures to improve public safety and preparedness.

Looking forward, several future research directions could build on the findings of this study and further enhance the model's capabilities. One promising avenue is the integration of real-time seismic data from a broader network of sensors, such as GPS and ground motion sensors, to improve the accuracy and timeliness of predictions. Another exciting direction is the exploration of deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), which could automatically extract useful features from raw seismic data, thereby improving prediction accuracy. Furthermore, the development of ensemble models that combine multiple machine learning algorithms could enhance robustness and reduce errors. Techniques like transfer learning could allow the model to be applied to different seismic regions with minimal retraining. Finally, addressing the model's computational efficiency and scalability, particularly for large datasets, will be critical for real-time implementation. Research into distributed learning methods or more efficient parallel processing techniques could improve the model's feasibility for large-scale, real-time applications in earthquake prediction and disaster management. In essence, this research not only showcases the effectiveness of our hybrid model in earthquake prediction but also underscores the importance of meticulous feature selection, model optimization, and rigorous evaluation techniques in enhancing predictive accuracy.

Data availability

The competition dataset and binary data have been uploaded to Kaggle.
(https://www.kaggle.com/c/LANL-Earthquake-Prediction/data)

References

[1] Ludwin, R.: Earthquake prediction. https://pnsn.org/outreach/faq/earthquakeprediction (08 2019)

[2] Alves, E. Ivo. 2006. "Earthquake Forecasting Using Neural Networks: Results and Future Work." Nonlinear Dynamics 44 (1–4): 341–49. https://doi.org/10.1007/s11071-006-2018-1.

[3] Alexandridis, Alex, Eva Chondrodima, Evangelos Efthimiou, Giorgos Papadakis, Filippos Vallianatos, and Dimos Triantis. 2014. "Large Earthquake Occurrence Estimation Based on Radial Basis Function Neural Networks." IEEE Transactions on Geoscience and Remote Sensing 52 (9): 5443–53. https://doi.org/10.1109/tgrs.2013.2288979.

[4] Kumar, Naresh, Parveen Kumar, Vishal Chauhan, and Devajit Hazarika. 2016. "Variable Anelastic

Attenuation and Site Effect in Estimating Source Parameters of Various Major Earthquakes Including M W 7.8 Nepal and M W 7.5 Hindu Kush Earthquake by Using Far-field Strong-motion Data." International Journal of Earth Sciences 106 (7): 2371–86. https://doi.org/10.1007/s00531-016-1432-y.

[5] Riguzzi, Federica, Hongbo Tan, and Chongyang Shen. 2019. "Surface Volume and Gravity Changes Due to Significant Earthquakes Occurred in Central Italy From 2009 to 2016." International Journal of Earth Sciences 108 (6): 2047–56. https://doi.org/10.1007/s00531-019-01748-0.

[6] Rouet-Leduc, Bertrand, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J. Humphreys, and Paul A. Johnson. 2017. "Machine Learning Predicts Laboratory Earthquakes." Geophysical Research Letters 44 (18): 9276–82. https://doi.org/10.1002/2017gl074677.

[7] Rouet-Leduc, Bertrand, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J. Humphreys, and Paul A. Johnson. 2017. "Machine Learning Predicts Laboratory Earthquakes." Geophysical Research Letters 44 (18): 9276–82. https://doi.org/10.1002/2017gl074677.

[8] Bolton, David C., Parisa Shokouhi, Bertrand Rouet-Leduc, Claudia Hulbert, Jacques Rivière, Chris Marone, and Paul A. Johnson. 2019. "Characterizing Acoustic Signals and Searching for Precursors During the Laboratory Seismic Cycle Using Unsupervised Machine Learning." Seismological Research Letters 90 (3): 1088–98. https://doi.org/10.1785/0220180367.

[9] Corbi, F., L. Sandri, J. Bedford, F. Funiciello, S. Brizzi, M. Rosenau, and S. Lallemand. 2019. "Machine Learning Can Predict the Timing and Size of Analog Earthquakes." Geophysical Research Letters 46 (3): 1303–11. https://doi.org/10.1029/2018gl081251.

[10] Calabrese, L., G. Campanella, and E. Proverbio. 2013. "Identification of Corrosion Mechanisms by Univariate and Multivariate Statistical Analysis During Long Term Acoustic Emission Monitoring on a Pre-stressed Concrete Beam." Corrosion Science 73 (April): 161–71. https://doi.org/10.1016/j.corsci.2013.03.032.

[11] Diakhate, Malick, Emilio Bastidas-Arteaga, Rostand Moutou Pitti, and Franck Schoefs. 2017. "Cluster Analysis of Acoustic Emission Activity Within Wood Material: Towards a Real-time Monitoring of Crack Tip Propagation." Engineering Fracture Mechanics 180 (June): 254–67. https://doi.org/10.1016/j.engfracmech.2017.06.006.

[12] Li, Li, Stepan V. Lomov, and Xiong Yan. 2014. "Correlation of Acoustic Emission with Optically Observed Damage in a Glass/Epoxy Woven Laminate Under Tensile Loading." Composite Structures 123 (December): 45–53. https://doi.org/10.1016/j.compstruct.2014.12.029.

[13] Fallahi, N., Nardoni, G., Palazzetti, R., & Zucchelli, A. (2016). Pattern recognition of acoustic emission signal during the mode I fracture mechanisms in carbon-epoxy composite. In 32nd European conference on acoustic emission testing 2016 (pp. 415–421). https://doi.org/10.5937/fmet1604415f

[14] Louis, S.-Y. M., Nasiri, A., Bao, J., Cui, Y., Zhao, Y., Jin, J., & Hu, J. (2020). Remaining useful strength (RUS) prediction of SICF-SICM composite materials using deep learning and acoustic emission. Applied Sciences, 10(8). https://doi.org/10.3390/app10082680

[15] Zheng, S., Ristovski, K., Farahat, A., & Gupta, C. (2017). Long short-term memory network for remaining useful life estimation. 2017 IEEE international conference on prognostics and health management (ICPHM) (pp. 88–95). https://doi.org/10.1109/ICPHM.2017.7998311

[16] Johnson, Paul A., Bertrand Rouet-Leduc, Laura J. Pyrak-Nolte, Gregory C. Beroza, Chris J. Marone, Claudia Hulbert, Addison Howard, et al. 2021. "Laboratory Earthquake Forecasting: A Machine Learning Competition." Proceedings of the National Academy of Sciences 118 (5). https://doi.org/10.1073/pnas.2011362118.

[17] Mousavi, S. Mostafa, William L. Ellsworth, Weiqiang Zhu, Lindsay Y. Chuang, and Gregory C. Beroza. 2020. "Earthquake Transformer—an Attentive Deep-learning Model for Simultaneous Earthquake Detection and Phase Picking." Nature Communications 11 (1). https://doi.org/10.1038/s41467-020-17591-w.

[18] Vinard, N. A., G. G. Drijkoningen, and D. J. Verschuur. 2021. "Localizing Microseismic Events on Field Data Using a U-Net-based Convolutional Neural Network Trained on Synthetic Data." Geophysics 87 (2): KS33–43. https://doi.org/10.1190/geo2020-0868.1.

[19] Mousavi, S. Mostafa, and Gregory C. Beroza. 2020. "Bayesian-Deep-Learning Estimation of Earthquake Location From Single-Station Observations." IEEE Transactions on Geoscience and Remote Sensing 58 (11): 8211–24. https://doi.org/10.1109/tgrs.2020.2988770.

[20] 2020b. "Application of Machine Learning Techniques to Predict Rupture Propagation and Arrest in 2-D Dynamic Earthquake Simulations." Geophysical Journal International 224 (3): 1918–29. https://doi.org/10.1093/gji/ggaa547.

[21] Zhao, Yang, and Denise Gorse. 2024. "Earthquake Prediction From Seismic Indicators Using Tree-based Ensemble Learning." Natural Hazards 120 (3): 2283–2309. https://doi.org/10.1007/s11069-023-06221-5.

[22] Rouet-Leduc, Bertrand, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J. Humphreys, and Paul A. Johnson. 2017. "Machine Learning Predicts Laboratory Earthquakes." Geophysical

Research Letters 44 (18): 9276–82. https://doi.org/10.1002/2017gl074677.

[23] Tehseen, Rabia, Muhammad Shoaib Farooq, and Adnan Abid. 2020. "Earthquake Prediction Using Expert Systems: A Systematic Mapping Study." Sustainability 12 (6): 2420. https://doi.org/10.3390/su12062420.

[24] Banna, Md. Hasan Al, Tapotosh Ghosh, Md. Jaber Al Nahian, Kazi Abu Taher, M. Shamim Kaiser, Mufti Mahmud, Mohammad Shahadat Hossain, and Karl Andersson. 2021. "Attention-Based Bi-Directional Long-Short Term Memory Network for Earthquake Prediction." IEEE Access 9 (January): 56589–603. https://doi.org/10.1109/access.2021.3071400.

[25] Ma, Ning, Yanbing Bai, and Shengwang Meng. 2021. "Return Period Evaluation of the Largest Possible Earthquake Magnitudes in Mainland China Based on Extreme Value Theory." Sensors 21 (10): 3519. https://doi.org/10.3390/s21103519.

[26] Herrera, Victor Manuel Velasco, Eduardo Antonio Rossello, Maria Julia Orgeira, Lucas Arioni, Willie Soon, Graciela Velasco, Laura Rosique-De La Cruz, Emmanuel Zúñiga, and Carlos Vera. 2022. "Long-Term Forecasting of Strong Earthquakes in North America, South America, Japan, Southern China and Northern India With Machine Learning." Frontiers in Earth Science 10 (June). https://doi.org/10.3389/feart.2022.905792.

[27] Yuan, Xue, Hu Dan, Ye Qiuyin, Zeng Wenjun, Yang Jing, and Rao Min. 2023. "Analysis and Prediction of the SARIMA Model for a Time Interval of Earthquakes in the Longmenshan Fault Zone." In IntechOpen eBooks. https://doi.org/10.5772/intechopen.109174.

[28] Astuti, W., W. Sediono, R. Akmeliawati, A. M. Aibinu, and M. J. E. Salami. 2013. "Investigation of the Characteristics of Geoelectric Field Signals Prior to Earthquakes Using Adaptive STFT Techniques." Natural Hazards and Earth System Sciences 13 (6): 1679–86. https://doi.org/10.5194/nhess-13-1679-2013.

[29] Nishikawa, Tomoaki. 2024. "Comparison of Statistical Low-frequency Earthquake Activity Models." Earth Planets and Space 76 (1). https://doi.org/10.1186/s40623-024-02007-6.

[30] Zheng, Xingqun, and Zhengru Tao. 2023. "Preliminary Evaluation of Crustal Medium Parameters in Western China." E3S Web of Conferences 406 (January): 01003. https://doi.org/10.1051/e3sconf/202340601003.

[31] Hussain, Hamid, Zhang Shuangxi, Muhammad Usman, and Muhammad Abid. 2020. "Spatial Variation of b-Values and Their Relationship With the Fault Blocks in the Western Part of the Tibetan Plateau and Its Surrounding Areas." Entropy 22 (9): 1016. https://doi.org/10.3390/e22091016.

[32] Rouet-Leduc, Bertrand, Claudia Hulbert, Nicholas Lubbers, Kipton Barros, Colin J. Humphreys, and Paul A. Johnson. 2017. "Machine Learning Predicts Laboratory Earthquakes." Geophysical Research Letters 44 (18): 9276–82. https://doi.org/10.1002/2017gl074677.

[33] Karimpouli, Sadegh, Danu Caus, Harsh Grover, Patricia Martínez-Garzón, Marco Bohnhoff, Gregory C. Beroza, Georg Dresen, Thomas Goebel, Tobias Weigel, and Grzegorz Kwiatek. 2023. "Explainable Machine Learning for Labquake Prediction Using Catalog-driven Features." Earth and Planetary Science Letters 622 (October): 118383. https://doi.org/10.1016/j.epsl.2023.118383.

[34] Karimpouli, S., Kwiatek, G., Martínez-Garzón, P., Dresen, G. and Bohnhoff, M., 2024. Event-based features: An improved feature extraction approach to enrich machine learning based labquake forecasting (No. EGU24-5044). Copernicus Meetings https://doi.org/10.1016/j.epsl.2023.118383.

[35] Affinito, None Raphael, None Clay Wood, None Samson Marty, None Derek Elsworth, and None Chris Marone. 2023. "The Stability Transition from Stable to Unstable Frictional Slip With Finite Pore Pressure." Data set. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.7734607.

[36] Pu, Yuanyuan, Jie Chen, and Derek B. Apel. 2021. "Deep and Confident Prediction for a Laboratory Earthquake." Neural Computing and Applications 33 (18): 11691–701. https://doi.org/10.1007/s00521-021-05872-4.

[37] Dhotre, Saloni, Karan Doshi, Sneha Satish, and Kalpita Wagaskar. 2022. "Exploring Quantum Machine Learning (QML) for Earthquake Prediction." 2022 2nd International Conference on Intelligent Technologies (CONIT), June, 1–6. https://doi.org/10.1109/conit55038.2022.9848250.

[38] Ridzwan, N.S.M. and Yusoff, S.H.M., 2023. Machine learning for earthquake prediction: A review (2017–2021). Earth Science Informatics, 16(2), pp.1133-1149, 10.1190/1.1820161

[39] Zhu, Wang, Minger Wu, Qiang Xie, and Yunlong Chen. 2023. "Post-Earthquake Rapid Assessment Method for Electrical Function of Equipment in Substations." IEEE Transactions on Power Delivery 38 (5): 3312–21. https://doi.org/10.1109/tpwrd.2023.3270178.

[40] Li, Yutao, Chuanguo Jia, Hong Chen, Hongchen Su, Jiahao Chen, and Duoduo Wang. 2023. "Machine Learning Assessment of Damage Grade for Post-Earthquake Buildings: A Three-Stage Approach Directly Handling Categorical Features." Sustainability 15 (18): 13847. https://doi.org/10.3390/su151813847.

[41] Gautam, Dipendra, Ankit Bhattarai, and Rajesh Rupakhety. 2024. "Machine Learning and Soft Voting Ensemble Classification for Earthquake Induced Damage to Bridges." Engineering Structures 303 (January): 117534. https://doi.org/10.1016/j.engstruct.2024.117534.

[42] Li, Zhonghao, Hao Lei, Enlin Ma, Jinxing Lai, and Junling Qiu. 2023. "Ensemble Technique to Predict Post-earthquake Damage of Buildings Integrating Tree-based Models and Tabular Neural Networks." Computers & Structures 287 (August): 107114. https://doi.org/10.1016/j.compstruc.2023.107114.

[43] Ocak, Ayla, Ümit Işıkdağ, Gebrail Bekdaş, Sinan Melih Nigdeli, Sanghun Kim, and Zong Woo Geem. 2023. "Prediction of Damping Capacity Demand in Seismic Base Isolators via Machine Learning." Computer Modeling in Engineering & Sciences 138 (3): 2899–2924. https://doi.org/10.32604/cmes.2023.030418.

[44] Karimpouli, Sadegh, Danu Caus, Harsh Grover, Patricia Martínez-Garzón, Marco Bohnhoff, Gregory C. Beroza, Georg Dresen, Thomas Goebel, Tobias Weigel, and Grzegorz Kwiatek. 2023. "Explainable Machine Learning for Labquake Prediction Using Catalog-driven Features." Earth and Planetary Science Letters 622 (October): 118383. https://doi.org/10.1016/j.epsl.2023.118383.

[45] Brykov, Michail Nikolaevich, Ivan Petryshynets, Catalin Iulian Pruncu, Vasily Georgievich Efremenko, Danil Yurievich Pimenov, Khaled Giasin, Serhii Anatolievich Sylenko, and Szymon Wojciechowski. 2020. "Machine Learning Modelling and Feature Engineering in Seismology Experiment." Sensors 20 (15): 4228. https://doi.org/10.3390/s20154228.

[46] Jasperson, Hope, David C. Bolton, Paul Johnson, Robert Guyer, Chris Marone, and Maarten V. De Hoop. 2021. "Attention Network Forecasts Time-to-Failure in Laboratory Shear Experiments." Journal of Geophysical Research Solid Earth 126 (11). https://doi.org/10.1029/2021jb022195.

[47] Zhang, Xitong, Will Reichard-Flynn, Miao Zhang, Matthew Hirn, and Youzuo Lin. 2022. "Spatiotemporal Graph Convolutional Networks for Earthquake Source Characterization." Journal of Geophysical Research Solid Earth 127 (11). https://doi.org/10.1029/2022jb024401.

[48] Bannigan, Pauric, Zeqing Bao, Riley J. Hickman, Matteo Aldeghi, Florian Häse, Alán Aspuru-Guzik, and Christine Allen. 2023. "Machine Learning Models to Accelerate the Design of Polymeric Long-acting Injectables." Nature Communications 14 (1). https://doi.org/10.1038/s41467-022-35343-w.

Improved Memory Efficient Computing Unit DWT Architecture For Satellite Images

A. Azhagu Jaisudhan Pazhani, P. Gunasekaran and A. Rameshbabu
Department of ECE, Ramco Institute of Technology, Rajapalayam, India
E-mail: alagujaisudhan@gmail.com, mailtogunasekar@gmail.com, rameshbabu@ritrjpm.ac.in

Keywords: VLSI architecture, look-up table, distributed arithmetic, DWT, 2D, ROM, memory

Received: February 21, 2024

The 2D Discrete Wavelet Transform is a signal transform that is frequently used in picture and video compression. It is a computationally costly signal transform. VLSI implementation of 2D DWT is susceptible to a set of restrictions such as area and power consumption due to its increasing use in high data rate communication and storage in portable and handheld devices. The Distributed Arithmetic architecture is one of several architectures for constraint-driven VLSI implementation of 2D DWT that have been developed in recent years. The Distributed Arithmetic architecture is used efficiently to execute inner product computations, eliminating the need for multiplication and increasing computation speed. Filtering is the most power-intensive process in DWT, and multipliers are more expensive, so in Distributed Arithmetic architecture, multipliers are substituted with shifts and ROM lookup tables. However, as the number of filter coefficients grows, the size of the ROM look-up table grows, which can be decreased using the lookup table compression technique. In this paper, an Improved Memory Efficient Distributed Arithmetic Architecture for DWT has been proposed. The look-up table is used to stock the inner product values and then compressed. The performance of the improved LUT compressed algorithm is superior than the existing technique.

Povzetek: Predlagana je optimizirana pomnilniško učinkovita VLSI arhitektura za 2D DWT pri obdelavi satelitskih slik. Z uporabo porazdeljene aritmetike in stiskanja LUT zmanjša stroške računanja, izboljša hitrost in učinkovitost za aplikacije z visoko hitrostjo prenosa podatkov.

1 Introduction

Wavelet-based approaches are used to tackle complicated problems in math and engineering, with current applications including data compression, signal processing, image processing, pattern recognition, computer graphics, aeroplane and submarine detection, and other medical imaging technologies. A wavelet is an orthogonal function that may be applied to a limited set of data in the sense of the Discrete Wavelet Transform (DWT).

Mohanty B.K. Meher P.K. introduced a distributed arithmetic (DA) formulation for DWT computation utilising 9/7 filters in 2009, and transferred it to bit-parallel and bit-serial architectures for high-throughput and low-hardware implementations, respectively. For low-hardware solutions, the bit-serial structure processes the input vector's bit-slices in a serial fashion, whereas the bit-parallel structure processes all the bit-slices in parallel for high-throughput computing. The hardware usage efficiency of the bit-parallel structure is 100 percent. The suggested DA DWT structure has a much greater throughput rate and requires less area-delay product than conventional multiplier-less arrangements.

To process N-bit input operands, the fundamental serial architecture needs N clock cycles [3]. The primary disadvantage of the serial DA design is that it consumes more clock cycles and the filter's performance is slow. To expedite the procedure, it is preferable to apply the DA in parallel. The input data is separated into even and odd samples based on their location in the parallel implementation. Even samples convolve with even and odd filter coefficients, while odd samples convolve with the same set of coefficients at the same time [2]. The result is achieved concurrently for both even and odd input samples. The number of clock cycles is lowered, resulting in faster processing and less memory.

Distributed arithmetic calculations are bit-serial in nature in their most evident and direct form, i.e., each bit of the input samples must be indexed before a new output sample becomes available. When the input samples are represented with B bits of accuracy, an inner-product computation takes B clock cycles to complete. By replicating the LUT and adder tree, a parallel realisation of distributed arithmetic allows multiple bits to be processed in one clock cycle. The odd bits are sent to one LUT and adder tree in a 2-bit parallel implementation, while the even bits are fed to an identical tree. To suitably weight the outcome, the bit partials are left shifted and added to the even partials before aggregating the aggregate. All input bits can be calculated in parallel and then concatenated in a shifting adder tree in the extreme scenario [4].

An LUT, a cascade of shift registers, and a scaling accumulator make up the distributed arithmetic implementation of the Daubechies 8-tap wavelet FIR filter. All potential sums of the Daubechies 8-tap wavelet coefficients are stored in the LUT. The bit-wide output is delivered to the bit serial shift register cascade, one bit at a time, as the input sample is serialised. The input sample is stored in a bit-serial format in the cascade, which is then utilised to generate the requisite inner-product computation. The shift register cascade's bit outputs are utilised as address inputs to the LUT. The scaling accumulator adds together partial LUT results to generate a final result at the filter output port.

The benefit of utilising DA for a wavelet with a greater number of coefficients, on the other hand, may be lost over time due to a huge rise in memory size. The needed number of table entries is 2n. As the number of filter coefficients 'n' rises, the size of the look-up database grows exponentially.

A recent 2D DWT implementation on the NVidia GeForce GTX TITAN Black GPU was proposed in [7]. The authors of the paper [7] used a register-based technique to propose their DWT algorithm, which they claimed was four times quicker than existing GPU-based software implementations of DWT.

Darji et al. [8] presented a lifting DWT-based multiplier-less 1D/2D DWT architecture. They employed an innovative z-scanning method to reduce the transposing buffer size to 0 by using an innovative z-scanning method. Their temporal buffer size, on the other hand, is proportional to the number of input data points. Their requirement for adders is likewise quite great. Other newer methods may be able to outperform their architecture in terms of real-time image decomposition. 9/7 and 5/3 filter architectures were proposed by Meher et al. [9]. They offered 9/7 and 5/3 architectures with and without pipelines, as well as reconfigurable 9/7 and 5/3 systems. They concentrated on drastically lowering the size of the area and memory. Despite the fact that their design is space-efficient and their working speed is sufficient, there is still room to reduce their CP and thus increase the maximum operating frequency, which is a critical design component for real-time signal processing.

A multiplier-less lifting-based 2D DWT architecture was proposed in the work [10]. A flipping-based 2D DWT architecture was also presented in the same paper [10]. The inherent low critical-path delay of flipping-based architecture might be realised utilising lifting-based DWT design, according to the paper [10]. To validate the contributions, both designs were compared to other existing works. Despite the fact that the designs provided in [10] claim to greatly minimise critical-path delays, the critical-path delays of both lifting- and flipping-based architectures are significantly higher than any convolutional DWT architecture. As a result, there is plenty of room for improving timing performance.

In the work of Hegde et al. [11], the authors proposed one lifting- and flipping-based DWT architecture which is memory and power efficient. They used area consumption, critical-path delay, and power consumption as the main performance metrics. They proposed 'look-up table' (LUT)-based multiplier to reduce area and critical-path delay. They developed the architecture using gate-level HDL language and provided the ASIC implementation details. By proposing LUT-based multiplier, they successfully achieved to reduce the critical-path delay and area consumption of their multiplier than any conventional popular multiplier. However, they did not completely omit multipliers from their designs. Therefore, their design's critical-path delay and power consumption are greater than any other multiplierless design. Moreover, LUT-based design uses a lot of registers or memory. Therefore, their design is also memory extensive.

We are now concentrating on briefly mentioning some of the most current works in the domain of DWT architectural design, having discussed some of the most recent and benchmark works in the subject. The authors introduced 1D/2D DWT architectures based on floating-point multiply and accumulator circuit' (MAC) units in their paper [12]. The 45 nm CMOS technology was used to implement the design. Though the validation and verification of the work is commendable, the performance in terms of critical-path delay, CT, and memory consumption should be improved further.

The study given in [13] is about the LeGall 5/3 DWT filter's DA-based DWT architecture. The work was implemented on an Altera FPGA, and the design's quality was compared to that of previous DWT-based works to demonstrate its superiority. However, there is still a lot of room for improvement in terms of area usage, power consumption, and operation speed with the DWT architecture. The authors of the paper [14] described a LeGall 5/3 DWT filter with a 1D DWT architecture based on 'canonical sign digit' (CSD)-based DA.

The authors used the CSD-based DA approach to propose a hardware-efficient DWT architecture that only required seven adders, a few shift registers, and multiplexers. However, their clock period is 100 ns [14]. This means that the working frequency of their design is only 10 MHz, which is far too low for many real-time applications. The work of [15] offered another major and current DWT architecture. A dual-memory controller-based 2D DWT architecture with a focus on real-time image processing was presented in the study [15]. The design's memory requirements were said to be streamlined to allow for real-time image processing.

An architecture that reduces the number of adders in a 1D Daub-4 filter module architecture and enhances the conventional Daub-4 very large-scale integration (VLSI) architecture design was proposed by Tiancai Lan et al [16]. The input image has a size of N × N matrix, and the output result is saved in the TM. Four sub-bands are obtained by reading the high and low frequencies one at a time to the second Daub-4 filter following the first Daub-4 filter's process.

Hussin et al. [17] proposed the 2D DWT and Huffman encoding for image compression. Once the input image has been chosen, the first step begins with RGB layer division. Next, superfluous image data at each RGB layer is eliminated using the lossy compression (DWT) technique. The output of the DWT process is then encoded

and stored using lossless compression (the Huffman encoding approach).

The major purpose of this study is to create a DWT with a memory-efficient multiplier-less architecture. In DWT filtering, the distributed arithmetic architecture is used to produce multiplier-less computing. The size of the ROM look-up table increases when the filter coefficients rise in DWT with DA architecture, which can be lowered by employing a more effective LUT compression mechanism.

The size of the LUT can be lowered by counting the number of toggles between each pair of entries and compressing the result. The idea behind compressing the table is to reduce the amount of bit transitions per column as much as possible, then save the indices just where a bit toggling occurs rather than the entire column. Using the look-up table decoding approach, the needed inner product value is created from the compressed look-up table.

The following is a breakdown of the paper's structure. The DA architecture for DWT implementation was covered in part II. The suggested DA-based DWT architecture with better compression algorithm is described in Section III. In section IV, the findings and debates are discussed. Section V brings the paper to a close.

2 Distributed arithmetic architecrure for dwt implementation

FPGA implementation may be difficult due to their lack of arithmetic capabilities compared to general-purpose DSP processors. The reprogrammable configuration of FPGA is, nevertheless, its most significant benefit. Field Programmable Gate Arrays (FPGAs) are utilized in this study to implement DWT in hardware. With a large reduction in calculation time, DWT gives enough information for analysis and synthesis of the original signal.

The DA-based DWT has several uses in science, engineering, mathematics, and computer science. The use of DWT as an analogue filter bank in biomedical signal processing for the creation of low-power pacemakers, as well as in ultra-wideband wireless communications, is demonstrated.

To disguise the multiplications, DA is a bit level rearrangement of a multiply accumulate. It's a useful strategy for shrinking parallel hardware multiply accumulates that's ideally suited to FPGA designs. Since its introduction over two decades ago, DA has been frequently employed in VLSI implementations of DSP systems. The majority of these applications rely heavily on computing, with multiplication and/or addition being the most common operations. The key benefit of the distributed arithmetic technique is that it speeds up the multiply process by computing and storing all potential medium values in a ROM. After that, the input data may be used to address the memory and the result directly.

Formulation of algorithm

An illustration of normal Multiply Accumulate (MAC) operation

$$y = A_1X_1 + A_2X_2 + ...A_iX_i \quad (1)$$

A_i = Coefficient, X_i = Input

Distributed arithmetic implementation of DWT

Let Xk be a N-bits scrambled 2's complement number $|X_k|<1$

X_k: $\{b_{k0}, b_{k1}, b_{k2}\ldots\ldots, b_{k(N-1)}\}$,
Where b_{k0} is the sign bit

X_k is expressed as

$$X_k = -b_{k0} + \sum_n^N \quad (2)$$

Substitute equation (2) in equation (1),

$$y = \sum_{k=1}^{k} A_k + \sum_n^N$$

$$y = \sum_{k=1}^{k} b_{k0}A_k + \sum_{k=1}^{k} A_k \sum_{n=1}^{N-1}(A_k b_{kn})\, 2^{-n}$$

$$y = -\sum_{k=1}^{k} b_{k0}A_k + \sum_{k=1}^{k}\sum_{n=1}^{N-1}(A_k b_{kn})2^{-n} \quad (3)$$

Expanding this part

$$y = -\sum_{k-1}^{k} b_{k0}A_k + \sum_{k=1}^{k}(A_k b_{k1})2^{-1} + (A_k b_{k2})2^{-2} + \cdots + \left(A_k b_{(N-1)}\right)2^{-(N-1)} \quad (4)$$

$$y = -[b_{10}A_1 + b_{20}A_2 + \cdots + b_{k0}A_k] + \left[(b_{11}A_1)2^{-1} + (b_{12}A_1)2^{-2} + \cdots + b_{1(N-1)}A_1 2^{-(N-1)}\right] + \cdots + [(b_{k1}A_k)2^{-k} + (b_{k2}A_k)2^{-k} + \cdots (b_{k(N-1)}A_k)2^{-(N-1)}]$$

$$y = -\sum_{k=1}^{k} b_{k0}A_k + \sum_{n=1}^{N-1}[b_{1n}A_1 + b_{2n}A_2 + \ldots + b_{kn}A_k]\, 2^{-n} \quad (5)$$

$$y = -\sum_{k=1}^{k} A_k(b_{k0}) + \sum_{n=1}^{N-1}[\sum_{k=1}^{k-1} A_k b_{kn}]\, 2^{-n} \quad (6)$$

Because each b_{kn} can only take on values of 0 and 1, there are only 2k potential possibilities. The memory holds the result y after N such cycles.

Hardware reduction in DA method

Figure 2.1 gives the hardware realization of the original equation (3) and for this original equation, the hardware utilization is high. The DA approach decreases hardware use, allowing the operation to run faster.

$$y = -\sum_{k=1}^{k} A_k(b_{k0}) + \sum_{k=1}^{k}\sum_{n=1}^{N-1}(A_k b_{kn})\, 2^{-n} \quad (7)$$

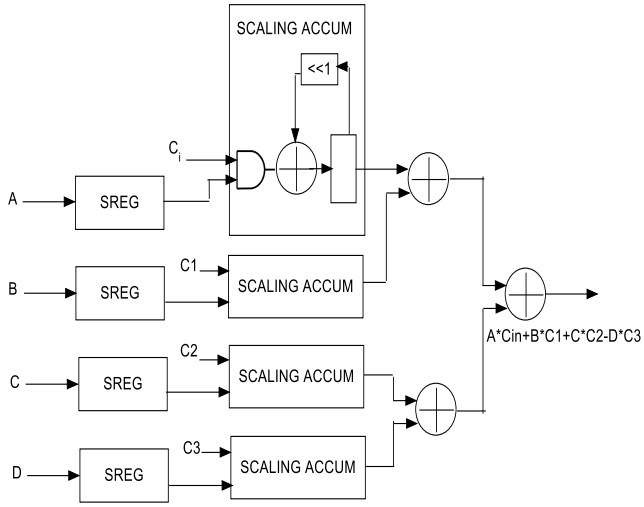


Figure 2.1: Hardware utilization for original equation

Figure 2.2 shows the hardware utilization in bit level rearrangement. In that hardware is reduced compared to original equation

$$y = -\sum_{k=1}^{k} A_k(b_{k0}) + \sum_{n=1}^{N-1}[\sum_{k=1}^{k-1} A_k b_{kn}]\; 2^{-n} \quad (8)$$

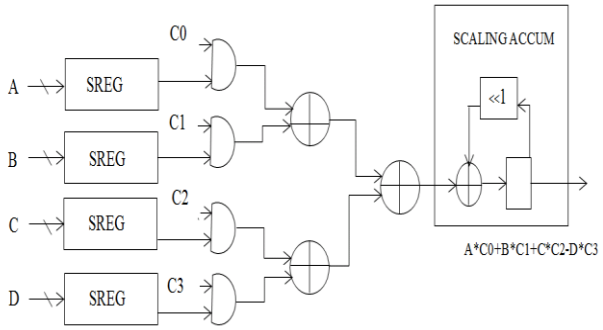


Figure 2.2: Hardware utilization in bit level rearrangement

DA architecture

The LUT, Shift registers, and scaling accumulator make up the DA architecture of a FIR filter. Various sums of the four coefficients make up the LUT data. The operands are loaded into the registers through a register chain in the shift registers. Depending on whether a serial or parallel architecture is used, the operands are then shifted 'n' bits at a time. In the scaling accumulator, the output of the DA LUT is added to the scaled output. It's made with an M-bit adder and a N+M-bit shift register at the output.

Serial DA architecture

As illustrated in Figure 2.3, the basic serial architecture requires N clock cycles to handle N-bit input operands. The LUT, adder tree, and scaling accumulator are all part of the critical path in the DA architecture, which runs from the input shift register to the output. The critical path delay is dominated by adder delays without the pipeline registers. When the design is fully pipelined, the significant fan-out loading delay incurred at the output of the shift register feeding the DA LUT inputs entirely masks the adder delays. If the loading factor is taken into account, the adder delays dominate the critical route latency, which may be considerably reduced by applying the technique outlined in. However, there will be little benefit from adopting quicker adder stages until the fan-out delays are addressed.

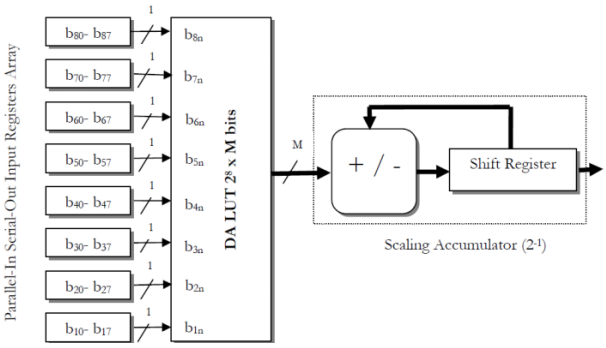


Figure 2.3: Serial DA architecture

The implementation findings show that by using parallelism with more than one bit at a time, the performance of DA systems may go up virtually linearly. Adding parallelism is the same as repeating the fundamental structure as many times as needed, each of which may function independently without clock frequency deterioration caused by pipelining.

Due to pipelining, the frequency of both operations stays the same. Furthermore, because each stage of the DA calculation is only a single basic FPGA element, the highest potential clock frequency for a particular FPGA device may be exploited. The main drawback of the serial DA architecture is, it requires more clock cycles and the speed of filter is low.

Parallel DA architecture

The procedure will be slower because the DA architecture is bit serial in nature. A parallel distributed arithmetic architecture is built to speed up the procedure [4]. Figure 2.4 depicts the parallel DA architecture. The input data is separated into even and odd samples based on their location in parallel implementation. Filter coefficients are also divided into even and odd samples. Even samples convolve with even and odd filter coefficients, while odd samples convolve with the same coefficients at the same time.

It is possible to receive results for both even and odd samples of input at the same time. The number of clock cycles is lowered, resulting in faster processing and less memory. The registers are loaded with the input

values for each cycle, and then the reloading procedure to registers is enabled for the following set of cycles. The serial shift register, which must access the look-up table, will receive the input x[n]. The old value will be moved into the next register when the new input arrives in the first register. Similarly, as new values enter registers, the old values are removed from the registers.

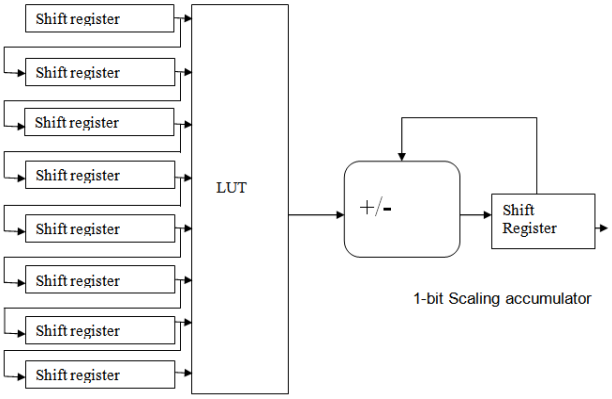


Figure 2.4: Parallel DA architecture

Consider the bit locations and retrieve the values of inputs from that bit position to get the address from the input values. Consider the LSBs of all serial registers to determine the initial address, for example. The initial position value will be generated using this address. Obtain all of the bit position addresses and the accompanying values from the look-up table in the same manner. Shift the values by the bit position value and provide them to the adder during adding. Finally, the output, which is the convolution of the filter coefficients and the inputs, will be generated.

Both the high-pass and low-pass filters will be built using the same design. If the input is 8 bits long, the convolved value takes 8 clock cycles to compute. The filter operations are stated using floating point arithmetic while computing the wavelet coefficients. In practice, though, integer arithmetic is employed. The filter coefficients are shortened as a result. The precision of the calculated coefficients suffers as a result of this reduction.

3 Proposed memory efficient da architecture for dwt

Implementing DWT with DA architecture may improve computation speed, but it will also increase memory size as the number of wavelet coefficients grows. The multi-level decomposition requires a high level of DWT implementation complexity. As a result, the benefit of employing DA will be effectively gone. The size of the look-up table in the DA architecture for DWT is reduced using a novel way. A table compression approach, as shown in Figure 3.1, can be used to minimize the size of the look up table required to record all possible combinations of input in DA architecture. The algorithm for compressing the LUT is the same as that used to save a processor's assembly language instructions [5]. A similar approach can be used to reduce the number of LUTs in DA architecture [1].

After going through high pass and low pass filters, the DWT coefficients are created. The filter coefficients are convolved with inputs to perform the filter operation with N input variables. The coefficients are fixed in this case. Binary can be used to represent inputs. The inputs are scaled to have absolute values less than one. In ROM look-up tables, the inner product for several inputs can be computed and saved in advance. If there are n wavelet coefficients, the look-up table will be 2n. All LSBs are assumed to be the first to receive data. Similarly, all bit positions are determined, and the look-up database is used to determine the appropriate values.

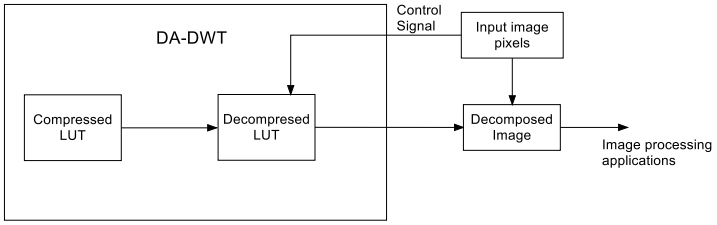


Figure 3.1: Memory reduced DA architecture

LUT encoding algorithm

The size of the LUT can be lowered by counting the number of toggles between each entry and compressing them [1]. The idea behind compressing the table is to reduce the amount of bit transitions per column as much as possible, then save the indices just where a bit toggling occurs rather than the entire column. Figure 2 displays an example of a LUT with seven symbols, each with eight bits. The table is 56 bits in size (before compression). There are 8 distinct binary words in the table, with an index length of 3 bits. As a result, if the column contains no more than two transitions, it can be compressed. Seven columns will be compressed in this example, but one column will remain uncompressed. After compression, the table's size is reduced to 34 bits (from 56 bits before). FPGA RAM blocks are used to hold the compressed table.

If the lookup table compression is modified using the following steps auxiliary compression can be achieved. The steps to be incorporated in the modified lookup table compression are as follows:

Total number of locations: LUT size: 2^n

if index< 2n/2
use rep with (n-1)-bits
else
n-bits

Using the above steps the table is further compressed as shown in Figure 3.2. Hence the LUT compression of 28 bits can be achieved.

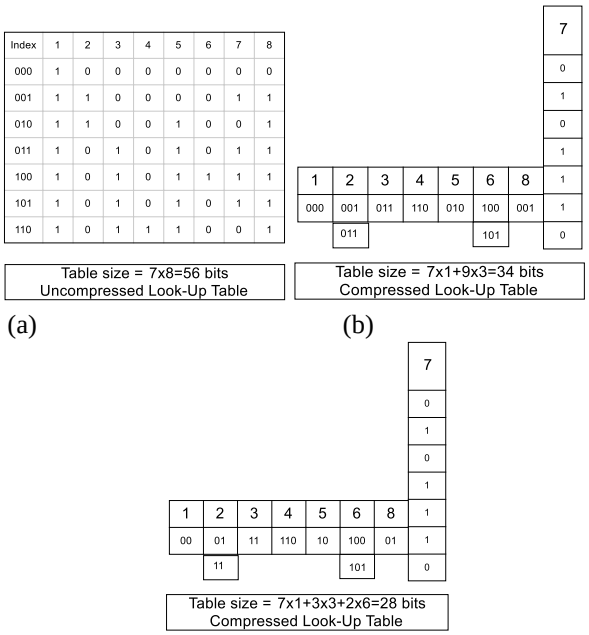


Figure 3.2: a) Uncompressed LUT b) Existing Compressed LUT c) Improved LUT Compression

Using the LUT compression methodology and the improved LUT compression, the size of the compressed LUT is decreased by 39.28% and 50%, respectively. Thus the modified LUT can be an efficient method for compressing the DWT coefficients.

LUT decoding algorithm

The needed inner product value is created from the compressed look-up table in this decoding process. When a certain input to a look-up table comes, it determines its location in each compressed table column.

- If the input is greater or equal to the compressed look-up table value, then generate '1'
- If the input is lesser to the compressed look-up table value, then generate '0'

The uncompressed table columns' original bits are received straight from the ROM.

DA DWT architecture

The parallel implementation of DA architecture is exposed in Figure 3.3. The input data is separated into even and odd samples based on their location in parallel implementation. As a result, even samples convolve with even and odd filter coefficients, whereas odd samples convolve with the same set of coefficients. The results for both even and odd samples of input are obtained. Here number of clock cycles are abridged which results in increased speed and decreased memory.

To access the LUT, the same number of registers must be used for accessing filter quantities. The data will be sent into a serial shift register, which will need to consult the look-up table. The old value will be moved into the next register when the new input arrives in the first register.

Similarly, when new values enter registers, the old values are removed from the registers by examining bit positions and determining the values of inputs based on that bit position. Finally, all bit position addresses are obtained from the look-up table and are given as input to adder by shifting its values. Finally, the result, which is the convolution of the filter coefficients and the inputs, will be achieved.

The DA architecture speeds up the operation by lowering memory use, but as the size of the look-up table grows larger, the decoding process becomes more time demanding.

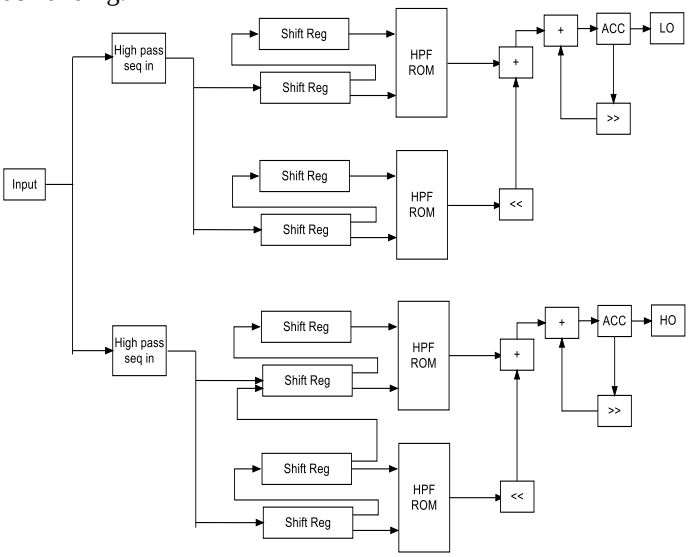


Figure 3.3: DA DWT architecture

4 Results and discussion

In this work, the distributed arithmetic architecture for DWT is designed and simulated using Verilog in MODEL SIM 0.61xd. Simulation verifies the functionality of both high pass and low pass filters. Then it is synthesized into Spartan3E FPGA platform using Xilinx ISE Design Suite 13.2.

Simulation and synthesized results for single level DWT

The synthesized results for the suggested design are presented in Figure 4.1 for the low pass and high pass filters. The Parallel DA-DWT Architecture reads input vectors from a ROM. The shredded outputs are saved, and simulated waveforms are used to illustrate them.

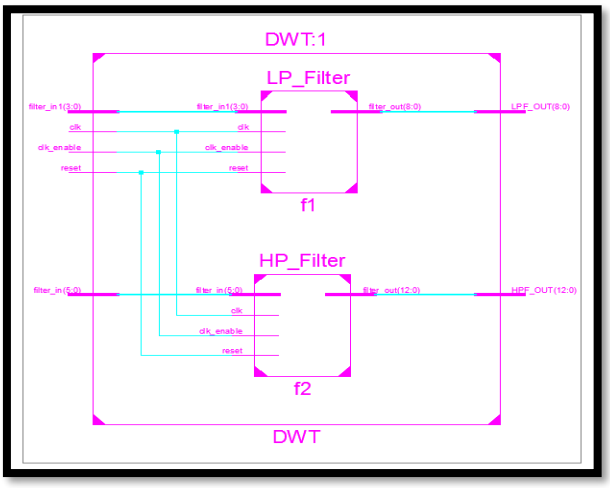


Figure 4.1: Synthesized result of single level DWT

Comparison of uncompressed and compressed DA ROM Size

The Table I give the memory size of the look-up table for low pass and high pass filter with uncompressed DA is reduced to 60% and 40% compared to compressed DA respectively. The proposed technique gives the compression efficiency of 50% for low pass and 72% for high pass filter whereas the existing technique gives the compression efficiency of 60% for low pass and 63% for high pass filter.

Table 1: Comparison of distributed arithmetic schemes

Architecture	**Memory size (ROM) Lowpass filter**	**Memory size (ROM) Highpass filter**
Uncompressed DA [1]	80 bits	256 bits
Existing Compressed DA [8]	48 bits	96 bits
Proposed Improved DA	40 bits	72 bits

Performance comparision

The performance comparison of different architecture for DWT is given in Table II.

Table 2: Performance comparison of various DWT architecture

Scheme	**Level = 1**
Filter implementation [9]	16 multipliers
Lifting implementation [8]	6 multipliers
Serial DA based implementation [14]	43 adders
Compressed DA based implementation	4 adders 4 subtractors

The Table II gives the requirement of adder and multiplier for different architectures to design DWT. The filter based implementation involves direct multiplication for inner product calculation in the filter, which requires more number of multipliers. The filter based implementation of DWT for single level requires 16 multipliers. The lifting scheme is implemented to reduce the arithmetic computation which requires 6 multipliers to implement the DWT for single level. The serial DA based architecture involves multiplier less operation for inner product calculation; it requires 43 adders to design single level DWT. The proposed method reduces up to 4 adders and 4 subtractors.

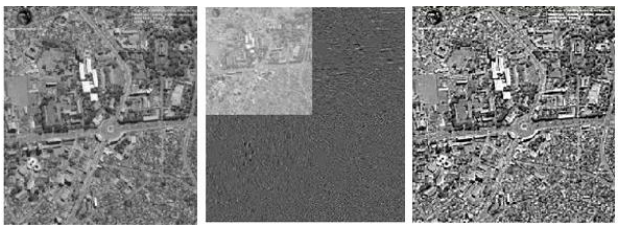
Hardware utilization comparision

The Table III gives the device utilization of DA architecture. It is less compared to convolution based architecture. The DA architecture uses LUT instead of multiplier for MAC unit to get inner product calculations.

Table 3: Hardware utilization comparison

LOGIC UTILIZATION	**CONVOLUATION BASED ARCHITECTURE (one level) [6]**	**DA BASED ARCHITECTURE (one level)**
Number of slices Flip Flops	47	102
Number of 4 input LUTs	294	115
Total number of occupied slices	209	91
Number of bonded IOBs	91	35
Number of BUFGMUXs	1	1

Images transform comparisons using 2D-DWT



(a) Input image (b) DWT Processed image (c) Output image

5 Conclusion

The memory efficient DA architecture for discrete wavelet transform is implemented using Spartan 3E FPGA. The DA architecture is built on the Look-up table technique for effective inner product computation. When using DA architecture to implement DWT, the size of the ROM look-up table grows as the filter coefficients grow. The revised look-up table compression technique reduces the size of the LUT up to 115. The compressed LUT is kept in the FPGA's ROM. Data can be decrypted by decompressing the table while conducting DWT calculation. The memory-based method enables the Parallel DA-DWT to achieve high computation speeds

while using a little silicon area by replacing multipliers with compact ROM tables. Saving adders, quick processing time, regular flow of data, and minimal control complexity are all advantages of the suggested architecture, making it suited for image compression systems. The proposed method reduces the memory size from 80 bits to 40 bits for LPF and 256 bits to 72 bits for HPF, but the decoding process will be time consuming while increasing the filter coefficients. The focus of future research will be on improving the speed of retrieval from LUTs and quick decoding.

Author contributions statement

"All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by [A. Azhagu Jaisudhan Pazhani], [P. Gunasekaran] and [A. Rameshbabu]. The first draft of the manuscript was written by [A. Azhagu Jaisudhan Pazhani]. All authors read and approved the final manuscript."

Conflict of interest

There is no conflict of interest in this paper regarding publication.

Data availability statement

The data that supports the findings of this study are available within the article.

Funding

No funding was received for this study.

References

[1] Remya Ajai A S, Nithin Nagaraj (2012), "A Novel methodology For Memory Reduction in Distributed Arithmetic Based DWT" International Conference on Communication Technology and System Design procedia Engineering 30, pp. 226-233.

[2] K.B. Sowmya, Dr. SavitaSonoli and M. Nagabushanam (2012), "Implementation of Parallel DA Technique for DWT-IDWT on FPGA for Image Compression", International Journal of Power Systems and Integrated Circuits, Vol. 2, pp 143 – 148.

[3] Mohanty B.K. Meher P.K (2009), "Efficient multiplier less designs for 1-D DWT using 9/7 filters based on distributed arithmetic", Dept. of Electronics and Communication Engineering., Jaypee Inst. of Eng. & Technol., Guna district, India, vol 1.1, no 6, pp 364 – 367.

[4] Al-Haj AM (2005), "An FPGA-Based Parallel Distributed Arithmetic Implementation of the 1-D Discrete Wavelet Transform," Informatica vol. 29, no 2, pp 241-247.

[5] Xixin Cao, QingqingXie, ChunganPeng, Qingchun Wang (1996), "An Efficient VLSI Implementation of Distributed Architecture for DWT" IEEE Transaction VLSI System, vol. 2, no 6, pp. 521-543.

[6] Basant kumar mohanty, Pramod kumar (2013), "Memory-Efficient High-Speed Convolution-Based Generic Structure for Multilevel 2-D DWT" IEEE Transaction on circuits and systems for video technology, vol. 23, No. 2.

[7] Enfedaque, P, Auli-Llinas F, Moure J.C (2014), "Implementation of the DWT in a GPU through a register-based strategy" IEEE Trans. Parallel Distrib. Syst. 26(12), 3394–3406.

[8] Darji A, Arun R, Merchant S.N, Chandorkar A (2015), "Multiplierless pipeline architecture for lifting-based two-dimensional discrete wavelet transform" IET Comput. Dig. Tech. 9(2), 113–123.

[9] Meher P.K, Mohanty B.K., Swamy M.M.S (2015), "Low-area and low-power reconfigurable architecture for convolution-based 1-D DWT using 9/7 and 5/3 filters" 28th International Conference on VLSI Design, Bangalore, pp. 327–332.

[10] Mohanty B.., Meher P.K, Srikanthan T (2015), "Critical-path optimization for efficient hardware realization of lifting and flipping DWTs" IEEE International Symposium on Circuits and Systems (ISCAS), Lisbon, pp. 1186–1189.

[11] Hegde G, Reddy K.S, Ramesh, T.K.S (2018), "A new approach for 1-D and 2-D DWT architectures using LUT based lifting and flipping cell", AEU Int. J. Electron. Commun. 97, 165–177.

[12] Mohamed Asan Basiri M, Noor Mahammad S (2018), "An efficient VLSI architecture for convolution-based DWT using MAC", 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems, Pune, pp. 271–276.

[13] Aziz, F, Javed S, Iftikhar Gardezi S.E, Jabbar Younis C, Alam M (2018), "Design and implementation of efficient DA architecture for LeGall 5/3 DWT", International Symposium on Recent Advances in Electrical Engineering (RAEE), Islamabad, pp. 1–5.

[14] Gardezi S.E.I, Aziz F, Javed S, Younis C.J, Alam M, Massoud Y (2019), "Design and VLSI implementation of CSD based DA architecture for 5/3 DWT", 16th IEEE International Bhurban Conference on Applied Sciences and Technology (IBCAST), pp.548–552.

[15] Naik P, Guhilot H, Tigadi A, Ganesh P (2019), "Reconfigured VLSI architecture for discrete wavelet transform", Soft Computing and Signal Processing. Springer, Singapore, pp. 709–720.

[16] Tiancai Lan, Chih-Hsien Hsia, Po-Ting Lai, Hsien-Wei Tseng and Cheng-Fu Yang (2022), "Memory efficient Very Large-Scale Integration Architecture of 2D Algebraic-integer-based Daubechies Discrete Wavelet
Transform", Sensors and Materials, Vol. 34, No. 9 3623–3636.

[17] M.A. Hussin, F.A. Poad, A. Joret (2021), "A comparative study on the performance of DWT and huffman compression technique on a 2D signal", J. Electron. Voltage Appl. 2 (1) 11–19.

Research on Sign Language Recognition for Hearing-Impaired People Through the Improved YOLOv5 Algorithm Combining CBAM with Focal CioU

Niqin Jing*, Yi Hu, Yanxia Wang
[1]Beijing Polytechnic, Beijing 100176, China
E-mail: jingnqniq@hotmail.com
* Corresponding author

Keywords: deep learning, hearing-impaired people, sign language recognition, YOLOv5

Received: November 14, 2024

Sign language recognition has become increasingly important as the number of hearing-impaired people increases. This paper optimized the you only look once version 5 (YOLOv5) algorithm from perspectives of attention mechanism and loss function. The convolutional block attention module (CBAM) was added to the network, and the original intersection over union (IoU) loss function was improved to focal complete IoU (CIoU). Experimental analyses were performed on the American Sign Language (ASL) dataset in the Windows 10 environment. Moreover, the ten-fold cross-validation was used. The experiments found that adding the CBAM to the neck part of YOLOv5 showed the most effective sign language recognition results. The improved algorithm showed improvements of 0.95% in P value, 4.19% in R value, and 2.66% in mean average precision (mAP) compared to the baseline algorithm. When comparing different loss functions, the focal CIoU performed the best. Compared with other recognition algorithms, the improved YOLOv5 algorithm performed better in sign language recognition, achieving P value, R value, and mAP of 93.26%, 96.77%, and 98.12%, respectively. These results verify the reliability of the improved YOLOv5 algorithm in sign language recognition for hearing-impaired people. It can be applied in practice.

Povzetek: Članek raziskuje prepoznavanje znakovnega jezika za naglušne osebe z izboljšanim algoritmom YOLOv5, ki združuje CBAM z Focal CioU. Avtorji so optimizirali algoritem YOLOv5 z dodajanjem pozornostnega mehanizma CBAM in izboljšanjem funkcije izgube IoU na Focal CIoU.

1 Introduction

Sign language is a main communication tool for hearing-impaired people [1]. The study of sign language has gained more attention as the number of people with hearing impairments continues to increase. Sign language, a type of body language, conveys complex meanings through gestures, which can be understood after specialized learning. However, the general population has limited exposure to sign language, posing significant challenges for hearing-impaired individuals in communicating with the outside world. With the continuous advancement of computer technology, using computers to achieve sign language recognition can provide reliable assistance for communication among the hearing-impaired population [2]. Sign language recognition can be categorized into the recognition of static sign language images and the recognition of dynamic sign language videos, which have been extensively investigated [3]. FAl Rafi et al. [4] studied the identification of Bengali sign language using pre-trained MobileNetV2 and a conditional deep convolutional generative adversarial network, achieving a test accuracy of 94.74%. Takahashi et al. [5] proposed a network that combined a 3D convolutional neural network (CNN) with a Transformer for isolated sign language identification. They demonstrated its effectiveness through experiments on LSA64. Yu et al. [6] explored Chinese sign language identification based on wearable sensors and used a deep belief network to recognize captured electromyography, accelerometer, and gyroscope signals, achieving favorable recognition accuracy. Joshi et al. [7] studied dynamic Gujarati sign language recognition. They extracted features based on the Mediapipe algorithm, established a deep learning model with six layers based on long short-term memory, and found high accuracy through experiments. Wang et al. [8] developed a gesture recognition method based on the Transformer model and trained it on a large corpus. Through experiments, it was found that the average word error rate of this method was 21.6%. Sharma et al. [9] proposed an attention-based real-time embedded long short-term memory (LSTM) for dynamic sign language identification and achieved a real-time recognition rate of 99.7%. Kourbane et al. [10] put forward a new deep learning-based framework to achieve hand pose estimation. Through extensive experiments on two datasets, they found that this method was superior to the existing methods. This paper primarily focused on the recognition of static sign language images. To address challenges such as feature extraction difficulties and poor recognition performance of sign language images and to further enhance the recognition performance of sign language images, an optimized you only look once version 5 (YOLOv5) model was developed based on deep learning. The effectiveness of this model in sign language recognition was verified through experiments, offering a more accurate approach for recognizing static sign language images. Moreover, the method enhanced

communication efficiency between hearing-impaired people and the outside world. The results also provide theoretical support for further utilization of deep learning methods.

2 Related works

The improved YOLOv5 algorithm developed in this paper was compared with some existing target recognition methods, and the following results were obtained.

Table 1: Comparison of related works.

	P/%	R/%	mAP@0.5/ %
Faster region-CNN [11]	80.12 ± 1.87	67.89 ± 1.65	79.84 ± 1.77
YOLOv3 [12]	87.77 ± 2.01	80.12 ± 1.77	90.31 ± 2.01
YOLOv4 [13]	88.05 ± 1.97	83.25 ± 1.56	92.56 ± 2.33
YOLOv5	88.12 ± 2.07	90.33 ± 1.64	94.21 ± 2.14
MobileNetV2 [14]	91.12 ± 2.56	81.94 ± 1.82	91.27 ± 2.05
ShuffleNetV2 [15]	91.08 ± 2.33	82.11 ± 2.01	91.26 ± 2.17
Improved YOLOv5	93.26 ± 2.77	96.77 ± 2.68	98.12 ± 2.32

The results in Table 1 verified the reliability of the improved YOLOv5 algorithm in recognition of static sign language images. Compared with the existing target detection methods, in this paper, based on the traditional model, the improvement of the detection performance was achieved through the introduction of the attention mechanism and the optimization of the loss function, enabling the model to pay more attention to the samples that are difficult to classify.

3 Improved YOLOv5 algorithm

3.1 Sign language and deep learning

Hearing impairment is a global health issue [16]. Based on the data published by the China Disabled Persons' Federation, the number of hearing-impaired people in China reached 20.54 million in 2010, accounting for the most significant proportion of disabilities (24.16%). Among them, children have a relatively high prevalence of Grade 1 and Grade 2 hearing disabilities. Moreover, at least 20,000 newborns are affected by hearing impairment annually, with a prevalence rate of 0.1%-0.3% for congenital hearing impairment in newborns and 0.27% for children under five years old.

The hearing-impaired people usually use sign language for communication. However, sign language interpreters are often necessary for effective communication between the general population and people who rely on sign language. Unfortunately, the severe shortage of such interpreters cannot meet the communication needs of these people. As technology develops, artificial intelligence-based sign language recognition has emerged as a prominent solution to address hearing-impaired people's communication requirements.

As a non-verbal communication, sign language does not rely on auditory language but utilizes a unique grammatical structure. It is the visual language for individuals with hearing impairments and plays a crucial role in communication [17]. Sign language recognition can aid hearing-impaired people in communicating with the society. It can be categorized into static and dynamic sign language recognition. The former involves identifying gestures in images and has wide applications in hospitals and banks. The latter refers to a series of movements within a short time. Hand trajectory is combined with position for accurate recognition; therefore, it is more complex than static gestures.

In recognizing static sign language images, rich gesture features are extracted from them, and a classifier is used for accurate recognition. There are two main approaches to feature extraction. The first approach involves extracting visual features from sign language images pre-processed by denoising and segmentation [18]. Sign language recognition can be achieved using methods like support vector machines (SVM) or extreme gradient boosting (XGBoost), which learn a limited number of features. The second approach is based on deep learning, which can learn advanced features from images and achieve faster training. It has shown excellent performance in tasks like image identification and target detection [19], making it increasingly popular in sign language recognition [20].

A convolutional neural network (CNN) is a basic deep learning approach [21]. Image features are extracted by convolution. The convolution operation is conducted on the input feature maps to get new feature maps. The formula for convolution operation is:

$$Y_k^m = f\left(\sum_{j\in T} W_{jk}^m * Y_j^{m-1} + b_k^m\right),$$

where T is the set of feature y_j^{m-1} in $m-1$, W_{jk}^m is the weight of the convolution kernel, b_k^m is the bias, and $*$ is the convolution operation.

The pooling layer reduces dimensionality through feature selection, which reduces the computation amount and avoids overfitting. Generally, there are two operations: maximum pooling and average pooling (Figure 1).

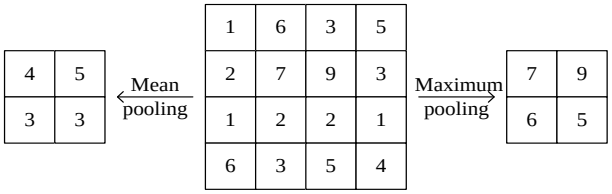


Figure 1: An example of pooling operations.

For the features that are learned by convolution and pooling, the CNN converts them into classification results in the output layer through a fully connected layer. A

Dropout layer is usually added to the network to avoid overfitting:

$$\tilde{y}^{(l)} = Bernoulli(p) \times y^{(l)},$$
$$z_i^{(l+1)} = w_i^{(l+1)}\tilde{y}^{(l)} + b_i^{(l+1)},$$
$$y_i^{(l+1)} = f\left(z_i^{(l+1)}\right),$$

where $y^{(l)}$ stands for the output vector of the l layer, $Bernoulli(p)$ is the Bernoulli function, $w_i^{(l+1)}$ and $b_i^{(l+1)}$ are the weight and bias of the $l+1$ layer, and $z_i^{(l+1)}$ is the input vector of the $l+1$ layer.

In CNN, nonlinear factors are introduced through activation functions to enhance the fitting ability of the network. Commonly used activation functions are:

(1) sigmoid function: $y = \frac{1}{1+e^{-x}}$,

(2) Tanh function: $y = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$,

(3) rectified linear unit (ReLU) function: $y = max\{0, x\} = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases}$.

3.2 YOLOv5 algorithm

Based on a CNN, the YOLO algorithm has various versions, such as YOLOv2 and YOLOv3. Among these versions, the most widely used is the YOLOv5 algorithm [22], which has a more lightweight structure and provides outstanding advantages in detection speed and accuracy. The YOLOv5 algorithm has five versions, namely n, s, m, l, and x, which differ in width and depth. The YOLOv5s algorithm is the lightest version and is particularly suitable for mobile deployment. Thus, this paper presents a sign language recognition method for hearing-impaired people based on the YOLOv5s algorithm.

The YOLOv5 network can be segmented into the following parts.

(1) Input end

Mosaic data augmentation is employed to expand the dataset and increase the diversity of the data. Moreover, the scaling of the input image is adaptively adjusted to enhance recognition accuracy and efficiency.

(2) Backbone network

① Focus module: The input image is sliced to get multiple low-resolution sub-images to reduce the amount of computation.

② Cross stage partial (CSP) network module: Convolution operation is combined with residual components to enhance the feature extraction capability of the model.

(3) Neck network

① Spatial pyramid pooling (SPP) structure: The feature maps of different sizes are divided into four blocks, which are subjected to maximum pooling of 1×1, 5×5, 9×9, and 13×13, and then the resulting feature maps are spliced and input to the next layer.

② Feature pyramid network (FPN) and path aggregation network (PAN) structures: They have multiple bottom-up and top-down paths to acquire more information.

(4) Head network

The feature maps output from the backbone and neck networks are post-processed to obtain the final recognition results. The binary cross entropy loss (BCELoss) is used as the classification loss function:

$$BCELoss = -\frac{1}{n}\sum[y_n \ln x_n + (1-y_n)\ln(1-x_n)],$$

where x_n is the first probability of the n-th sample and y_n is the binary label value (0 or 1).

The complete intersection over union (CIoU) loss is used as the bounding box loss function:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2} + \alpha v,$$
$$\alpha = \frac{v}{(1-IoU)+v},$$
$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2,$$

where IoU is the IoU between the predictive box and true box, $\rho^2(b, b^{gt})$ is the Euclidean distance between predictive box b and true box b^{gt}, c is the diagonal length of the minimum outer rectangle of the predictive box and true box, α is the weighting function, v is the width-to-height ratio similarity, w^{gt} and h^{gt} are the width and height of the predictive box, w and h are the width and height of the predictive box.

The YOLOv5 algorithm also employs non-maximum suppression (NMS) as a post-processing technique to eliminate duplicate recognition results and filter out the best detection box:

$$s_i = \begin{cases} s_i, iou(M, b_i) < N \\ 0, iou(M, b_i) \geq N \end{cases},$$

where s_i is the confidence level of the i-th predictive box, M is the current predictive box with the highest confidence level, b_i is the i-th predictive box, and N is the IoU threshold.

3.3 Improved YOLOv5 algorithm

This paper optimized the YOLOv5 algorithm in terms of both the attention mechanism and the loss function to further improve its performance in sign language recognition.

Adding the attention mechanism can make the model allocate greater focus towards essential parts and thus improve the recognition performance, which has promising applications in machine vision, natural language processing, and other fields [23]. This paper adds the convolutional block attention module (CBAM) [24] to the YOLOv5 algorithm to enhance the network's generalization ability.

The CBAM module has been well applied in image recognition tasks, such as remote sensing images [25] and radar images [26]. The structure of CBAM is presented in Figure 2.

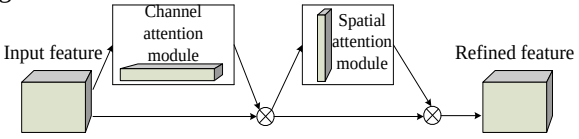


Figure 2: CBAM structure.

For feature map $F \in R^{C\times H\times W}$, C is the number of channels, and H and W are length and width. The formula for channel attention is:

$$M_C(F) = \sigma\left(W_1\left(W_2\left(F_{avg}^C\right)\right) + W_1\left(W_2\left(F_{max}^C\right)\right)\right),$$

where F_{avg}^{C} and F_{max}^{C} are feature maps after mean pooling and maximum pooling, σ is the sigmoid activation function, W_1 and W_2 are weights.

The input of spatial attention is the multiplication result of M_C and original feature map F. The calculation formula is:

$$M_S(F_S) = \sigma\left(f^{7\times7}\left(\left[F_{avg}^{S}; F_{max}^{S}\right]\right)\right),$$

$$F_S = M_C \otimes F.$$

The computation formula of the output feature map is:

$$M_F(F) = max(0, (M_S \otimes F_S) \oplus F).$$

In sign language recognition, CIoU loss may not fully take into account the diversity of sign language in shape. In order to better focus on the difficult-to-recognize gestures, this paper introduces focal loss [27] as a loss function. Focal loss can assign higher weights to samples that are difficult to classify. The combination of focal loss with CIoU enables it to pay better attention to difficult samples, reduce missed detections, and improve detection performance.

$$L_{FocalCIoU} = \left(1 - IoU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v\right)^{\gamma},$$

where IoU refers to the intersection over union between the prediction box and the true box, $\rho^2(b, b^{gt})$ is the Euclidean distance between prediction box b and true box b^{gt}, c is the diagonal length of the minimum enclosing rectangle of the prediction box and the true box, α is the weight function, v refers to the aspect ratio similarity, and γ is an adjustment factor to mitigate the effect of sample imbalance on identification, 1.5 here.

4 Results and analysis

4.1 Experimental setup

The experiment was conducted in a Windows 10 environment, and the specific configuration is displayed in Table 2.

Table 2: Experimental environment.

	Configuration
Operating system	Windows 10
Compute unified device architecture	11.0
Programming language	Python 3.7
Deep learning framework	PyTorch 1.7.0
Central processing unit	Intel(R) Xeon(R) Gold 5218
Graphic processing unit	Tesla T4
YOLOv5 version	YOLOv5 v6.1
Image processing library	OpenCV 4.1.2; Pillow 8.2.0

Table 3 presents the parameter settings in the improved YOLOv5 algorithm.

Table 3: The training parameters of the improved YOLOv5 algorithm.

	Numerical value
IoU threshold	0.5
Epochs	200
Batch size	16
Optimizer	Stochastic gradient descent
Initial learning rate	0.01
Weight decay factor	0.0005

The following indicators were used to evaluate the effectiveness of sign language recognition:

(1) $Precision = \frac{TP}{TP+FP}$,

(2) $Recall = \frac{TP}{TP+FN}$,

(3) $mAP = \frac{\sum_{K=1}^{N} P(K)\Delta R(K)}{C}$.

In the above equations, TP denotes the quantity of positive samples identified as positive, FP is the quantity of negative samples identified as positive, FN is the quantity of positive samples identified as negative, N is the sample size of the test set, C is the number of categories, $P(K)$ is the P value when simultaneously identifying K samples, and $\Delta R(K)$ is the change of the R value when the number of samples to be identified changes from $K-1$ to K.

The mean average precision (mAP) when the IoU threshold was 0.5 was used.

Static sign language recognition has significant social significance in practice and can provide convenience for hearing-impaired people. Therefore, this paper mainly studied static sign language identification. The static sign language images used were from the American Sign Language (ASL) dataset [28]. This dataset contains 26 English letters and has been widely applied in the current research of static sign language recognition. Moreover, it involved 36 sign languages: space, del, nothing, and the letters A-Z, and included 87,000 images in a size of 200×200. Thirty thousand images were randomly selected for the experiments. Ten-fold cross-validations were used, and the results were expressed as mean ± standard deviation. Moreover, statistical tests and analyses were conducted in the SPSS26.0 software.

4.2 Experimental results

In order to determine the optimal location of the CBAM in the YOLOv5 network, the effects of different CBAM locations on sign language recognition were compared. The YOLOv5 algorithm without CBAM was used as a baseline model, and the CBAM was added at the following locations:

(1) after the CSP structure of the backbone network,

(2) after the SPP structure of the neck network,

(3) before the convolutional structure of the head network.

It is assumed that if CBAM is added after the SPP structure of the neck network, it can pay more attention to the easily ignored targets.

Table 4: Effects of different locations of CBAM on sign language recognition.

	P/%	**R/%**	**mAP@0.5 %**
Base	88.12 ± 2.74	90.33 ± 3.01	94.21 ± 2.81
Backbone	88.97 ± 2.78	90.59 ± 3.98	94.77 ± 3.68

Neck	89.07 ± 3.01	94.52 ± 4.27	96.87 ± 3.56
Head	81.17 ± 2.89	95.12 ± 3.64	95.07 ± 3.62
F value	3.695	3.841	3.261
P value	0.001**	0.002**	0.004**

Note: **: $p < 0.01$

As shown in Table 4, the addition of the CBAM at different locations within the YOLOv5 network had an impact on sign language recognition results. For instance, when the CBAM was added to the head section, the P value was the lowest, only 81.17%, but the R value was improved to 95.12±3.64%, and the final mAP value was 95.07±3.62%. Moreover, when the CBAM was added in the neck section, the P value was the highest, the R value was second only to the head, and the mAP value was also the highest, reaching 96.87 ± 3.56%. It was found through comparison that different locations of CBAM led to significant differences in sign language recognition results ($p < 0.01$). The performance was the best when the CBAM module was added to the neck part.

In order to assess the optimization effectiveness of focal CIoU on the YOLOv5 algorithm, the loss function, including IoU, generalized IoU (GIoU) [29], distance IoU (DIoU) [30], CIoU, and focal CIoU, were respectively used in the original YOLOv5 algorithm.

Table 5 shows that the traditional YOLOv5 algorithm (with the IoU loss function) had a low P value, R value, and mAP, suggesting a poor performance in sign language recognition. However, after improving the loss function, the sign language recognition performance of the YOLOv5 algorithm showed an improvement. It was found through comparison that different loss functions resulted in significant differences in sign language recognition results ($p < 0.01$), and the performance was best when focal CIoU was used.

Table 5: Effects of loss function on handwriting recognition.

	P	**R**	**mAP@0.5%**
IoU	88.12 ± 2.74	90.33 ± 3.01	94.21 ± 2.81
GIoU	89.07 ± 2.68	90.56 ± 2.87	94.33 ± 2.79
DIoU	90.12 ± 2.77	91.88 ± 2.93	94.95 ± 2.87
CIoU	90.54 ± 2.76	92.37 ± 2.84	95.12 ± 3.12
Focal CIoU	91.67 ± 2.61	94.87± 3.21	96.64 ± 3.07
F value	3.564	3.528	3.425
P value	0.002**	0.007**	0.009**

Note: **: $p < 0.01$

Ablation experiments were performed on the improved algorithm to analyze the effect of various module improvements on the model's performance (Table 6).

Table 6: Ablation experiments.

	P/%	**R/%**	**mAP@0.5/%**
Base	88.12 ± 2.74	90.33 ± 3.01	94.21 ± 2.81
Base+CBAM	89.07 ± 2.64	94.52 ± 2.32	96.87 ± 2.56
Base+CBAM+Focal CIoU	93.26 ± 2.77	96.77 ± 2.68	98.12 ± 2.32
F value	3.784	3.452	3.415
P value	0.007**	0.005**	0.006**

Note: **: $p < 0.01$

It was found that adding the CBAM to the YOLOv5 algorithm significantly improved the R value. Introducing focal CIoU based on CBAM further enhanced the model's recognition performance. It was found through comparison that the differences were significant ($p < 0.01$). These results validated the effectiveness of the improvement made to the YOLOv5 algorithm.

Moreover, the improved YOLOv5 algorithm was compared with other recognition methods (Table 7).

The Faster region-CNN algorithm was less effective in sign language recognition. Among the YOLO series algorithms, the YOLOv3 and YOLOv4 algorithms achieved mAP values slightly lower than the improved YOLOv5 algorithm. The results demonstrated the effectiveness of experiments on the improved YOLOv5 algorithm. Comparing the improved YOLOv5 algorithm with MobileNetV2 and ShuffleNetV2, the improved YOLOv5 algorithm achieved a higher mAP value. The statistical tests also suggested significant differences. These findings further validated the effectiveness of the proposed approach for sign language recognition.

Table 7: Comparison with other recognition algorithms.

	P/%	**R/%**	**mAP@0.5/%**
Faster region-CNN	80.12 ± 1.87	67.89 ± 1.65	79.84 ± 1.77
YOLOv3	87.77 ± 2.01	80.12 ± 1.77	90.31 ± 2.01
YOLOv4	88.05 ± 1.97	83.25 ± 1.56	92.56 ± 2.33
YOLOv5	88.12 ± 2.07	90.33 ± 1.64	94.21 ± 2.14
MobileNetV2	91.12 ± 2.56	81.94 ± 1.82	91.27 ± 2.05
ShuffleNetV2	91.08 ± 2.33	82.11 ± 2.01	91.26 ± 2.17
Improved YOLOv5	93.26 ± 2.77	96.77 ± 2.68	98.12 ± 2.32
F value	3.427	3.714	3.526
P value	0.008**	0.007**	0.008**

Note: **: $p < 0.01$

5 Discussion

This paper developed a YOLOv5 algorithm combining the CBAM attention module and focal CIoU to recognize static sign language images. The performance of the proposed method in static sign language identification was verified through experiments on the ASL dataset.

The results showed that adding the CBAM attention module and focal CIoU improved the detection performance of the YOLOv5 algorithm. CBAM can adaptively learn which pixels and channels are more important, which can not only improve the accuracy but also reduce the complexity of the model and alleviate overfitting. It has extensive applications in deep neural networks. The experimental results on the ASL dataset also verified the reliability of embedding the CBAM module into the YOLOv5 structure. Focal CIoU improves the detection performance by better focusing on the targets that may be ignored. Through comparison, it was found that compared with other loss functions, the P, R, and mAP values of focal CIoU were all higher, and the differences were significant ($p < 0.01$).

The results verified the performance of the improved YOLOv5 algorithm in recognizing static sign language images. Therefore, this method can be extended to the recognition of other static images, and it can also be introduced into the recognition of dynamic sign language videos by converting dynamic sign language videos into static sign language images.

However, there are also some limitations in this study. For instance, experiments were only conducted on a single dataset, and the recognition of continuous sign language was not achieved. In future work, further verification will be carried out on a broader range of datasets, and the recognition issues of dynamic and continuous sign language will be considered.

6 Conclusion

This paper presents an improved YOLOv5 algorithm for sign language identification in hearing-impaired people. The performance of the proposed algorithm was assessed using the ASL dataset. The results demonstrated that adding the CBAM enhanced the algorithm's recognition performance. Specifically, introducing the CBAM into the neck section yielded the best results. Moreover, focal loss further improved the algorithm's performance in sign language recognition. These results highlight the practical applicability of the proposed approach in actual sign language recognition, ultimately aiding in communication for people with hearing impairments.

References

[1] Nandhini MAS, Shiva Roopan D, Shiyaam S, Yogesh S. Sign language recognition using convolutional neural network. *Journal of Physics: Conference Series*, 1916(1), pp. 1-11. https://doi.org/10.1088/1742-6596/1916/1/012091.

[2] Sahoo AK (2021). Indian sign language recognition using machine learning techniques. *Macromolecular Symposia*, 397(1), pp. 2000241-1-2000241-7. https://doi.org/10.1002/masy.202000241.

[3] Xu B, Huang S, Ye Z (2021). Application of tensor train decomposition in S2VT model for sign language recognition. *IEEE Access*, 9, pp. 35646-35653, https://doi.org/10.1109/ACCESS.2021.3059660.

[4] Al Rafi A, Hassan R, Rabiul Islam M, Nahiduzzaman M (2023). Real-time lightweight bangla sign language recognition model using pre-trained MobileNetV2 and conditional DCGAN. *Proceedings of International Conference on Information and Communication Technology for Development*, 2023, pp. 263-276. https://doi.org/10.1007/978-981-19-7528-8_21.

[5] Takahashi R, Saito H (2022). Sign language recognition with 3D CNN transformer. *Proceedings of the Annual Conference of JSAI*, , pp. 4C1GS703-4C1GS703. https://doi.org/10.11517/pjsai.JSAI2022.0_4C1GS703.

[6] Yu Y, Chen X, Cao S, Zhang X, Chen X (2020). Exploration of chinese sign language recognition using wearable sensors based on deep belief net. *IEEE Journal of Biomedical and Health Informatics*, 24(5), pp. 1310-1320. https://doi.org/10.1109/JBHI.2019.2941535.

[7] Joshi JM, Patel DU (2024). GIDSL: Indian-Gujarati isolated dynamic sign language recognition using deep learning. *SN Computer Science*, 5, pp. 527. https://doi.org/10.1007/s42979-024-02776-7

[8] Wang QS, Zheng ZW, Wang Q, Deng D, Zhang J (2024). Generalizations of wearable device placements and sentences in sign language recognition with transformer-based model. *IEEE Transactions on Mobile Computing*, 23(10), pp. 10046-10059. https://doi.org/10.1109/TMC.2024.3373472

[9] Sharma V, Sharma A, Saini S (2024). Real-time attention-based embedded LSTM for dynamic sign language recognition on edge devices. *Journal of Real-Time Image Processing*, 21(2), pp. 53.1-53.13.

[10] Kourbane I, Genc Y (0021). Skeleton-aware multi-scale heatmap regression for 2D hand pose estimation. *Informatica*, 45(4), pp. 593-604. https://doi.org/10.48550/arXiv.2105.10904.

[11] Ren S, He K, Girshick R, Sun J (2017). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), pp. 1137-1149. https://doi.org/10.1109/TPAMI.2016.2577031.

[12] Yeh CC, Chang YL, Alkhaleefah M, Hsu PH, Eng W, Koo VC, Huang B, Chang L (2021). YOLOv3-based matching approach for roof region detection from drone images. *Remote Sensing*, 13(1), pp. 1-23. https://doi.org/10.3390/rs13010127.

[13] Wang L, Zhao Y, Liu S, Li Y, Chen S, Lan Y. (2022). Precision detection of dense plums in orchards using the improved YOLOv4 model. *Frontiers in Plant*

Science, 13, pp. 839269. https://doi.org/10.3389/fpls.2022.839269.

[14] Sandler M, Howard A, Zhu M, Zhmoginov A. Chen LC (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510-4520. https://doi.org/10.1109/CVPR.2018.00474.

[15] Ma N, Zhang X, Zheng HT, Sun J. (2018). ShuffleNet V2: Practical guidelines for efficient cnn architecture design. *European Conference on Computer Vision*, pp. 122-138. https://doi.org/10.1007/978-3-030-01264-9_8.

[16] Ogawa T, Uchida Y, Nishita Y, Tange C, Sugiura S, Ueda H, Nakada T, Suzuki H, Otsuka R, Ando F, Shimokata H (2019). Hearing-impaired elderly people have smaller social networks: A population-based aging study - ScienceDirect. *Archives of Gerontology and Geriatrics*, 83, pp. 75-80. https://doi.org/10.1016/j.archger.2019.03.004.

[17] Enikeev DG, Mustafina SA (2021). Sign language recognition through Leap Motion controller and input prediction algorithm. *Journal of Physics: Conference Series*, 1715(1), pp. 1-7. https://doi.org/10.1088/1742-6596/1715/1/012008.

[18] Tyagi A, Bansal S (2022). Hybrid FiST_CNN approach for feature extraction for vision-based Indian sign language recognition. *The International Arab Journal of Information Technology*, 19, pp. 403-411. https://doi.org/10.34028/iajit/19/3/15.

[19] Fu L, Yu H, Li X, Przybyla CP, Wang S. Deep learning for object detection in materials-science images: a tutorial. *IEEE Signal Processing Magazine*, 39(1), pp. 78-88. https://doi.org/10.1109/MSP.2021.3121558.

[20] Mopidevi S, Prasad MVD, Kishore PVV (2023). Multiview meta-metric learning for sign language recognition using triplet loss embeddings. *Pattern Analysis and Applications: PAA*, 26(3), pp. 1125-1141. https://doi.org/10.1007/s10044-023-01134-2.

[21] Das S, Biswas S K, Purkayastha B (2024). Occlusion robust sign language recognition system for indian sign language using CNN and pose features. *Multimedia Tools and Applications*, 83(36), pp. 84141-84160. https://doi.org/10.1007/s11042-024-19068-0.

[22] Yadav YG, Kiran VS, Karthik V, Thadikamalla GA, Kumaran P (2024). Real time sign language recognition using custom convolutional neural network and YOLOv5. *International Conference on Intelligent Computing, Smart Communication and Network Technologies*, pp. 157-171. https://doi.org/10.1007/978-3-031-75957-4_14.

[23] Nath B, Sarkar S, Das S, Mukhopadhyay S (2022). Neural machine translation for Indian language pair using hybrid attention mechanism. *Innovations in Systems and Software Engineering*, 20, pp. 175-183. https://doi.org/10.1007/s11334-021-00429-z.

[24] Zhu W, Shu Y, Liu S (2022). Power grid field violation recognition algorithm based on enhanced YOLOv5. *Journal of Physics: Conference Series*, 2209(1), pp. 1-10. https://doi.org/10.1088/1742-6596/2209/1/012033.

[25] Lv S, Liu X, Cao Y (2024). Remote sensing image recognition of dust cover net construction waste: a method combining convolutional block attention module and U-Net. *Sensors & Materials*, 36(7, Part 3), pp. 3131. https://doi.org/10.18494/SAM5182.

[26] Li R, Wang X, Wang J, Song Y, Lei L (2020). SAR target recognition based on efficient fully convolutional attention block CNN. *IEEE Geoscience and Remote Sensing Letters*, 19, pp. 1-5. https://doi.org/10.1109/LGRS.2020.3037256.

[27] Wang S, Chen M, Ratnavelu K, Shibghatullah ASB, Keoy KH (2024). Online classroom student engagement analysis based on facial expression recognition using enhanced YOLOv5 for mitigating cyberbullying. *Measurement Science and Technology*, 36(1), pp. 015419. https://doi.org/10.1088/1361-6501/ad8a80.

[28] Sharma A, Chopra A, Singh M, Pandey A (2022). American sign language gesture analysis using tensorflow and integration in a drive-through. *International Conference on Advances in Computing and Data Sciences*, pp. 399-414. https://doi.org/10.1007/978-3-031-12638-3_33.

[29] Qian X, Zhang N, Wang W (2023). Smooth GIoU loss for oriented object detection in remote sensing images. *Remote Sensing*, 15, pp. 1259. https://doi.org/10.3390/rs15051259.

[30] Yuan D, Shu X, Fan N, Chang X, Liu Q, He Z (2022). Accurate bounding-box regression with distance-IoU loss for visual tracking. *Journal of Visual Communication and Image Representation*, 83, pp. 1.1-1.10. https://doi.org/10.1016/j.jvcir.2021.103428.

Deciphering COVID-19 Narratives: A Comparative Study of ML Models (RF, MNB, GB, LR, SVM) and DL Models (CNN, Bi-LSTM) for News Article Classification

Kana Das[1], Md. Asadullah[2], Md. Murad Hossain1[*], Annita Siddeka Tanni[1], Shahidul Islam[3], Masudul Islam[4], Mst Sharmin Akter Sumy[5]
[1]Department of Statistics, Bangabandhu Sheikh Mujibur Rahman Science and Technology University, Gopalganj-8100, Bangladesh
[2]Bangladesh Rice Research Institute, Gazipur-1701, Bangladesh
[3]MIS Department, Friendship, Bangladesh
[4]Statistics Discipline, Khulna University, Khulna, Bangladesh; masudul_stat@ku.ac.bd;
[5]Department of Bioinformatics and Biostatistics, University of Louisville, 485 E.
Gray St, Louisville, 40202, KY, USA
E-mail: 12kanadas43@gmail.com, asadullahstat@gmail.com, murad.stat@bsmrstu.edu.bd, annitatanni28@gmail.com, jushahidms@gmail.com, masudul_stat@ku.ac.bd , mstsharminakter.sumy@louisville.edu
[*]Corresponding author

Keywords: Machine Learning (ML), Natural Language Processing (NLP), Roc Curve, GloVe, and FastText

Received: June 24, 2024

The COVID-19 pandemic has provided an unprecedented amount of information in news outlets, which include scientific, health-related, political, economic, and social narratives. This study compares the effectiveness of machine learning and deep learning algorithms for classifying text data, with a certain emphasis on how well the former handle COVID-19 news narratives. The study dataset contains news articles regarding COVID-19. To achieve the primary purpose of this research is to classify COVID-19 related news, we integrate multiple datasets. The analysis reveals machine learning models exhibit superior performance in text data classification. In particular, the Random Forest model reaches a 98% accuracy rate. In contrast, with regards to deep learning models, the Bidirectional Long Short-Term Memory model with FastText integration turns out to be the best option due to its exceptional accuracy. Exploratory data techniques such as topic modeling and word cloud approaches are incorporated to uncover hidden patterns in the data. Pre-trained (e.g., deep learning) and non-pre-trained ML models are implemented highlighting the versatility of ML in text classification tasks. The specific purpose to compare to the deep learning and machine learning algorithm to classification of the new article. Notably, a predictive model employing Bi-LSTM with the FastText pre-trained model achieved an impressive 94% accuracy in classifying COVID-19 news reports.

Povzetek: Primerjalna analiza modelov ML in DL za klasifikacijo novic COVID-19 razkrije RF kot najbolj natančno (98 %), medtem ko Bi-LSTM s FastText (94 %) odlikuje kontekstualno razumevanje, kar izboljšuje učinkovitost klasifikacije besedila.

1 Background of the study

Natural language processing has made extensive use of text classification to divide texts into different classes. In the context of COVID-19 news articles, text classification is crucial for identifying and categorizing the vast amount of information generated daily. In text categorization tasks, machine learning algorithms have demonstrated encouraging results, especially when applied to COVID-19 news items. Empirical evaluation has been conducted to determine the efficacy of several machine learning techniques for text classification of COVID-19 news items.
Jin et al. (2024) investigated the use of AI techniques, such as natural language processing and machine learning, to improve text classification [1]. Their findings highlight the potential of technologies to enhance accuracy and efficiency in text processing, supporting information retrieval and decision-making despite operational challenges. Didi et al. (2022) undertook studied on the categorization of tweets related to COVID-19 using machine learning methods and analysed public sentiment about the pandemic through their novel hybrid feature extraction method that combines syntactic elements with semantic aspects for more accurate text data representation and enhanced classification. Their research built upon previous work exploring Twitter's potential in understanding public opinion during the pandemic, focusing on sentiment analysis using ML models as naïve Bayes and Logistic Regression [2]. The authors demonstrated with a focus on diagnostics and predictive modelling. It emphasized deep learning's impact on healthcare applications, especially

Convolutional Neural Networks while addressing challenges like data scarcity and diversity [3]. Abdeen et al. (2021) introduced NeoNet, a cutting-edge machine learning algorithm created to categorize news stories and medical papers about COVID-19 according to their degree of veracity. Leveraging advanced Term Frequency-Inverse Document Frequency bigram features, NeoNet enables accurate prediction of document relevance and accuracy, with the goal of widespread adoption across various social media platforms [4]. In his evaluation of the COVID-19 pandemic's fake news prevalence and its effects, Abhishek Koirala discussed earlier studies on the identification of fake news and brings up the establishment of the Liar Dataset. The study addressed new challenges posed by COVID-19-related news, experimenting with deep learning models to improve classification accuracy but notes inconsistencies in the dataset that may hinder this process [5].

The methodologies and efficacy of neural network and neuro-fuzzy algorithms in detecting Covid-19-related disinformation on social media are thoroughly examined by Ravichandran and Keikhosrokiani (2023) [6]. This study highlights the role of NF and NN methodologies in assessing their strengths, and limitations, and providing recommendations for future research. Chughtai et al. (2021) paper emphasizes the role feature selection has in model performance and uses benchmark datasets to demonstrate the efficacy of their SVM classifier using MMR algorithm, emphasizing its improvement in F1-score and role in detecting misinformation [7]. Arbane et al. (2023) stress the importance of using advanced machine learning algorithms for automatic sentiment analysis to gain insights from social media data during the COVID-19 pandemic [8]. Previous studies, including those by Mansoor et al. (2020) and Samuel et al. (2020) used various machine-learning models to evaluate public sentiments at different pandemic stages, revealing details such as the impact of lockdowns on emotional responses [9,10]. Additionally, other researchers incorporated deep learning methods to examine sentiments [11]. Using machine learning classifiers [11], researchers investigated sentiments expressed on Twitter during the pandemic using various machine learning classifiers and predicts sentiments in two datasets collected before and after lockdown periods. The results suggest changes in public sentiment during and after lockdowns, providing insights into public opinion dynamics in response to the pandemic and associated restrictions [12].

Dangi et al. (2022) examines the nature of COVID 19 news coverage across the United Kingdom, India, Japan, and South Korea using topic modeling and sentiment analysis with top2vec and RoBERTa to deeply analyse a large amount of news data. It introduces an extensive approach that utilizes the top2vec algorithm to identify underlying topics in news articles, and RoBERTa for categorizing sentiments [11]. They focused on sarcasm detection using lexical and word embedding approaches while others developed systems for identifying fake news by analysing sentiment and named entity features from Twitter data [12]. Madani (2021) discussed about the MVEDL ensemble deep learning model, which is intended to categorize tweets about COVID-19 as informative or not. They evaluated on the "COVID-19 English labelled tweets" dataset, transformer models like RoBERTa, BERTweet, and CT-BERT that demonstrated a strong performance with an accuracy of 91.75% and an F1-score of 91.14% [14]. Malla and Alphonse (2021) investigate how well several transformer-based models—including BERT, RoBERTa, ALBERT, and DistilBERT perform in text classification for uses like sentiment analysis and the identification of false news. Their findings suggest that these models can significantly improve accuracy in classifying challenging data types found on social media platforms [15].

Qasim et al. (2022) explores the application of domain-specific BERT-based models, such as BioBERT and CovBERT. They highlight the superiority of CovBERT in handling vocabulary deficiencies in scientific summaries, achieving up to 94% accuracy compared to its predecessors [16]. They also examine the transition from machine learning to deep learning models and highlight the potential of the Cov-Dat-20 dataset in assisting epidemiologists in addressing the challenges posed by COVID-19. Khadhraoui et al. (2022) explores sentiment analysis of COVID-19-related tweets, covering research in different languages like Nepali and review underscores the constraints of current methodologies. They support a backdrop for the authors' exploration of hybrid feature extraction methods to enhance the classification of Nepali COVID-19 tweets [17]. Shahi et al. (2022) integrates Bidirectional Long Short-Term Memory neural networks with local interpretable model-agnostic explanations, aiming for an effective model and proposes experimentation with different RNN-based models, machine learning techniques, explains ability methods such as LIME and SHAP using datasets like the Constraint 2021 COVID-19 fake news dataset and the WNUT COVID-19 tweet dataset [18].

The challenge of accurately classifying biomedical papers related to the COVID-19 pandemic into relevant categories addressed by Ahmed et al. (2022). They look into numerous machine learning approaches such as document representation strategies, neural networks and random forests, focusing on a subset of the Lit Covid corpus and employing pre-processing and feature engineering methods such as TF-IDF, BOW, and BERT embedding [19]. Rabby and Berka (2023) discusses a range of studies related to the detection of fake news and spam, including exploration of review spam across millions of Amazon reviews, identification of spam in Arabic texts, comparison of different machine learning models using n-gram analysis to flag false information and focus on online articles for detecting fake news with a high accuracy rate [20]. Etaiwi (2022) also build an ensemble model utilizing fusion vector multiplication on a COVID-19 English fake news dataset, with a 98.88% accuracy and an F1-score of 98.93% for achieving high performance in their model evaluation [21]. Malla and Alphonse (2022) discuss the

challenge of classifying Covid-19 misinformation on social media conducting a systematic review of studies from 2018 to 2021 that use various machine learning techniques of misinformation classification, evaluates their efficiency, strengths, and limitations. They propose a novel hybrid ANFIS-DNN model to enhance accuracy and effectiveness in this domain [22].

Ravichandran and Keikhosrokiani (2023) present a novel collection of 24 Information Society (IS) laws, exploring their connections with electronics and artificial intelligence (AI) [23]. These laws highlight exponential growth in areas such as processing, storage, and communication, while addressing their societal and economic impacts. The researchers also discuss how AI advancements contribute to electronics and broader IS progress, emphasizing their interdependence and potential future developments [24]. Janko et al. (2021) investigate factors influencing the early spread of COVID-19 across countries, focusing on the period before countermeasures [25]. By analyzing a diverse dataset with statistical methods and machine learning (ML) feature selection, they identify key factors like culture, development, and travel. They also use a novel rule discovery algorithm to explore factor interconnections, cautioning against overreliance on ML alone. The best model, using a decision tree classifier, predicts infection classes with about 80% accuracy. The researcher of [26] presents SentiTextRank, an emotional variant of TextRank, for extractive summarization and classifying sentences into eight emotional categories from SenticNet, SentiTextRank using both single and multi-document summarization tasks. The work of authors in [27] investigate the emotional component of successful medical web pages related to spine pathology, hypothesizing that they would exhibit distinct emotional patterns. Using sentiment analysis and machine learning, the study retrieves high classification accuracy, with disgust emerging as a key emotion. The findings suggest that digital content impacts patients' biopsychosocial ecosystems, influencing chronic pain and health behaviours, raising ethical concerns for health information providers.

The comprehensive review of existing literature on text classification in the context of COVID-19 indicates several significant areas where further research is needed. The application of natural language processing, deep learning, and machine learning methods for categorizing documents, and analyzing sentiment, and fake news identification has advanced significantly, yet there are still unaddressed issues. To begin with, despite numerous studies focusing on fake news detection and sentiment analysis, the lack of standardized datasets and inconsistent labelling practices poses challenges for model generalization and benchmarking. Additionally, many studies have predominantly focused on English-language datasets without considering the requirement for multilingual models to combat misinformation across diverse cultural and linguistic contexts. While some research acknowledges the

Table 1: Comparison among the related articles.

Autor and Title References	Dataset	Algorithm	Result
[2]	Tweets dataset	TF-IDF, Word2Vec, Glove, and FastText, SVM.	Fast Text with TF-IDF performed better
[4]	COVID-19 News Articles Open Research dataset	TF-IDF, NN, SVM, and RF, NeoNet.	NeoNet better perform
[5]	COVID-19 News articles dataset	LR, Embedded LSTM, LSTM, Bidirectional LSTM.	Embedded LSTM Hybrid models better performer
[6]	COVID-19 News dataset between July 2018 and May 2021.	Neuro-fuzzy, NN and specially ANFIS.	Hybrid ANFIS-DNN better performer
[8]	COVID-19 related tweets and comments dataset.	LSTM, Bi-LSTM,	Bi-LSTM model is superior over LSTM.
[10]	Twitter datasets	LR, RF, MNB, SVM, and DT	Decision Tree Classifier better performer.
[11]	COVID-19 articles dataset	top2vec and RoBERTa	RoBERTa
[13]	COVID-19 English labeled tweets dataset	RoBERTA, CT-BERT, and BERTweet	Ensemble Deep Learning (MVEDL) model
[14]	COVID-19 fake news dataset and COVID-19 English tweet dataset	BERT-base, BERT-large, RoBERTa-base, RoBERTa-large, DistilBERT, XLM-RoBERTa-base, ALBERT-	RoBERTa-base model achieved the highest accuracy in COVID-19 fake news dataset, Bart-large, BERT-base are the respective winners of other datasets

		base-v2, Electra-small, and BART-large	
[15]	Dataset of biomedical articles on COVID-19	CovBERT and BERT	CovBERT outperform
[16]	Nepali-COVID-19 tweets dataset	FastText + TF-IDF, LR, SVM, NB, KNN, DT, RF, Extreme Tree classifier, AdaBoost, and MLP	FastText with TF-IDF, SVM + RBF is the best performing classifier.
[17]	Covid-19 fake news dataset	BiLSTM, LSTM, GRU, RNN, CNN, SVM, DT	BiLSTM model high classification accuracy
[18]	COVID-19 Open Research dataset	RF, Logistic regression, KNN, DT Multi-layer Perceptron, Neural Network (BERT), BOW	Random Forest and Neural Network (BERT)
[20]	COVID-19 fake news dataset	BERT, BERTweet, AlBERT, CT-BERT, RoBERTa and DistlBERT	RoBERTa
[21]	Covid-19 misinformation related papers dataset	Neuro-Fuzzy, Neural Network	ANFIS-DNN model

importance of explain ability in classification models; limited research exists on implementing and evaluating reasonable AI specifically tailored for COVID-19-related text classification. The following Table 1 highlights the current state of research on text classification using machine learning algorithms for COVID-19 news articles with comparison among them.

Table 1 shows most of the research for text data analysis are not using feature extraction technique. Some of the literature use FastText or TF-IDF technique for feature extraction technique. But they skip some powerful feature extraction technique like BERT or Glove technique. In this research, we utilize FastText, TF-IDF and BERT techniques and compare them.

2 Methods and materials

The three segments of the text classification approach are features extraction, text pre-processing, and dataset description are considered. In the Algorithm Selection phase, the innovative deep learning algorithms have been merged. How text pre-processing was done for machine learning is explained in the Text Classification Approach section. The mathematical description of the process is used to extract private data from the dataset and how text data can be transformed into a numeric form shown in the third phase. Lastly, the algorithms that have been shortlisted in this type of study are discussed in the fourth phase. Learning rate for Adam optimizer: 0.001 is considered as starting point. Batch sizes (e.g**.,** 16, 32, or 64) typically work well for text data analysis, especially in tasks like text classification or sentiment analysis. We use 32 for our text data analysis. Start with 10 epochs use early stopping to avoid overfitting.

2.1 Dataset description

The dataset from Kapoor et.al (2020) [34] and Lipenkova et.al (2021) [35] contained news articles regarding COVID-19 since the primary purpose of this research is to categorize news related to COVID-19. In this research we use news article text data. Two Datasets of news articles are extracted from www.inshorts.com and then labeled based on relation to COVID as well as the sentiment. They have been assembled from different repositories and reformatting for a similar distribution. After combining the two datasets, the sample size of our dataset is 14012 in total. Balancing imbalanced data for classification tasks in machine learning (ML) is crucial because imbalanced datasets can lead to biased models that favour the majority class and fail to detect the minority class effectively. The imbalance data use SMOTE (Synthetic Minority Over-Sampling Technique) interpolating between existing samples of the minority class. For analysis text data we pre-process the data by removing punctuation**,** removing numbers, removing special characters and symbols, removing URLs, emails, and mentions, removing stop words, tokenization, text normalization**,** vectorization of text, term frequency - inverse document frequency. The features column of Table 1, and the feature's value of Table 2 is the header-wise first content information of the dataset. This dataset consists of six attributes which are depicted below.

Table 2: Attribute of the proposed research dataset

Features	Feature's Value
Headline	Headline of the article
Sentiment	1 if the article is positive, 0 otherwise
Covid	1 if the article is related to COVID, 0 otherwise
Description	Description of the news
Image	Image URL
Source	Source URL

2.2 Features extraction algorithm

Two new classifications, pre-trained model configuration, and non-pre-trained model setup have been incorporated into the features extraction. This process comprises two parts: the non-pre-trained model setup and the pre-trained model configuration. The unique word embedding technique Text to sequence, Fast Text, and Glove is demonstrated by the PTMS. However, the NPTMS provided an explanation of the typical features extraction method: Inverse document frequency paired with term frequency (TF-IDF).

A) **Pre-trained model structure**

FastText: FastText is a library developed by Facebook's AI research (FAIR) lab, designed for efficient text classification and representation learning. It is particularly useful for text classification, word representation (word embeddings), and language modeling tasks. FastText improves traditional **Word2Vec** by representing words as **subword-level units** (i.e., n-grams), making it more effective for handling rare or out-of-vocabulary (OOV) words. Additionally, FastText supports effective training and inference, making it a popular tool for text classification tasks. It is an efficient and powerful tool for text classification, leveraging subword information to handle out-of-vocabulary (OOV) words and offering better performance in languages with rich morphology. FastText is a novel technique for word embedding represents words as bags of n-gram characters. This approach addresses the issue of morphology neglect, in other words embedding representations by capturing subword information explained [28]**.** Consider the term "introduce" with n equal to 3, FastText generates three-gram characters shown in the following representation:

$< in,$ int, $ntr,$ tro, rod, odu, duc, uce, ce $>$

We are considering a word w that is correlated using an n-gram dictionary with a size of G as a way to represent the vector for each n-gram g. In this case, the acquired scoring function defined in Spirovski et al. (2018) [28] is:

$$s(w,c) = \sum_{g \in gw} z_g^T v_c \quad (1)$$
$$\text{where } g_w \in \{1,2,\dots,G\}$$

Global Vectors (Glove): GloVe is another popular method for **word embedding** that is widely used in text analysis. GloVe was developed by Stanford researchers and designed to capture **global statistical information** about a corpus, unlike methods such as **Word2Vec**, which focus on local context windows. The key idea behind GloVe is the **co-occurrence matrix** of words in a corpus contains valuable information about the relationships between words. GloVe uses the co-occurrence data to generate dense word vector. Global Vectors (GloVe), a potent word embedding method has been applied to text classification [29]. This strategy bears a strong resemblance to Word2Vec, which provides a high-dimensional vector of each word and trains it across an extensive corpus using surrounding terms. Pre-trained word embedding are widely used, based on 50 dimensions for word presentation in Wikipedia 2014 and Gigaword 5, as well as 400,000 vocabularies introduced as the corpus [36]. Unlike the traditional word embeddings such as Word2Vec, which can't generate vectors for words not seen in training data, FastText can generate embeddings for any word by breaking it into subword units (n-grams). FastText is particularly useful for languages with complex morphology (e.g., Turkish, Finnish) or rare words as it captures meaningful subword features. Pre-trained FastText models are available for many languages, which can be used directly for feature extraction in downstream tasks, saving time and computational resources.

B) Non-pre-trained model structure

Term Frequency-Inverse Document Frequency (TF-IDF)

In order to reduce the influence of frequently occurring words in the dataset, inverse document frequency is a method that should be combined with term frequency [30]. Terms in the document that have a high or low frequency are given a higher weight by IDF. TF-IDF combines term frequency and inverse document frequency. Equation 2 mathematically represents a term's weight that is used in this study.

$$\text{TF-IDF}.W(d,t) = TF(d,t) * \log\left(\frac{N}{df(t)}\right) \quad (2)$$

In this scenario, $df(t)$ represents the number of documents in the corpus containing the word t, and N is the total document count. According to Tokunaga and Makoto the initial factor in equation 2 enhances recall, while the second term improves word embedding accuracy [37]. TF-IDF is a simple and computationally efficient method for text feature extraction. Unlike GloVe or FastText, TF-IDF doesn't require a pre-trained model. It can be computed directly from the text corpus. The

resulting features (weights) are easy to interpret since they are based on word frequency and document distribution.

2.3 Deep learning algorithms

CNN and RNN are the main types of deep learning architectures used for text classification. Hierarchical machine learning or deep learning involves a series of algorithms performed in sequential order.

2.3.1 Bidirectional long short-term-memory (Bi-LSTM)

Bi-LSTM input sequences can be in both directions with two neuron sub-layers. This orientation is to generate a complete
input context. There are also backward hidden sequences, namely $\overleftarrow{h}$, $\overrightarrow{h}$. From this configuration, we can compute the output sequence y: two neuron sub-layers can be used in both directions for Bi-LSTM input sequences. The goal of this viewpoint is to produce an entire input context. Backward hidden sequences are also present, denoted as $(h)^{\leftarrow}$ and $(h)^{\rightarrow}$. We can calculate the output sequence based on this arrangement y:

$$\vec{h}_t = \mathcal{H}\left(W_{x\vec{h}}x_t + W_{\vec{h}}\vec{h}\vec{h}_{t-1} + b_{\vec{h}}\right) \quad (3)$$

$$\overleftarrow{h}_t = \mathcal{H}\left(W_x h x_t + W_{\leftarrow\leftarrow h}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}}\right) \quad (4)$$

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y \quad (5)$$

It is an advanced architecture of the Long Short-Term Memory (LSTM) network, a type of recurrent neural network (RNN) is designed to learn sequential dependencies in data. The key feature of a Bi-LSTM is its ability to consider both past (backward) and future (forward) context while processing sequences. This makes it especially effective for tasks where the entire context of a sequence is critical, such as natural language processing (NLP), speech recognition, and time-series prediction.

2.3.2 Convolutional neural networks (CNN)

A widely used deep learning structure for categorizing hierarchical documents is the convolutional neural network defined [31]. While initially constructed for the processing of images, CNNs have proven to be useful for text classification as well explained [32]. CNN's use pooling to minimize the output's size from one layer to the next in the network to reduce the complexity of the computation. To minimize outcomes while retaining essential features, various pooling techniques are used [33]. The process of choosing the highest value in the pooling window is referred to as max pooling, which is a commonly employed technique. The feature maps are transformed into a single column before transmitting the pooled output from stacked feature maps to the next layer. In general, both the weights and the feature detector filters are modified during the back-propagation phase of a convolutional neural network. The number of channels is a potential issue that emerges when using CNN for text classification (size of the feature space). In general, the program has few channels (e.g., just 3 RGB channels) and can be very broad for text classification applications, resulting in very high dimensionality. The CNN based text classification architecture includes word embedding as input layer 1D convolution layers, 1D pooling layer, completely connected layers, and finally, the output layer [33].

2.4 ML model performance measure

Precision: The ratio of the model's accurate true positive estimate to the total positive estimate (including both correct and incorrect classifications). It is expressed as:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

Recall / Sensitivity: The predictive ratio shows a positive correlation and is expressed as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (7)$$

F1 score: This provides a more accurate estimate than the accuracy metric for the misclassified instances; it is calculated as the harmonic mean of Precision and Recall. In mathematical terms, it can be expressed as

$$\text{F1} = 2 \cdot \frac{\text{Precision.Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

Accuracy: The sum of all the precisely forecasted events. It is presented as:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (9)$$

2.5 Proposed algorithms architecture

The proposed algorithms provided a description of the algorithms utilized for COVID-19 related news classification. The findings indicate the effectiveness of machine learning algorithms in analyzing the text data used in this research. Both machine learning and deep learning methods are applied to categorize the text, as detailed in our proposed architecture shown in Figure 1.

3 Results analysis

The classification result is gathered using the deep learning (DL) approach and presented by the empirical consequence. The model assesses and analyzes precisely and the best model that emerged.

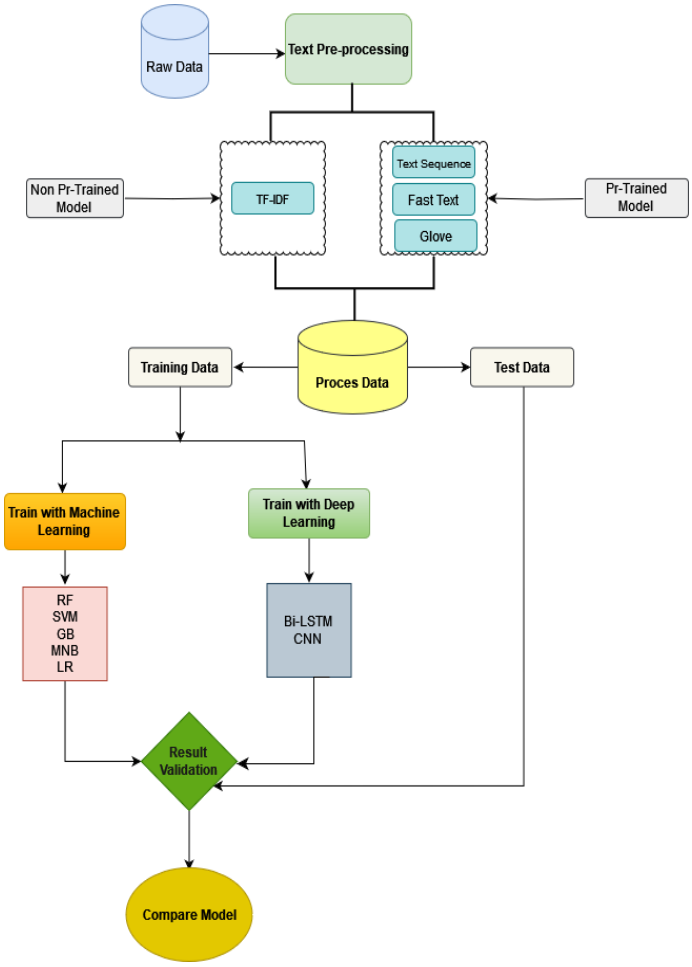


Figure 1: Architecture of the proposed algorithm

Table 3: Classification result for deep learning algorithms

Algorithm	In case of "0"				In case of "1"			
	Features Extraction Technique	**Precision**	**Recall**	**F1-score**	**Precision**	**Recall**	**F1-score**	**Accuracy**
CNN	Text to sequence	.93	.90	.92	.90	.93	.92	.92
BI-LSTM	Text to sequence	.93	.90	.92	.90	.93	.92	.92
BI-LSTM	Fast Text	.95	.92	.94	.93	.95	.94	.94
BI-LSTM	Glove	.93	.90	.92	.90	.99	.93	.92

Table 4: Classification result for machine learning algorithms (Baseline model)

Algorithm	In case of "0"				In case of "1"			
	Features Extraction Technique (FET)	Precision	Recall	F1-score	Precision	Recall	F1-score	Accuracy
RF	TF-IDF	.92	.85	.90	.88	.94	.9	.94
MNB	TF-IDF	.75	.86	.82	.87	.73	.8	.81
GB	TF-IDF	.88	.67	.78	.74	.91	.82	.8
LR	TF-IDF	.81	.82	.83	.84	.82	.83	.83
SVM	TF-IDF	.93	.93	.94	.95	.94	.94	.92

Table 5: Classification result for machine learning algorithms with (TF-IDF)

Algorithm	In case of "0"				In case of "1"			
	Features Extraction Technique (FET)	Precision	Recall	F1-score	Precision	Recall	F1-score	Accuracy
RF	TF-IDF	.97	.91	.94	.92	.98	.94	.98
MNB	TF-IDF	.80	.92	.86	.91	.77	.84	.85
GB	TF-IDF	.93	.73	.82	.78	.95	.86	.84
LR	TF-IDF	.86	.88	.87	.88	.86	.87	.87
SVM	TF-IDF	.98	.99	.98	.99	.98	.98	.96

The accuracy of the classification algorithms is evaluated based on a set of metrics for each class. These metrics involve accuracy, recall, and f1-score, computed using true and false positives along with false negatives. In Table 3, the classification results for deep learning algorithms and a cutting-edge natural language processing technique are presented based on feature extraction methodology. Additionally, Table 4 and Table 5 exhibit the classification outcomes from machine learning methods without any feature extraction technique and utilizing TF-IDF (term frequency inverse document frequency) approach.

3.1 Deep learning model result

Model assessment includes receiver operating characteristics, area under the curve, and confusion matrix. The algorithm that provides the best accuracy is used as the foundation for the model assessment. Using the Bi-LSTM and the Fast Text method, we were able to obtain a good score in Figure 9. The model assessment illustrates how well our suggested model works for the specific task.

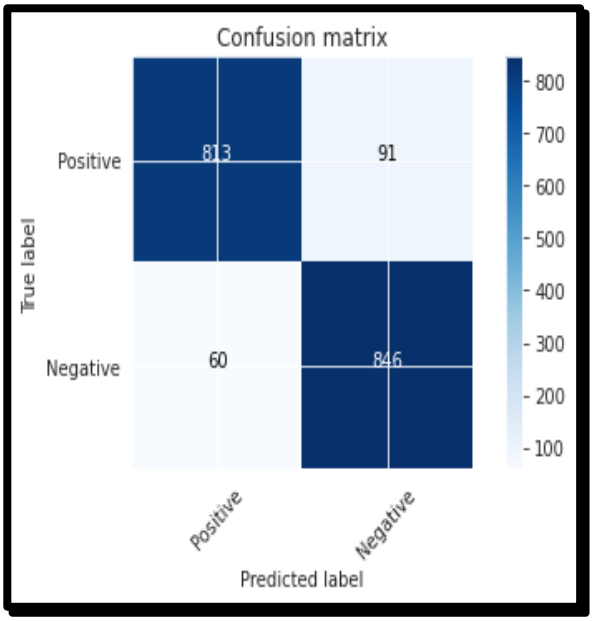


Figure-2: Confusion Matrix of Bi-LSTM using Glove

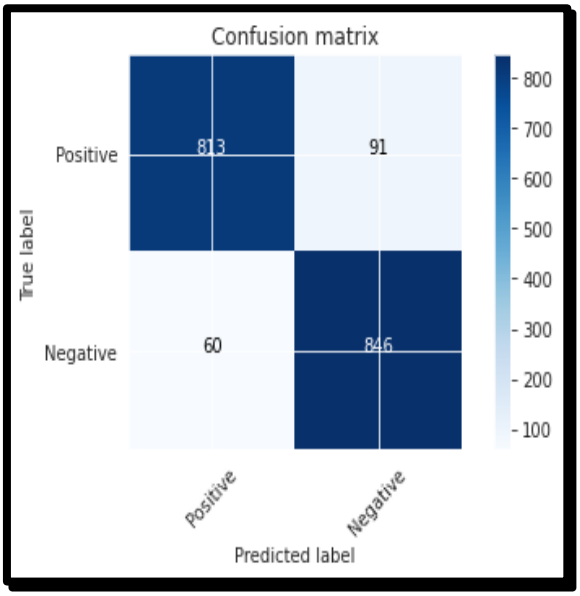


Figure-4: Confusion Matrix of CNN with text sequence

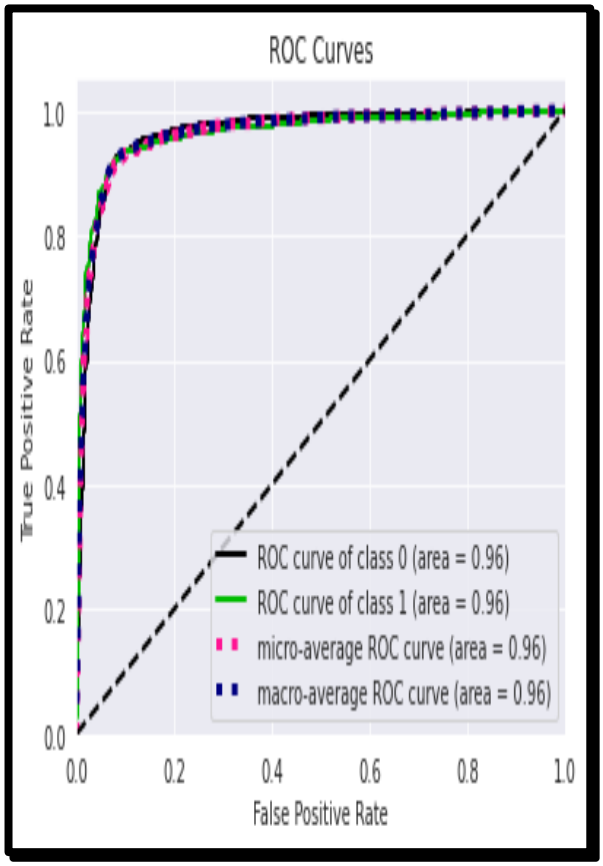


Figure-3: ROC-AUC curve of Bi-LSTM using Glove

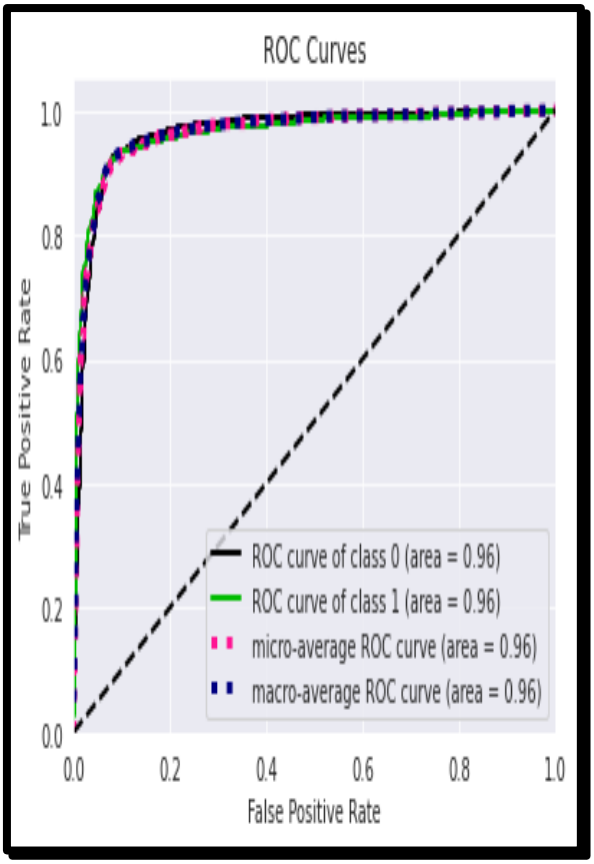


Figure-5: ROC-AUC curve of CNN with Text Sequence

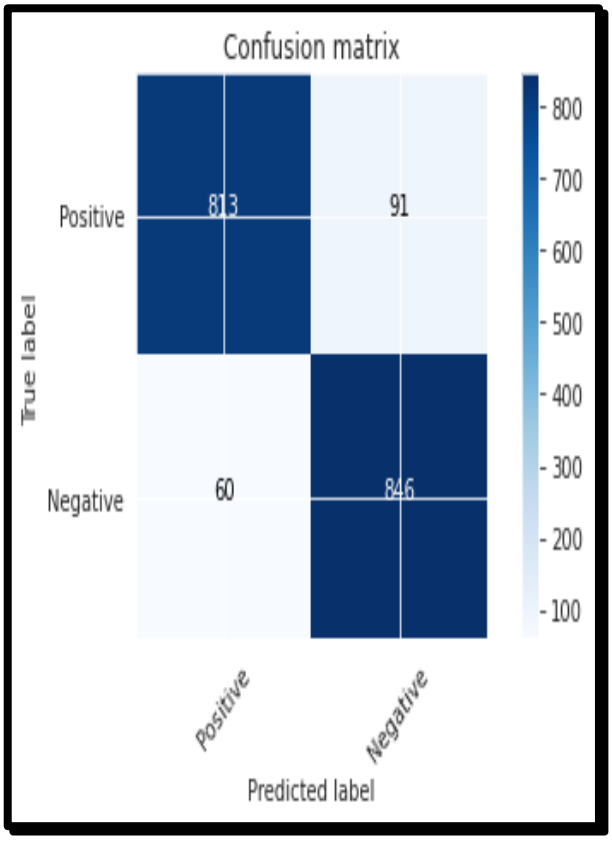


Figure-6: Confusion Matrix of Bi-LSTM with Text

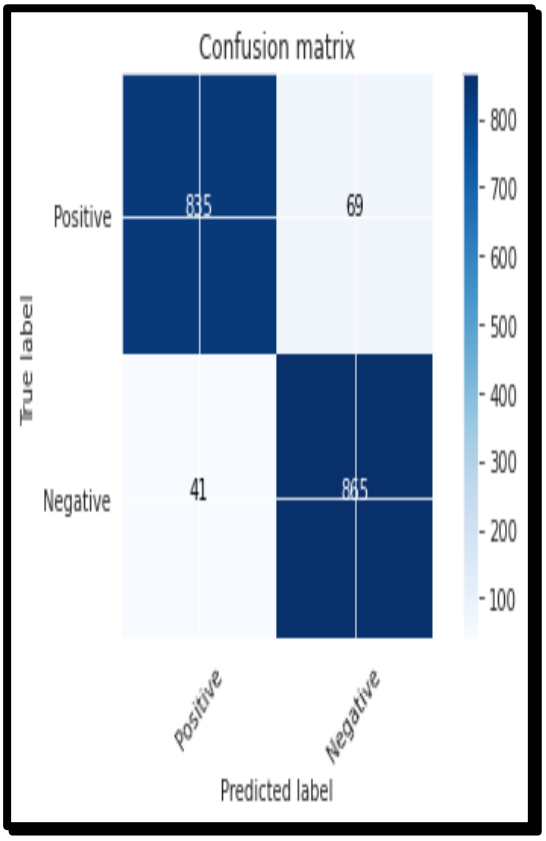


Figure-8: Confusion Matrix of Bi-LSTM using FastText

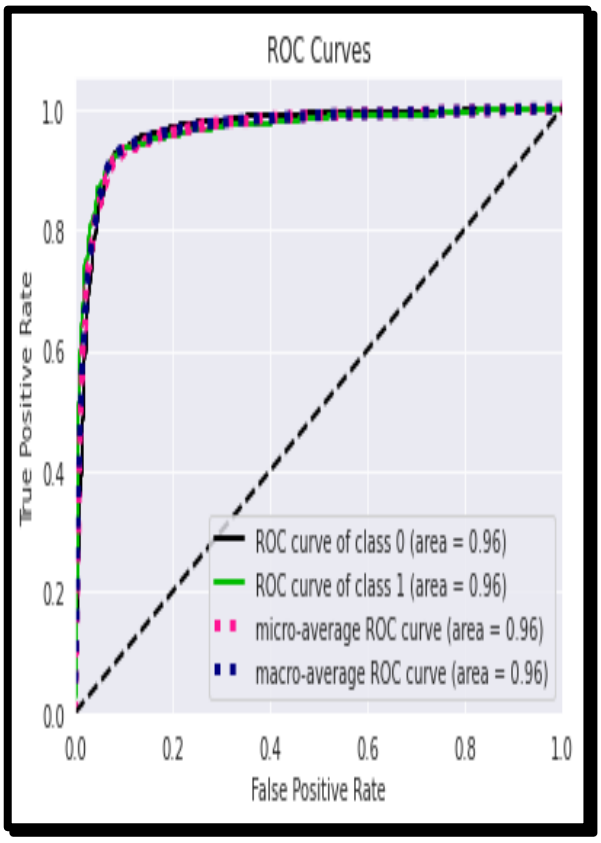


Figure-7: ROC-AUC curve f Bi-LSTM Text Sequence

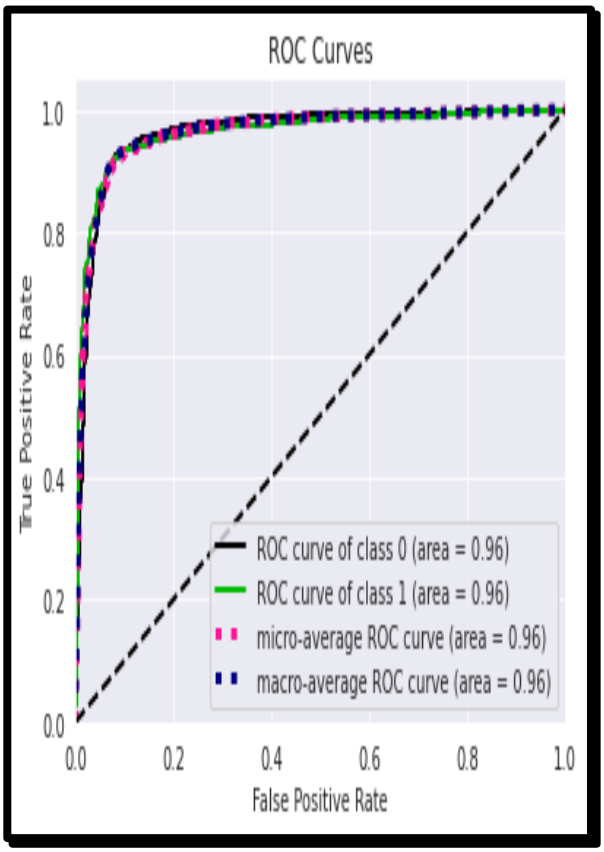


Figure-9: ROC-AUC curve of Bi-LSTM using FastText

3.2 ML model assessment

ML model evaluation is conducted using various approaches, including ROC, AUC, Accuracy of Models, and Confusion Matrix. Notably, the assessment in this section was performed utilizing the algorithm that yields the highest accuracy across all ML methods. We found a satisfactory score by using the SVM method. In this section shows tradional model assessment how well performs compare to the deep learning method in text classification. We only show the best model in machine learning method.

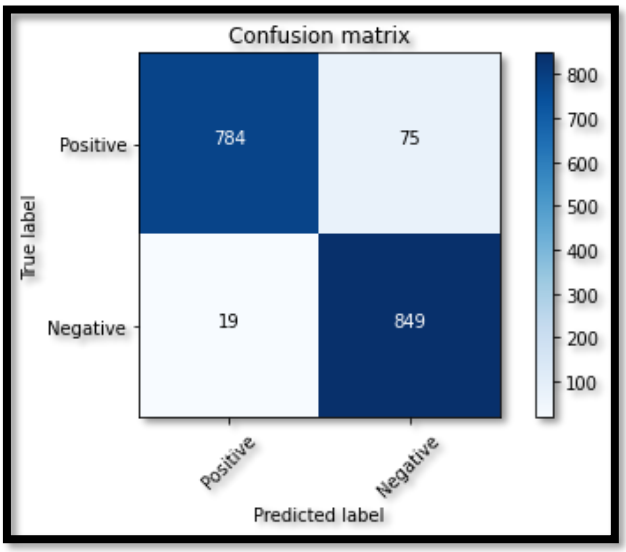


Figure-10: Confusion Matrix of RF

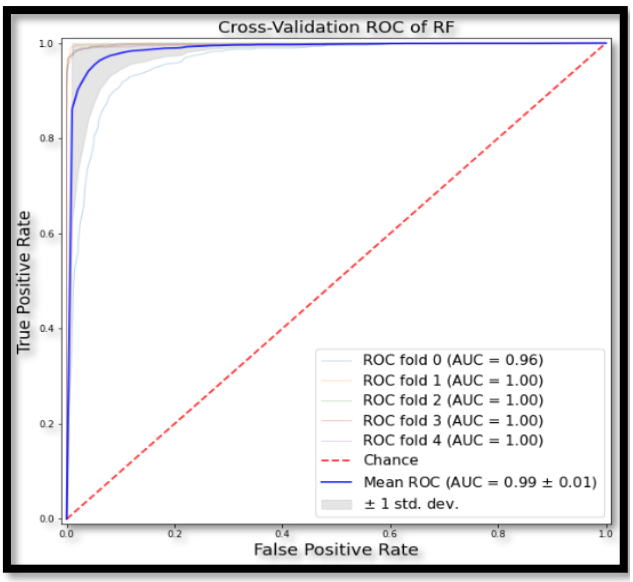


Figure-11: ROC-AUC curve of RF

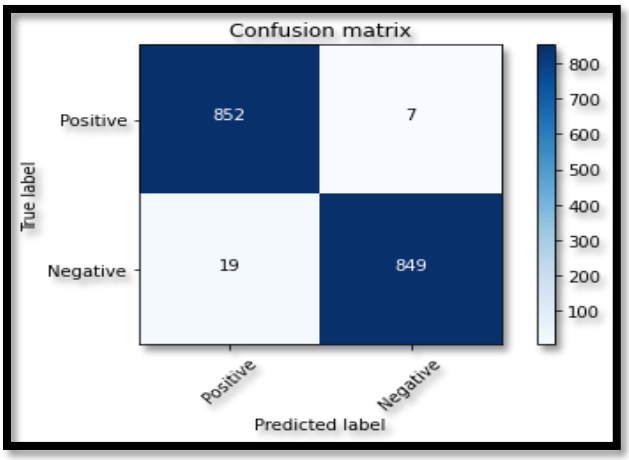


Figure-12: Confusion Matrix of SVM

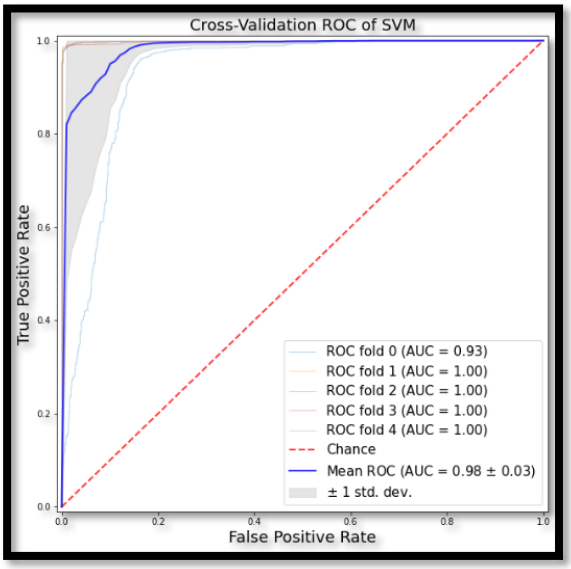


Figure-13: ROC-AUC curve of SVM

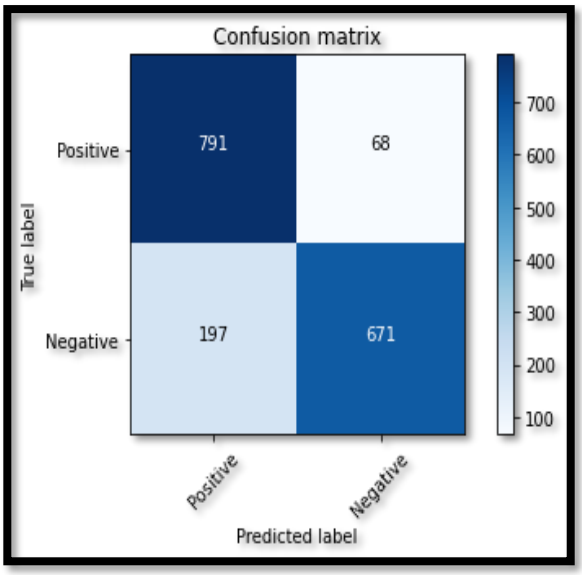


Figure-14: Confusion Matrix of MNB

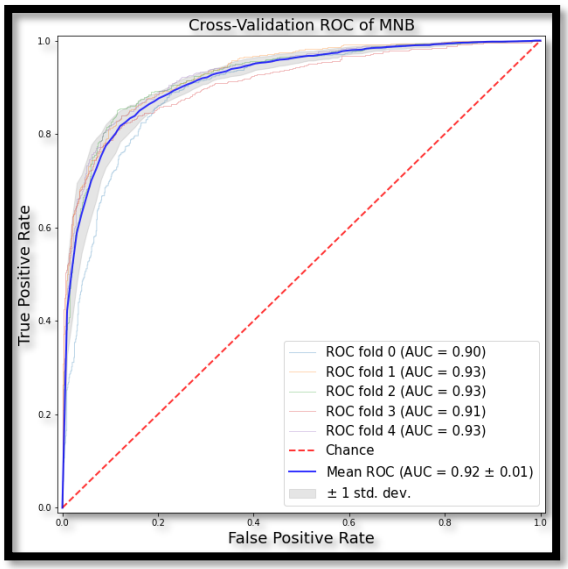


Figure-15: ROC-AUC curve of MNB

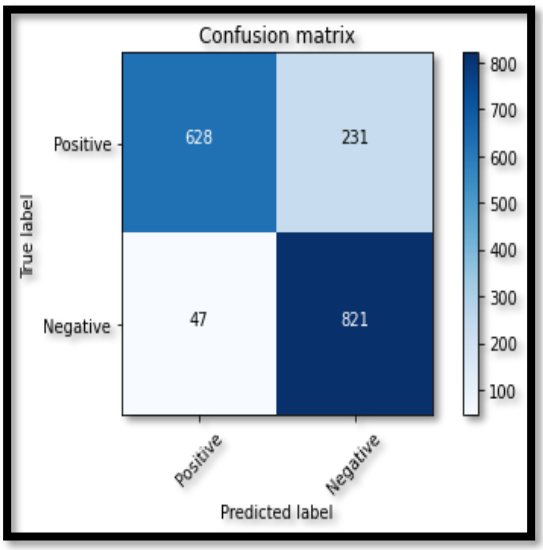


Figure-16: Confusion Matrix of GB

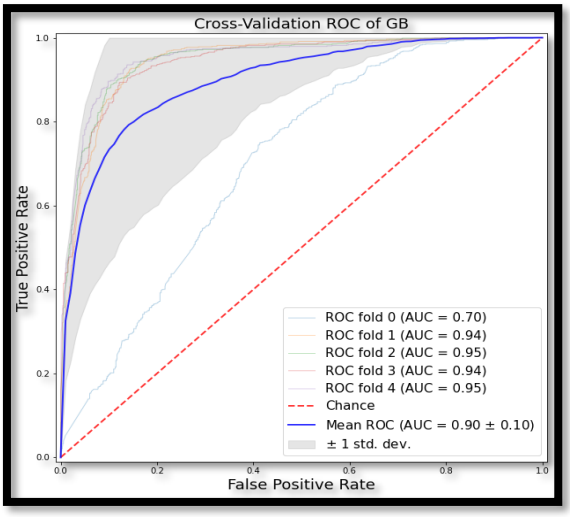


Figure-17: ROC-AUC curve of GB

Estimating binary classification problems often relies on the receiver operator characteristic curve, that plots true positive rate against the false positive rate at different threshold levels. The ROC serves as a probability graph that effectively distinguishes between signal and noise. A key metric derived from the ROC curve is crucial for evaluating a classifier's ability to differentiate between two groups: an AUC of 1 indicating accurate discrimination between positive and negative class points; while an AUC of 0 implies all negatives are predicted as positives and vice versa. For instance, as shown in Figure 11, the RF classifier model achieves an AUC value of approximately 0.96, demonstrating its precise discrimination capability between positive and negative class points.

3.3 Exploratory data analysis

Exploratory data analysis (EDA) is an important technique for examining the dataset and understanding its fundamental characteristics. The EDA provides valuable insights into the dataset combining topic modeling. While EDA uncovers meaningful patterns and observations, the topic modeling approach illustrates the hidden semantic structure of text and figured out the most dominant word in each sentence. In this study, text data was analyzed using these approaches to explore new things from the dataset.

3.3.1 Topic modelling approach

As defined by Hornik and Grün (2011), the topic Modeling approach is a systematic method for classifying items that are present in a written document and extracting hidden patterns from a text corpus [38]. This technique is widely applied for tasks such as feature selection, document clustering, and information extraction from unorganized data. Text in a document can be categorized into distinct topics using the Latent Dirichlet Allocation (LDA), which is an illustration of a topic modeling. The coherence score for latent Dirichlet allocation (LDA) is displayed in Figure 18 below, which is based on the topic count.

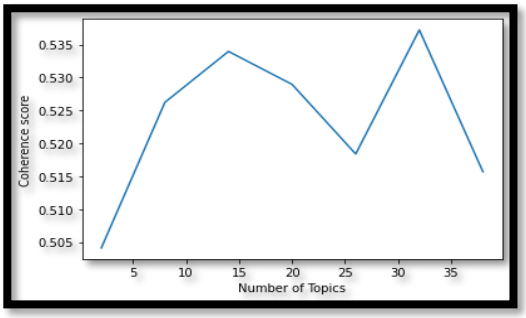


Figure-18: Coherence score for latent Dirichlet allocation (LDA)

We use the "corpora. Dictionary ()" function to create a dictionary of documents. The "Dictionary ()" function iterates through each document, giving each unique token a unique id and counting them. Word Cloud Visualization (WCV) has been developed, showing the dataset's most frequently occurring words in the word cloud. The word's visual representation is an important way to distinguish news categories and derive insightful intuitions. Nevertheless, Figure 19 provides a detailed illustration of Word Cloud, highlighting the dominant words and their importance.

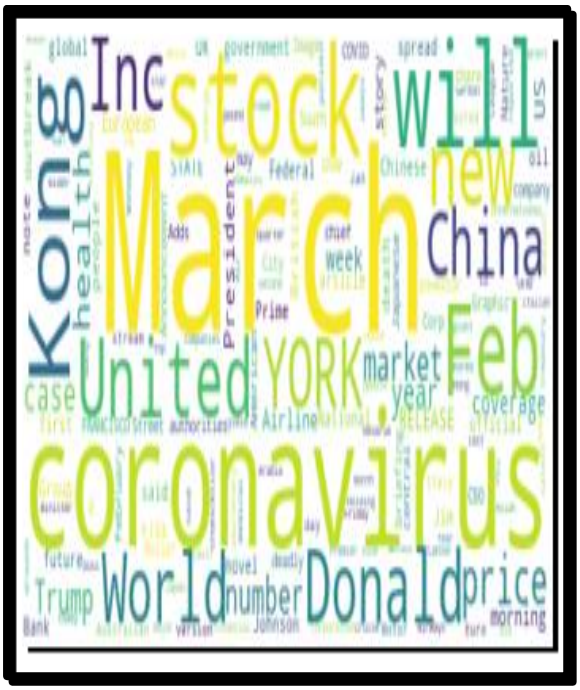


Figure-19: Word Cloud for News classification

4 Conclusion

To protect people's safety in the framework of COVID-19, it is essential to implement measures that make sure the spreading of precise information online. This study employs a predictive model-based utilize the fast-test pre-trained bidirectional long short-term memory (Bi LSTM) with an accuracy of 94% in classifying COVID-19 news. Additionally, the LDA technique is applied to reveal themes and detect hidden patterns within the dataset. The study reveals the deep learning method has lower accuracy compared to the machine learning method for text classification. Although the RF algorithm has achieved 98% accuracy, all method accuracy is lower compared to all deep learning method accuracy. However, Bi-LSTM with fast-test method perform better among all deep learning methods for text classification. Our results indicate that the Bi-LSTM + FastText model outperforms several traditional text classification models, such as SVM and RF, in terms of accuracy and robustness. This performance is consistent with findings in SOTA literature [39], where Bi-LSTM has been shown to capture long-term dependencies in text data more effectively than traditional models. The success of the Bi-LSTM can be attributed to its ability to process information from both the past and future context. This feature is particularly advantageous in sentiment analysis tasks where context is crucial. FastText is well known for its efficient handling of subword information, and also contributes to the performance by capturing richer semantic representations of words, even when out-of-vocabulary terms appear. This method embeddings overcome the limitations of traditional models (such as word2vec or GloVe) face when dealing with out-of-vocabulary (OOV) words. Different models rely deeply on predefined features (e.g., bag-of-words, TF-IDF). Our approach leverages word embeddings from FastText, allows the model to learn features directly from data. The most of research only perform the Bi-LSTM or other machine learning model without embedding or feature extraction technique or hyper parameter technique. In this paper, we incorporate all of these techniques for the classification of text analysis. Our findings show that embedding these techniques all together enhance performance. However, the difference in feature extraction methods could explain the differences in performance between our results and models that rely on more manual feature engineering. Variations in hyper parameter tuning may also explain performance differences. While our study provides a basic hyper parameter grid search, and a more extensive search or fine-tuning of layer sizes, learning rates, and regularization strategies may improve further performance. Overall, the machine learning model is the best classifier for text data classification.

5 Limitations

In this paper, a new machine learning and deep learning methods for classifying COVID-19 related electronics news were provided. While our analysis of the associated experimental data to demonstrate the suggested solution. Inaccurate news classification yielded a decent result for our trials. Nevertheless, this study has several shortcomings such as the dataset that used in this research is modest size in comparison to COVID 19 related text data. Our dataset contains lacks noun, verb, adjective, and other information since it does not use Part of Speech (POS) tags. During text pre-processing, we employed the lemmatization strategy, which takes a lot of time and results in poorly tuned hyperparameters on the BI-LSTM model. Despite these limitations, this study highlights the effectiveness of combining machine learning and deep learning methods for text classification and provides a foundation for future research to address these challenges.

References

[1] Jin, H., & Zhang, L. (2024, September). Text classification and its application based on natural language processing and machine learning. In *AIP*

Conference Proceedings (Vol. 3131, No. 1). AIP Publishing. https://doi.org/10.1063/5.0230295

[2] Didi, Y., Walha, A., & Wali, A. (2022). COVID-19 tweets classification based on a hybrid word embedding method. *Big Data and Cognitive Computing*, *6*(2), 58. https://doi.org/10.3390/bdcc6020058

[3] Tiwari, S., Chanak, P., & Singh, S. K. (2022). A review of the machine learning algorithms for COVID-19 case analysis. *IEEE Transactions on Artificial Intelligence*.https://doi.org/10.1109/TAI.2022.3142241

[4] Abdeen, M. A., Hamed, A. A., & Wu, X. (2021). Fighting the COVID-19 Infodemic in News Articles and False Publications: The NeoNet Text Classifier, a Supervised Machine Learning Algorithm. *Applied Sciences*, *11*(16),7265. https://doi.org/10.3390/app11167265

[5] Koirala, A. (2020). COVID-19 fake news classification with deep learning. *Preprint*, *4*. https://www.researchgate.net/profile/Abhishek-Koirala/publication/344966237_COVID-19_Fake_News_Classification_with_Deep_Learning/links/5f9b6ba5299bf1b53e5130b8/COVID-19-Fake-News-Classification-with-Deep-Learning.pdf

[6] Ravichandran, B. D., & Keikhosrokiani, P. (2023). Classification of Covid-19 misinformation on social media based on neuro-fuzzy and neural network: A systematic review. *Neural Computing and Applications*, *35*(1),699-717. https://doi.org/10.1007/s00521-022-07797-y(0123456789().,-volV) (0123456789().,-volV)

[7] Chughtai, M. A., Hou, J., Long, H., Li, Q., & Ismail, M. (2021, November). Design of a predictor for COVID-19 misinformation prediction. In *2021 International Conference on Innovative Computing (ICIC)* (pp.1-7). IEEE. https://doi.org/10.1109/ICIC53490.2021.9693057

[8] Arbane, M., Benlamri, R., Brik, Y., & Alahmar, A. D. (2023). Social media-based COVID-19 sentiment classification model using Bi-LSTM. *Expert Systems with Applications*, *212*, 118710. https://doi.org/10.1016/j.eswa.2022.118710

[9] Mansoor, M., Ur Rehman, Z., Shaheen, M., Khan, M. A., & Habib, M. (2020). Deep learning based semantic similarity detection using text data. *Information Technology and Control*, 49(4), 495-510. https://doi.org/10.5755/j01.itc.49.4.27118

[10] Samuel, J., Ali, G. M. N., Rahman, M. M., Esawi, E., & Samuel, Y. (2020). Covid-19 public sentiment insights and machine learning for tweets classification. *Information*, 11(6),314. https://doi.org/10.3390/info11060314

[11] Hossain, M. M., Asadullah, M., Tamanna, S., Tazwar, M. A. S., Alam, M. M., Hossain, M. A., Islam, M. & Sumy, M. S. A. (2024). Automated Machine Learning Algorithms for Predicting Anxiety and Depression in Bangladeshi University Students. *Journal of Information Systems Research and Practice*, 2(3), 16-31. https://mojc.um.edu.my/index.php/JISRP/article/view/54235

[12] Dangi, D., Dixit, D. K., & Bhagat, A. (2022). Sentiment analysis of COVID-19 social media data through machine learning. *Multimedia Tools and Applications*, *81*(29),42261-42283. https://doi.org/10.1007/s11042-022-13492-w:

[13] Ghasiya, P., & Okamura, K. (2021). Investigating COVID-19 news across four nations: A topic modeling and sentiment analysis approach. *Ieee Access*, *9*, 36645-36656. 10.1109/ACCESS.2021.3062875

[14] Madani, Y., Erritali, M., & Bouikhalene, B. (2021). Fake News Detection Approach Using Parallel Predictive Models and Spark to Avoid Misinformation Related to Covid-19 Epidemic. In *Intelligent Systems in Big Data, Semantic Web and Machine Learning* (pp. 179-195). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-72588-4_13

[15] Malla, S., & Alphonse, P. J. A. (2021). COVID-19 outbreak: An ensemble pre-trained deep learning model for detecting informative tweets. *Applied Soft Computing*, *107*,107495. https://doi.org/10.1016/j.asoc.2021.107495

[16] Qasim, R., Bangyal, W. H., Alqarni, M. A., & Ali Almazroi, A. (2022). A fine-tuned BERT-based transfer learning approach for text classification. *Journal of healthcare engineering*, *2022*. https://doi.org/10.1155/2022/3498123

[17] Khadhraoui, M., Bellaaj, H., Ammar, M. B., Hamam, H., & Jmaiel, M. (2022). Survey of BERT-based models for scientific text classification: COVID-19 case study. *Applied Sciences*, *12*(6), 2891. https://doi.org/10.3390/app12062891

[18] Shahi, T. B., Sitaula, C., & Paudel, N. (2022). A hybrid feature extraction method for Nepali COVID-19-related tweets classification. *Computational Intelligence and Neuroscience*, *2022*. https://doi.org/10.1155/2022/5681574

[19] Ahmed, M., Hossain, M. S., Islam, R. U., & Andersson, K. (2022). Explainable Text Classification Model for COVID-19 Fake News Detection. *Journal of Internet Services and Information Security (JISIS)*, *12*(2), 51-69. DOI:10.22667/JISIS.2022.05.31.051

[20] Rabby, G., & Berka, P. (2023). Multi-class classification of COVID-19 documents using machine learning algorithms. *Journal of Intelligent Information Systems*, *60*(2), 571-591. https://doi.org/10.1007/s10844-022-00768-8

[21] Etaiwi, H. A. (2022). Empirical Evaluation of Machine Learning Classification Algorithms for Detecting COVID-19 Fake News. *International Journal of Advances in Soft Computing & Its Applications*, *14*(1). DOI: 10.15849/IJASCA.220328.04

[22] Malla, S., & Alphonse, P. J. A. (2022). Fake or real news about COVID-19? Pretrained transformer model to detect potential misleading news. *The European Physical Journal Special Topics*, *231*(18), 3347-3356. https://doi.org/10.1140/epjs/s11734-022-00436-6

[23] Ravichandran, B. D., & Keikhosrokiani, P. (2023). Classification of Covid-19 misinformation on social media based on neuro-fuzzy and neural network: A systematic review. *Neural Computing and Applications*, *35*(1), 699-717. https://doi.org/10.1007/s00521-022-07797-y

[24] Gams, M., & Kolenik, T. (2021). Relations between electronics, artificial intelligence and information society through information society rules. *Electronics*, 10(4), 514. https://doi.org/10.3390/electronics10040514

[25] Janko, V., Slapničar, G., Dovgan, E., Reščič, N., Kolenik, T., Gjoreski, M., ... & Luštrek, M. (2021). Machine learning for analyzing non-countermeasure factors affecting early spread of COVID-19. *International Journal of Environmental Research and Public Health*, 18(13), 6750. https://doi.org/10.3390/ijerph18136750

[26] Hossain, M. M., Anselma, L., & Mazzei, A. (2023). Exploring sentiments in summarization: SentiTextRank, an Emotional Variant of TextRank. In CEUR WORKSHOP PROCEEDINGS (Vol. 3596, pp. 1-5). CEUR-WS. https://hdl.handle.net/2318/1950757

[27] Caldo, D., Bologna, S., Conte, L., Amin, M. S., Anselma, L., Basile, V., ... & De Nunzio, G. (2023). Machine learning algorithms distinguish discrete digital emotional fingerprints for web pages related to back pain. *Scientific Reports*, 13(1), 4654. https://doi.org/10.1038/s41598-023-31741-2

[28] Spirovski, K., Stevanoska, E., Kulakov, A., Popeska, Z., & Velinov, G. (2018, June). Comparison of different model's performances in task of document classification. In *Proceedings of the 8th international conference on web intelligence, mining and semantics* (pp. 1-12). https://doi.org/10.1145/3227609.322766

[29] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543). https://aclanthology.org/D14-1162.pdf

[30] Sparck Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, *28*(1), 11-21. https://doi.org/10.1108/eb026526

[31] Jaderberg, M., Simonyan, K., Vedaldi, A., & Zisserman, A. (2016). Reading text in the wild with convolutional neural networks. *International journal of computer vision*, *116*, 1-20. https://doi.org/10.1007/s11263-015-0823-z

[32] LeCun, Y., Touresky, D., Hinton, G., & Sejnowski, T. (1988, June). A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school* (Vol. 1, pp. 21-28). https://www.researchgate.net/profile/Yann-Lecun/publication/2360531_A_Theoretical_Framework_for_Back-Propagation/links/0deec519dfa297eac1000000/A-Theoretical-Framework-for-Back-Propagation.pdf

[33] Scherer, D., Müller, A., & Behnke, S. (2010, September). Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks* (pp. 92-101). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-15825-4_10

[34] Wang, W., Zhu, D., Wang, X., Hu, Y., Qiu, Y., Wang, C., ... & Scherer, S. (2020, October). Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4909-4916). IEEE. 10.1109/IROS45743.2020.9341801

[35] Lu, G., Businger, M., Dollfus, C., Wozniak, T., Fleck, M., Heroth, T., ... & Lipenkova, J. (2023). Agenda-setting for COVID-19: A study of large-scale economic news coverage using natural language processing. *International Journal of Data Science and Analytics*, *15*(3), 291-312. https://doi.org/10.1007/s41060-022-00364-7

[36] ADEMI, A. (2016). EVALUATION OF THE MODELS USED TO CREATE VECTOR SPACE REPRESENTATION OF WORDS. *SCIENCE, INNOVATION NEW Technology*, 31.

[37] Iwayama, M., & Tokunaga, T. Department of Computer Science Tokyo Institute of Technology. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e10595e5fa4c94df2ceb8867327ad2fa0825c089

[38] Grün, B., & Hornik, K. (2011). topicmodels: An R package for fitting topic models. *Journal of statistical software*, *40*, 1-30. https://www.jstatsoft.org/article/view/v040i13

[39] Klaiber, M. (2021). A fundamental overview of sota-ensemble learning methods for deep learning: a systematic literature review. *Science in Information Technology Letters*, *2*(2), 1-14. https://pubs2.ascee.org/index.php/sitech/article/view/549

Damage Identification of Prestressed Concrete Components Based on Machine Learning Optimization Algorithm and Piezoelectric Wave Measurement

Hongtao Zhu[1], Shuyun Guo[2*]
[1]School of Civil Engineering, Xinyang College, Xinyang 464000, China
[2]School of Foreign Languages, Gushi Vocational Education Center, Xinyang 464000, China
E-mail: zhuhongtao2008hi@163.com, 15713761803@163.com
[*]Corresponding author

Keywords: GA, BP, SVM, concrete, damage

Received: October 24, 2024

Prestressed concrete components have high resistance to cracks and stiffness, yet prone to damage leading to accidents under adverse environments and extreme loads. The study uses machine learning algorithms to construct an intelligent concrete damage recognition model aimed at accurately assessing its health status. The piezoelectric wave measurement method is used to collect small wave signals from concrete. The improved back propagation network is used to identify concrete damage characteristics in the signals, and the support vector machine is taken to correct the identification results. The genetic algorithm is used to optimize the back propagation neural network, obtaining the optimal threshold and weight of the back propagation neural network to improve its robustness and feature extraction ability in noisy environments. According to the results, the model constructed by integrating two classification algorithms has a mean square error of 7.962×10-4, a coefficient of determination of 0.9756, and an F1 score of 0.9836 in the damage location recognition results. In the identification results of the degree of damage, the mean square error of the research model was 6.548×10-2, the coefficient of determination was 0.9531, and the F1 score was 0.9925, respectively. In the environment with introduced noise, the recognition accuracy of the research model was 93.7%. The results indicate that the research method has higher accuracy and robustness in damage identification compared with other models, which can be used for concrete damage detection in large buildings or long-term high load buildings.

Povzetek: A hybrid GA-BP-SVM model enhances damage identification in prestressed concrete using piezoelectric wave measurement, achieving high accuracy (F1: 0.9925) and robustness (93.7% under noise), improving structural health monitoring.

1 Introduction

Concrete materials have abundant raw materials, low prices, simple production processes, durability, and strong plasticity, which are widely used in infrastructure construction and building construction. With the large-scale and diversified development of civil engineering, concrete may be subjected to strong external pressure or long-term corrosion, causing damage and leading to structural safety in buildings [1]. However, the early characteristics of concrete damage are not obvious. It is difficult to fully cover them through regular manual inspections, which cannot provide accurate results for assessing its health status. The piezoelectric wave measurement method uses piezoelectric materials to generate and detect waves. The instrument is small, easy to use, and not affected by the environment. It can be used for monitoring and warning, improving structural safety. The piezoelectric wave measurement method combined with machine learning algorithms can adaptively detect concrete damage, with high efficiency and accuracy. Back Propagation (BP) network can handle complex nonlinear problems and automatically extract patterns between data. It has high adaptability and self-learning ability, which is a commonly used feature classification algorithm. Genetic Algorithm (GA) simulates the natural selection process and has an automatic elimination mechanism. Through selection, hybridization, and mutation, the population evolves gradually towards the optimal solution. It is a meta-heuristic algorithm used for optimization. Support Vector Machine (SVM) is an extensively applied supervised learning algorithm. SVM has high accuracy in classification and regression tasks, especially in high-dimensional spaces, which has high generalization ability. Many scholars have conducted relevant research on piezoelectric wave measurement methods, GA, BP, and SVM algorithms.

In response to the inaccurate measurement of explosion shock wave pressure under the instantaneous high temperature effect in the explosion field, Shi et al. used piezoelectric wave sensors to analyze the effects of environmental temperature and transient temperature. A theoretical analysis method for transient temperature was proposed. A transient temperature control strategy was designed by coating 0.5 mm thick lubricating silicone oil on the sensor membrane and 0.2 mm thick glass fiber cloth on the sensor side. The accuracy of the explosion shock wave pressure was improved to 97.8% [2]. To accurately predict whether the surface roughness of

precision manufactured aluminum alloys meets the requirements, Bai et al. designed a BP prediction method. The results showed that the predictive progress of the model was improved by 7.8%. BP recognized small features in high-precision manufacturing and solved the insufficient accuracy [3]. In response to the inaccurate prediction of the self-diffusion coefficient in pure liquids, Zeng et al. used the BP to establish a nonlinear method that predicted the self-diffusion coefficient of pure liquids at normal pressure. The R2, AARD, and RMSE for predicting the self-diffusion coefficient were 0.9940, 7.09%, and 0.1106, respectively [4]. Jiang et al. used an optimized GA to plan the optimal layout of the clothing production line for the uneven work intensity among employees. The results showed that the balance rate of the production line increased from 70.5% to 97.05%, and the production cycle was shortened by 32.80%, verifying the efficiency improvement performance [5]. To solve the difficulty in early warning of nonlinear macroeconomic systems, Yin et al. proposed a CNN economic early warning system based on the IGA-BP. The correlation coefficient was 0.89, and the delay number was 0. The economic warning system based on IGA-BP algorithm was effective. The BP optimized by GA could improve its feature classification ability [6]. To optimize the ability to prevent and control large-scale crop diseases, Gangadevi et al. proposed an improved SVM plant disease and pest identification model. The recognition accuracy of the research model reached 91.1%, which was superior to other models. The powerful feature classification ability of SVM could solve the crop disease prevention and control [7]. Dong et al. proposed a runoff prediction model based on SOA-SVM to address the difficult runoff prediction caused by its nonlinear and non-stationary characteristics. The results indicated that the average error and RMSE indicators of the research model were superior to other models [8]. Zhang et al. designed a method for automatically extracting damage characteristics from point cloud data to address the inaccurate damage detection in reinforced concrete structures. The results indicated that the research method was an effective approach for post disaster impact assessment and large-scale building damage detection [9].

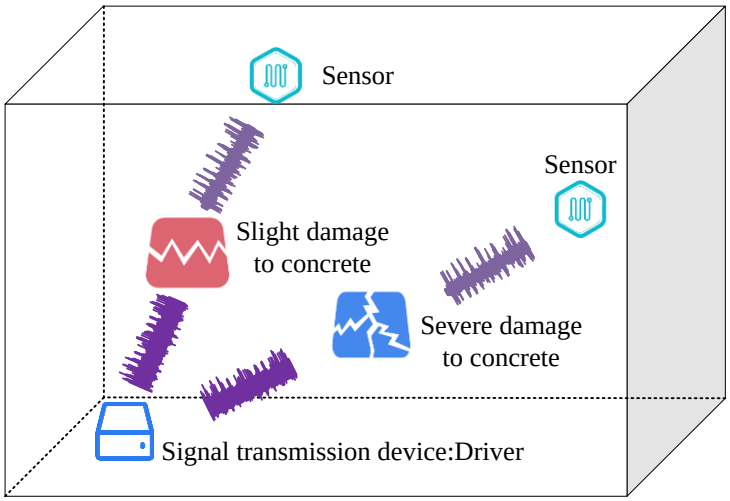


Figure 1: The principle of piezoelectric wave measurement

The above research indicates that existing detection methods have insufficient ability to detect small cracks and other damages in concrete structures in the early stages. Although existing literature has made significant progress in the application of machine learning and optimization algorithms, relatively little research has been done in the field of damage identification of prestressed concrete members, especially in the combination of piezoelectric wave measurement and GABP-SVM model. Existing methods often do not perform well when dealing with noise and uncertainty in complex environments. The piezoelectric wave measurement method can identify extremely small wave changes, providing a data basis for damage detection of concrete components. GA can be applied to optimize the BP for extracting damage features from piezoelectric wave measurement data. Therefore, introducing SVM algorithm for feature classification can improve the accuracy of damage detection in prestressed concrete components. The research aims to combine piezoelectric wave measurement method with GA, BP, and SVM algorithms to construct a new intelligent damage identification model for prestressed concrete components, monitoring the health status of concrete.

The research is structured from three sections. The first introduces related algorithms and mechanisms in the concrete damage recognition model based on GABP-SVM. The second section tests the model. The third section summarizes the research results. The latest research and comparison of research results in this field are shown in Table 1.

Table 1: Comparison chart of SOTA and research achievements in this field

Algorithms	Year	Researcher	Method	MSE	R2	F1	Prediction accuracy	SOTA lacks
Optimize BP	2023	Bai et al[3]	BP	/	/	/	/	Traditional BP may fall into local optima
	2022	Zeng et al[4]	BP-ANN	/	0.9940	/	/	Not tested for robustness in complex environments
	2022	Yin et al[6]	IGA-BP	/	/	/	/	Robustness unverified
	/	This study	Improved GA-BP	/	0.9235	0.00724	/	/
Research model	2024	Gangadevi et al[7]	FOA-SVM	/	/	0.945	91.1%	Unoptimized FOA may obtain non optimal solutions
	2023	Dong et al[8]	SOA-SVM	/	/	/	/	Robustness unverified
	2022	Zhang et al[9]	Point cloud data	/	/	/	/	Low efficiency and unstable accuracy
	/	This study	GA-BP-SVM	6.548×10-2	0.9531	0.9836	93.7%	/

2 Methods and materials

The study first introduces the principle of electromagnetic wave dynamic measurement method. Then, the GA is taken to optimize BP to improve the BP algorithm to identify concrete damage characteristics from piezoelectric wave measurement data. Finally, SVM is used to further classify and modify the output of GABP, improving the accuracy of damage detection in prestressed concrete components.

2.1 Design of concrete damage feature classification algorithm based on GABP

Under earthquake, high load and other conditions, cracks and other damages may occur on the surface or inside of prestressed concrete components. When cracks are small or damage occurs inside concrete components, they are usually difficult to detect. If not repaired timely, it may lead to major accidents such as structural collapse. Piezoelectric smart materials have positive/negative piezoelectric effects. The positive piezoelectric effect is manifested by the internal polarization when subjected to external forces, releasing charges proportional to the pressure. The negative pressure electric effect is manifested in the external electric field, where materials convert electrical energy into mechanical energy and undergo deformation [10-11]. Therefore, piezoelectric smart materials can be used as signal sensors, which are taken as signal transmission device in concrete structure damage detection. The principle of using piezoelectric intelligent materials for piezoelectric wave measurement is shown in Figure 1.

In Figure 1, the signal transmission device deforms under the action of an electric field to generate sinusoidal stress waves, which propagate inside the concrete. During the propagation process, the signal will gradually decay due to energy loss caused by friction, scattering, and crack absorption between the medium and particles.

Therefore, there is the energy conservation $E_I = E_H + E_R + E_T$. E_I is the energy of stress waves, which is the initial stress wave energy. E_H represents energy loss, which refers to the energy lost during the propagation of stress waves in concrete due to friction, scattering, and absorption. E_R is the energy of the transmitted wave received by the sensor after passing through the concrete. E_T is the reflected wave energy, which refers to the energy reflected back by stress waves when they encounter damage or interfaces inside concrete. Therefore, by collecting signals through sensors, concrete damage can be monitored. There is much interference noise in the signals collected by sensors. Many options for dimensionality reduction can preserve the main information. Singular value decomposition can select the k largest singular values to explain most of the variance in the signal, preserving the main information. Therefore, the study uses singular value decomposition to reduce the dimensionality of the signals, set the threshold of cumulative variance contribution to 90%, and introduces the BP algorithm to identify the characteristics of concrete damage information in the signals. The BP is displayed in Figure 2.

In Figure 2, during the forward propagation process, initialization is first performed, and then the sample data enters the hidden layer for calculation using a transfer function. The result is then fed to the output layer, where the error and calculation result are calculated. Finally, the termination condition is reached and the prediction result is output. During the iteration process, if the termination condition is not satisfied, BP is performed, and updated the obtained training error to all neurons. Each neuron adjusts the weights and thresholds of the entire network on the basis of the training error. If the training frequency reaches the set condition or the error reaches the

minimum, the neural network ends. Otherwise, it enters the hidden layer to continue calculation until the condition is met. The forward propagation is shown in equation (1) [12].

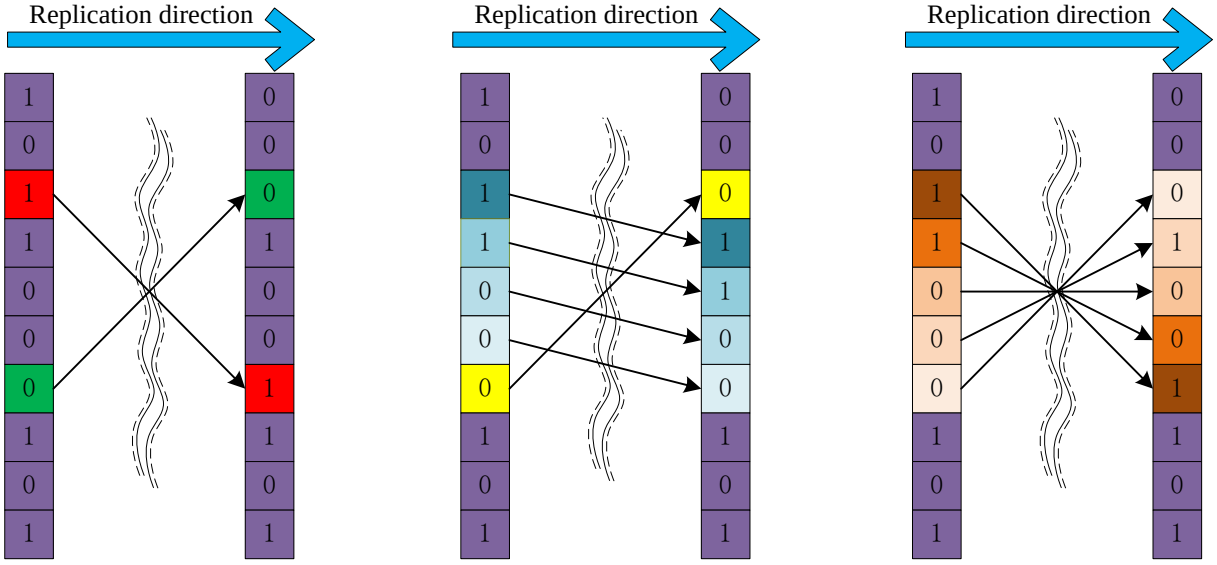
$$\begin{cases} r_q = f_1(H_{R\varphi}, E), \varphi = 1,2,...,l \\ o_s = f_2(H_{\varphi s}, A), s = 1,2...,N \end{cases} \tag{1}$$

In equation (1), r_q is the output of the hidden layer. o_s is the output of the output layer. E is the input sample, A is the output of the hidden layer. N is the sample size. $H_{R\varphi}$ and $H_{\varphi s}$ represent the weight matrices of the hidden layer and the output layer. f_1 is the activation function of the hidden layer. f_2 is the activation function of the output layer. φ is the neuron index of the hidden layer. s is the neuron index of the output layer. The training error is shown in equation (2).

$$J = \frac{1}{2}\sum_{\varphi=1}^{N}\sum_{s}^{m}(i_{ns} - o_{ns})^2 \tag{2}$$

In equation (2), m satisfies the output layer node. n satisfies the input layer node. i_{ns} satisfies the expected output of the network. J represents the training error. $\frac{1}{2}$ is the coefficient used in error calculation. i_{ns} is the expected output value of the network for the i-th output node and the s-th sample. o_{ns} is the output value of the n-th neuron in the output layer for the s-th sample. The updated network weights are shown in equation (3).

$$\begin{cases} \varpi_{\vartheta s}^{\upsilon+1} = \varpi_{\vartheta s}^{\upsilon} + \psi\sigma_k^K r_q, s = 1,2,...,S; d = 1,2,...,L \\ \varpi_{g\vartheta}^{\upsilon+1} = \varpi_{g\vartheta}^{\upsilon} + \psi\sigma_d^L E_g, l = 1,2,...L; g = 1,2,...,G \end{cases} \tag{3}$$



(a) Exchange variation (b) Insertion variation (c) Flashback variation

Figure 3: Genetic algorithm variation operation

In equation (3), $\varpi_{\vartheta s}^{\upsilon+1}$ represents the weight matrix of the $\upsilon+1$-th iteration from the hidden layer ϑ to the output layer s. $\varpi_{\vartheta s}^{\upsilon}$ is the weight matrix before the update. S is the maximum number of output layers. σ_k^K is the specific layer weight update for the k-th iteration. K is the total number of iterations. $\varpi_{g\vartheta}^{\upsilon}$ is the weight matrix from the input layer g to the output layer ϑ during the υ-th iteration. $\varpi_{g\vartheta}^{\upsilon+1}$ is the updated weight matrix. G is the total number of input layers. σ_d^L is the weight update of the output layer for the l-th iteration. L is the maximum number of iterations. ψ satisfies the learning rate. υ is the maximum number of iterations. r_q is the output of the hidden layer in the q-th iteration. E_g is the input sample in the g-th iteration. Due to the BP over-fitting in noisy environments, there are numerous heuristic algorithms that can be used to optimize BP, among which GA exhibits high stability and robustness when dealing with complex problems, and can better cope with noise and uncertainty. Therefore, the study introduces GA to optimize BP. The GA evaluates the suitability of individuals based on the fitness function and the gap between them and the optimal target. Genetic methods such as selection, crossover, and mutation are performed to simulate natural selection of better individuals. Different fields will use different fitness functions. Generally, the fitness function of GA is transformed from the objective

function to the fitness function, as displayed in equation (4).

$$f(x)=\begin{cases} g(x) \\ -g(x) \end{cases} \tag{4}$$

In equation (4), when solving the minimization problem, the fitness function $f(x)=-g(x)$ is used. When solving the maximization problem, the fitness function is $g(x)$. When the fitness function is not directly transformed from the objective function, the fitness function to solve the minimization problem is shown in equation (5).

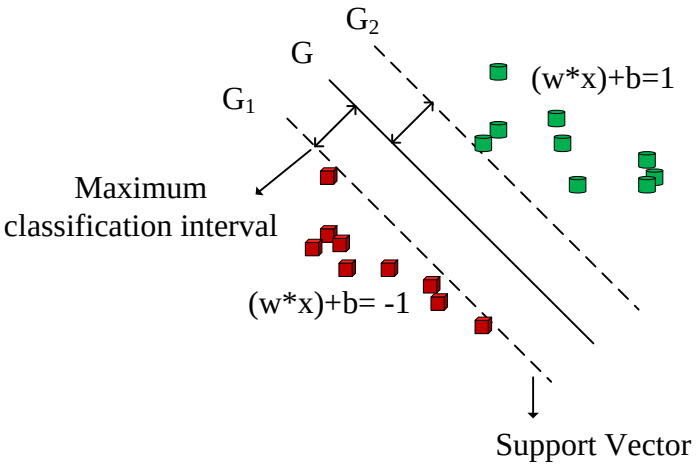


Figure 4: Optimal classification hyperplane for SVM

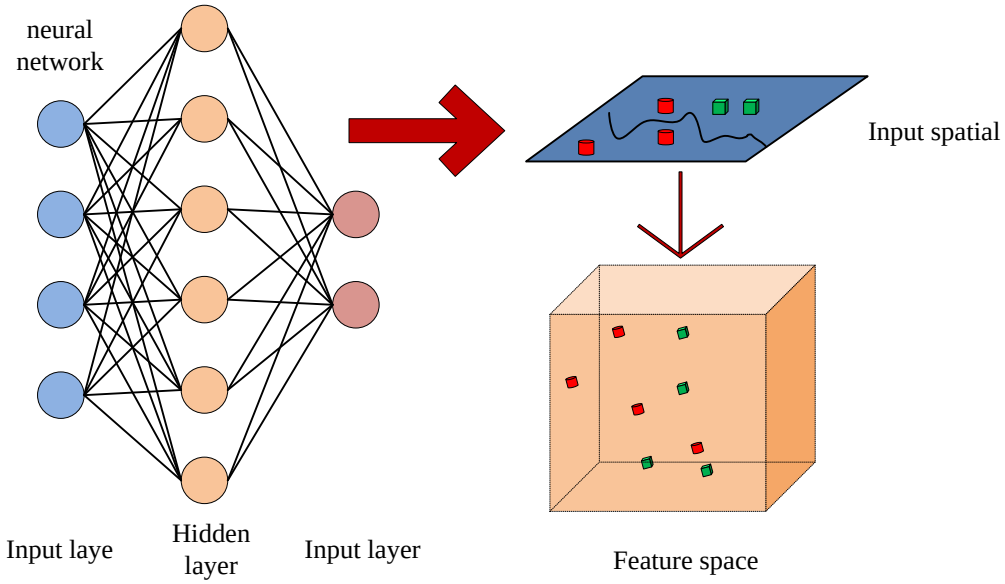


Figure 5: SVM network topology diagram

$$f(x)=\begin{cases} C_{max}-g(x), \quad g(x)<C_{max} \\ 0, \quad else \\ f(x)=\dfrac{1}{1+C+\mathrm{g}(x)}, \quad C\geq 0, \quad C+g(x)\geq 0 \end{cases} \tag{5}$$

In equation (5), C_{max} is the maximum fitness value. C is the fitness value. When solving the maximization problem, the fitness function is shown in equation (6).

$$f(x)=\begin{cases} g(x)-C_{min}, \quad g(x)<C_{max} \\ 0, \quad else \\ \dfrac{1}{1+c-g(x)}, \quad C\geq 0, \quad c-g(x)\geq 0 \end{cases} \tag{6}$$

In equation (6), C_{min} is the minimum fitness value. c is the fitness threshold. The core of GA operators is crossover and mutation, which bring the entire selection

process closer to the optimal solution. GA randomly generates new offspring in the population through crossover operations. When designing the crossover probability, excellent individuals should avoid cross pairing. This can avoid the disappearance of excellent individuals and ensure that new individuals are close to the optimal solution [13]. Similar to natural genetic rules, when there is only cross inheritance, the diversity of the population is insufficient, and individuals who are too similar can easily slow down or even stop the population evolution speed. Therefore, GA includes mutation mechanisms, which increase population diversity through mutation. The genetic mutation operation is displayed in Figure 3.

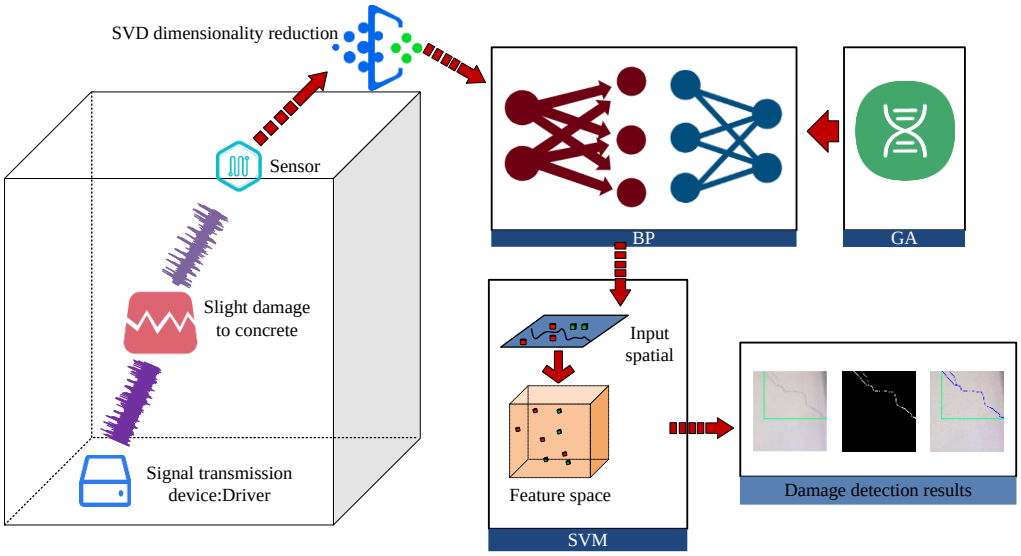


Figure 6: Damage identification model for prestressed concrete components based on GABP-SVM

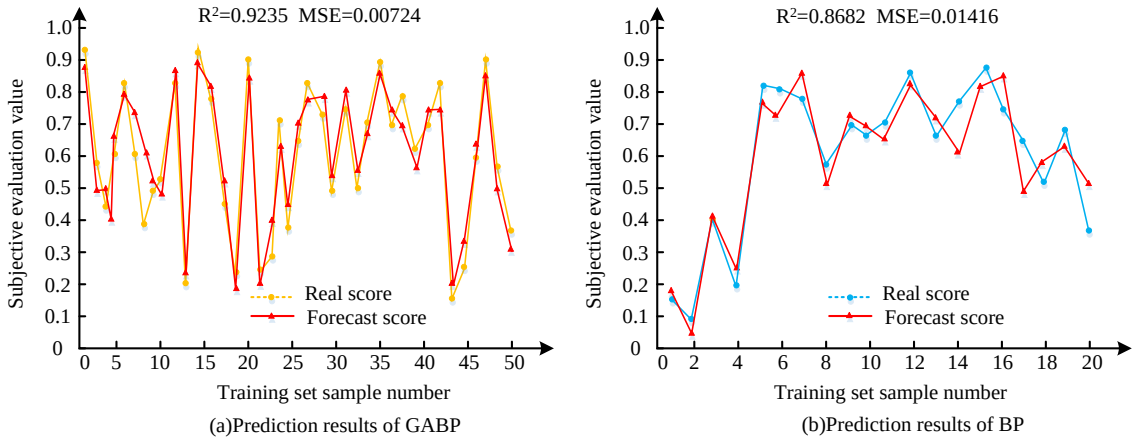


Figure 7: Comparative experimental results of BP and GABP

Figure 3 displays the mutation operation of binary encoding. Figure 3 (a), Figure 3 (b), and Figure 3 (c) depict the exchange, insertion, and flashback during the replication process. Genetic variation produces new individuals, which determines the local search of GA and can bring diversity to the population. These three genetic operators of GA simulate the recombination and mutation process of sexual reproduction, and select better individuals through fitness scores to evolve the entire population towards the optimal solution. The BP optimized by GA is used to map the space vector from n -dimensional to m -dimensional, which can reduce the blindness in the weight adjustment process. The optimized weight correction is shown in equation (7) [14].

$$\begin{cases} w_{jk}(\alpha+1) = w_{jk}(\alpha) + \beta h_k E_r \\ w_{ij}(\alpha+1) = w_{ij}(\alpha) + \gamma i_k E_p \end{cases} \tag{7}$$

In equation (7), γ and β are learning factors, used to control the magnitude of weight adjustment. $w_{ij}(\alpha)$ and $w_{jk}(\alpha)$ satisfy the weights of each neuron in the hidden and output layers after α iterations. r is the sample size. E_p is the sample error, used to guide the

adjustment of weights. $w_{ij}(\alpha+1)$ and $w_{jk}(\alpha+1)$ are the weights of each neuron in the hidden layer and output layer after $\alpha+1$ iterations. h_k and i_k are the gradient terms for weight updates, used to minimize the error of the network. The new fitness function is shown in equation (8).

$$F = k\left(\sum_{p=1}^{m} asb\left(o_p - t_p\right)\right) \tag{8}$$

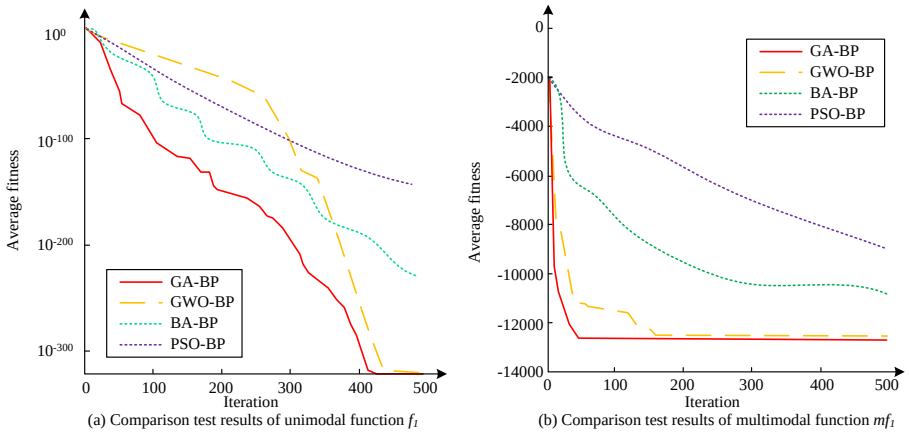
In equation (8), k is the coefficient. m is the number of output layer nodes. $o_p - t_p$ is the output error of the network. During the optimization process, floating-point encoding is used to encode the basic solution space. After encoding is completed, the population of the GA is initialized. The population M is shown in equation (9).

$$M = m \times n + m \times q + q + m \tag{9}$$

In equation (9), n is the input layer node size in BP. q is the hidden layer node size in BP. The probability p_i of an individual being selected in GA is shown in equation (10).

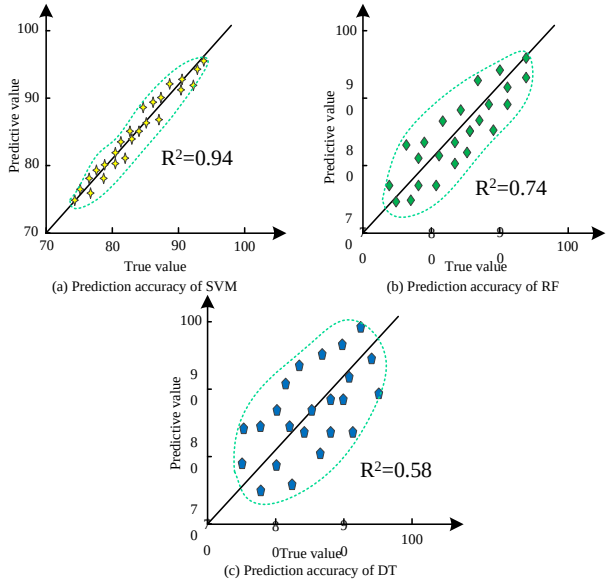
$$p_i = \frac{f_i}{\sum_{j=1}^{M} f_j} \tag{10}$$

In equation (10), $f_i = k / F_i$. F_i represents the fitness value of node i, calculated by equation (8). After determining the individual population F, individuals are decoded and corresponding network connection thresholds and weights are generated. Genetic operations are performed on individuals with lower F -values until the maximum iteration is satisfied to obtain the optimal threshold and weights for the BP network, ultimately optimizing the BP network.



(a) Comparison test results of unimodal function f_1 (b) Comparison test results of multimodal function mf_1

Figure 8: Comparison of optimization algorithms and experimental results



(a) Prediction accuracy of SVM (b) Prediction accuracy of RF (c) Prediction accuracy of DT

Figure 9: Comparison test results of SVM algorithm and classification algorithm

2.2 Construction of damage identification model for prestressed concrete components based on GABP-SVM

In practical applications, prestressed concrete components may have small cracks in the concrete due to factors such as insufficient compaction, natural shrinkage, and uncleared wood chips, which reduces the GABP feature recognition accuracy [15]. SVM is a supervised learning algorithm widely used in classification and regression analysis. The study introduces SVM to further classify GABP recognition results, aiming to improve the accuracy of concrete component damage detection. In classification problems, SVM attempts to find a hyperplane to maximize the boundary between two categories. The optimal classification hyperplane for SVM is shown in Figure 4.

In Figure 4, in two-dimensional space, the classification hyperplane can be imagined as a line that separates two categories. In a higher dimensional space, the optimal classification hyperplane becomes a hyperplane. SVM can train a classifier for concrete damage feature classification by collecting and analyzing various concrete damage data. The optimal classification plane and lines G, G1, and G2 in SVM algorithm are displayed in equation (11).

$$f(x) = w^T x + b \tag{11}$$

In equation (11), w satisfies the normal vector. b satisfies the bias amount. Among all the classified samples, (x_1, y_1), (x_2, y_2), ... and (x_i, y_i) need to satisfy equation (12).

$$y_i(wx_i + b) \geq 1 \tag{12}$$

In equation (12), different w and b can determine different position planes. The optimal classification plane calculation is shown in equation (13).

$$\begin{cases} \max \dfrac{1}{\| w \|} = \min \dfrac{1}{2} \| w \|^2 \\ s.t, y_i(wx_i + b) \geq 1, i = 1, 2, 3...n \end{cases} \tag{13}$$

In equation (13), $\min \dfrac{1}{2}\|w\|^2$ represents the minimum confidence range. Equation (13) can be converted into equation (14).

$$\begin{cases} \min \dfrac{1}{2} \| w \|^2 \\ s.t, y_i - wx_i - b \leq \varepsilon \\ wx_i - y_i + b \leq \varepsilon, i = 1, 2, 3..., n \end{cases} \tag{14}$$

In equation (14), $\min \dfrac{1}{2}\|w\|^2$ can be used as a prerequisite for the optimization problem. When the current condition cannot be met, relaxation variables ξ_i and ξ_i^* are introduced to relax the range of the premise conditions. At this time, the objective function is to minimize the confidence range, as shown in equation (15).

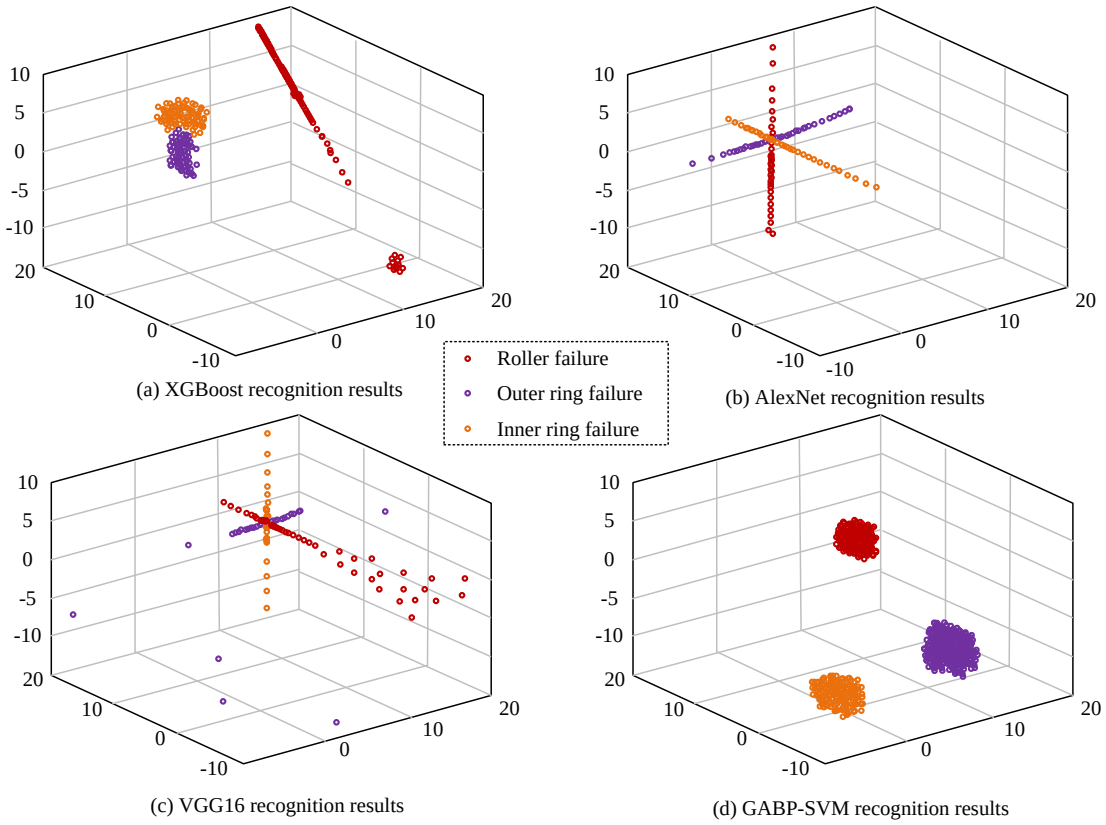
$$\begin{cases} \min \dfrac{1}{2} \| w \| + C \sum_{i=1}^{N} \left(\xi_i + \xi_i^* \right) \\ s.t, y_i - wx_i - b \leq \varepsilon + \xi_i \\ wx_i - y_i + b \leq \varepsilon + \xi_i^*, i = 1, 2, 3...n \\ \xi_i \geq 0 \\ \xi_i^* \geq 0 \end{cases} \tag{15}$$

In equation (15), C satisfies the penalty factor, which is the punishment for data that exceeds the ε range [16]. Through the above classification constraints, SVM has excellent ability in classification. The SVM is displayed in Figure 5.

In Figure 5, in neural network architecture, the input layer receives feature vectors from the original input of the data source, with each feature vector representing a sample. The hidden layer is composed of multiple neurons and connected to each input node in the input layer, forming a fully connected structure. The output layer is responsible for outputting classification results, and the number of neurons is usually equal to the number of categories in the classification problem. In the SVM structure, the input space refers to the space of the original input data, while the feature space refers to the space of the feature vectors obtained through hidden layer processing. SVM can map data from the input space to the feature space to complete classification tasks. The model structure for identifying damage in prestressed concrete components combining GABP and SVM is shown in Figure 6.

In Figure 6, the sensor of piezoelectric fluctuation measurement method identifies the fluctuation changes in prestressed concrete components. After collecting recognition data and conducting preliminary processing, it is input into a BP optimized by GA to identify the characteristics of concrete damage. Afterwards, the calculation results are fed into the SVM for further classification and correction, improving the accuracy of damage detection in prestressed concrete components. At the end, the visualization results of damage identification are output. A damage recognition method for prestressed concrete components is constructed on the basis of GABP-SVM. During the training, the original signal is first preprocessed. This includes removing the DC bias from the original signal to make the average value of the signal zero, and using a bandpass filter to remove high-frequency noise and low-frequency interference to preserve useful signal components. The most important

thing in the pre-processing process is to reduce dimensionality through SVD, retain the main singular values, and remove noise and redundant information. After that, a BP neural network is created and the dimensions of the input layer, hidden layer, and output layer are set. Then, GA is used to optimize the weights and thresholds of BP neural network to improve the performance of the model. Finally, SVM is used to classify the output of the optimized BP neural network to further improve the classification accuracy and complete the model training. Compared with the traditional SVM, this study uses SVD to denoise the data. Moreover, the powerful nonlinear modeling capability of the improved BP algorithm is combined with the linear classification performance of SVM. By fully utilizing the advantages of these two algorithms, the classification problem of complex data collected by sensors can be better solved. This method enables GABP-SVM to better perform damage detection tasks.



(a) XGBoost recognition results

(b) AlexNet recognition results

(c) VGG16 recognition results

(d) GABP-SVM recognition results

Figure 10: Comparison test results between GABP-SVM algorithm and other algorithms

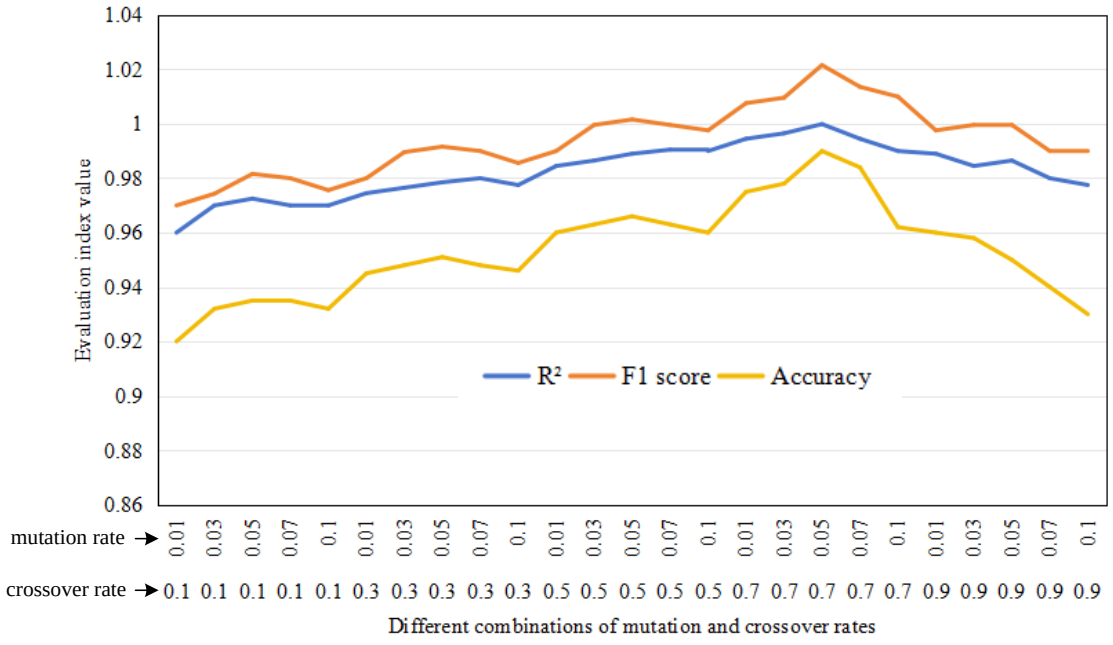
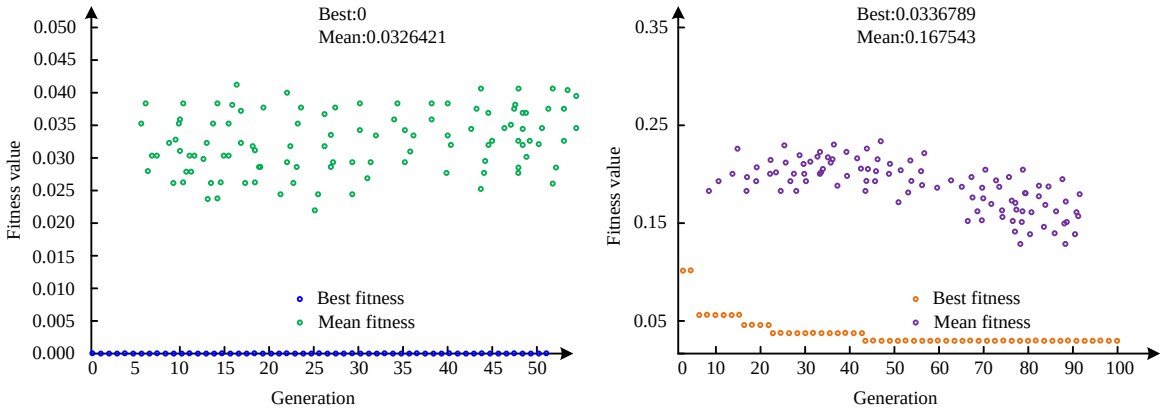
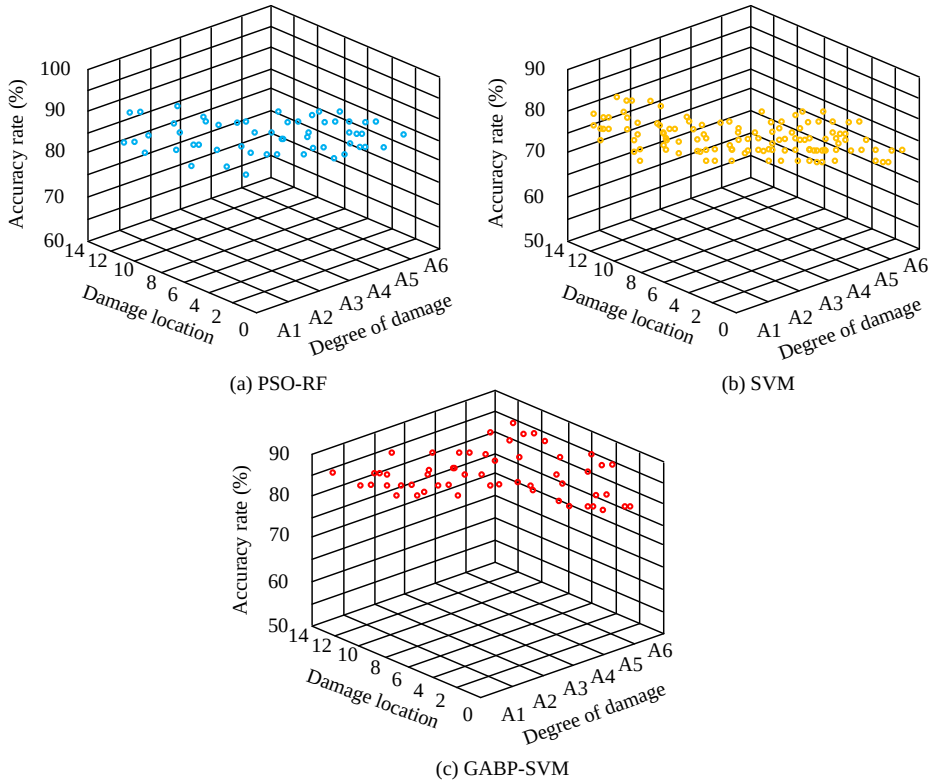


Figure 11: Comparison test results between GABP-SVM algorithm and other algorithms



(a) Damage location identification (b) Damage degree identification

Figure 12: Experimental results of training GABP



(a) PSO-RF (b) SVM

(c) GABP-SVM

Figure 13: Comparison of 4 algorithms for fault recognition in 3 parts

3 Results

To verify the damage identification model for prestressed concrete components based on GABP-SVM, relevant experiments are conducted. The experiment conducts performance tests on the optimization algorithm GA, feature classification algorithm GABP, and SVM used in the research, verifying the computational efficiency and classification accuracy. Afterwards, comparative tests are conducted with other concrete component damage identification models to verify the generalization ability and the damage identification performance.

3.1 Performance analysis of damage feature classification algorithm based on GABP-SVM

The study uses the piezoelectric wave measurement method to extract rich wave data information from prestressed concrete components. GA, BP, and SVM algorithms are used to identify damage characteristics. To verify the effect of BP optimized by GA, a comparative test is conducted between BP and GABP. The computer configuration used in the experiment is Intel ® CoreTM i9-9980XE, with an 8-core 2.1GHz CPU, 16GB of memory, and 1TB of hard drive. The

experiment selected 2,000 samples from the classic MNIST dataset for algorithm training and testing, with 80% of the dataset used for training and 20% for testing, as displayed in Figure 7.

In Figure 7 (a), the experimental results of the GABP algorithm on the testing set showed that the fitting degree between the GABP feature classification results and the true values was high. The R2 of the GABP was 0.9235, and the MSE was 0.00724. In Figure 7 (b), the BP algorithm on the testing set show that the fitting degree between the BP feature classification results and the true values was lower than that of GABP. The R2 of the BP was 0.8682, and the MSE was 0.01416. The results showed that the R2 value of GABP was 0.0553 higher than that of BP, and the MSE value of GABP was 0.00692 lower. The GABP has better feature classification ability than the basic BP algorithm. To verify the computational efficiency of the improved method, a comparative test is conducted between the GABP algorithm and three heuristic algorithms, namely Grey Wolf Optimization (GWO), Bat Algorithm (BA), and Particle Swarm Optimization (PSO), to optimize the BP algorithm, as displayed in Figure 8.

In Figure 8 (a), in the unimodal function, the average fitness of GABP dropped to the lowest at 417 iterations, and the average fitness of GWO-BP dropped to the lowest at 440 iterations. However, BA-BP algorithm and PSO-BP algorithm had slower iteration speeds than BA-BP and GWO-BP algorithms, and had not yet converged after 500 iterations, with average fitness values still at a high level. In Figure 8 (b), in multimodal functions, the average fitness value of the GABP algorithm dropped to the lowest after 48 iterations. The global search speed of the GWO-BP in the early stage was similar to that of the BA-BP algorithm, but the speed was slower in local fine search. The average fitness value of the GWO-BP algorithm dropped to the lowest after 174 iterations. The PSO-BP and BA-BP algorithms converged faster in multi-modal functions than in unimodal functions, but convergence was still incomplete after 500 iterations. The genetic operation of GA provides various ways to generate new solutions, increase the diversity of understanding, and help GA better explore the solution space. Therefore, GA has strong global search capabilities. Moreover, GA has strong parallel computing ability, which can significantly improve the convergence efficiency, so that the convergence efficiency of GA-BP is higher than that of PSO-BP and BA-BP. To verify the classification accuracy of SVM, a comparative test is conducted using Random Forest (RF), Decision Tree (DT), and SVM. 1,600 samples from the CIFAR-10 dataset were selected for algorithm training and testing, with 80% of the dataset used for training and 20% for testing. The results are shown in Figure 9.

In Figure 9 (a), the predicted value of SVM was close to the standard value, with a prediction accuracy of 0.94. In Figure 9 (b), the predicted value of the RF was far from the standard value, with a prediction accuracy of 0.74. In Figure 9 (c), compared with the standard value, the predicted value of DT was very discrete, and the distance between the predicted value and the standard value was farther than SVM and RF, with a prediction accuracy of 0.58.

3.2 Performance Analysis of GABP-SVM Algorithm

To verify the performance of the GABP-SVM algorithm, experimental tests are conducted. The study conducted comparative experiments with Extreme Gradient Boosting (XGBoost) in ensemble learning, Visual Geometry Group Network (VGG), and AlexNet Neural Network (AlexNet). The experiment selected 20,000 data samples from the Kesi West Reserve bearing dataset, with 80% of the dataset used for model training and 20% for model testing. Four algorithms were used in the test to identify and classify faults in the inner, outer, and rolling elements of bearings. The additive Gaussian white noise in $-8 \le SNR(db) \le 3$ was introduced as interference data. The experimental results are shown in Figure 10.

In Figure 10 (a), XGBoost algorithm identified the fault characteristics of the rolling elements, but the fault characteristics of the inner and outer rings were not accurately classified. In Figure 10 (b), AlexNet algorithm could recognize the inner, outer, and rolling element features, but did not complete the classification task for the three types of features. In Figure 10 (c), VGG16 algorithm did not complete the recognition and classification of fault types, and the recognition results had many discrete points. The accuracy and robustness of the model were both at a low level. In Figure 10 (d), GABP-SVM algorithm accurately identified the fault characteristics of the inner and outer rings and rolling elements of rolling bearings and completed the classification. The results show that in noisy environments, the feature recognition ability of GABP-SVM algorithm is superior to XGBoost, AlexNet, and VGG16 algorithms. The setting of crossover and mutation rates in GA will affect the optimization effect of parameters, thereby affecting the performance of the model. To analyze the impact of GA parameters on model performance, a sensitivity analysis is conducted. The study conducted tests different combinations of mutation rate and crossover rate, and the test results are shown in Figure 11.

In Figure 11, as the crossover rate increased, the performance of the model gradually improved. When the crossover rate was 0.7, the R2, F1 score, and accuracy of the model were all the highest. When the crossover rate was 0.9, the indicator values decreased. In different crossover rate values, the indicator value first increased and then decreased with the increase of mutation rate. When the crossover rate was 0.7 and the mutation rate was 0.05, the R2, F1 score, and accuracy were at their highest values, which was the optimal parameter setting for GA in GABP-SVM.

3.3 Performance analysis of damage identification model for prestressed concrete components based on GABP-SVM

The above experiment shows that the feature classification algorithm used in the study has high accuracy. To verify the performance of combining the two for damage feature recognition, a comparative experiment is conducted between the GABP-SVM model and concrete damage recognition models based on SVM and GABP, as well as a model that integrates PSO and RF. The experiment first trains GABP to obtain the optimal weights and thresholds of BP, as presented in Figure 12.

In Figure 12 (a), after 54 iterations, the calculated optimal fitness value was 0.0326421, and the optimal number of hidden layers was 9. In Figure 12 (b), after 100 iterations, the optimal fitness value calculated was 0.0336789, and the optimal hidden layer was 21. The BP is set as the optimal threshold and weights. A comparative experiment is conducted between the optimized GABP-SVM and other models mentioned above, as presented in Table 2.

In Table 2, the symbol "*" indicates a $P<0.05$ when compared to other algorithms, indicating that the difference in results is statistically significant. Among them, R2 is mainly used for regression analysis to determine the classification ability of the model. The range of R2 values is [0, 1], and the larger the R2, the higher the classification accuracy of the model. The Mean Squared Error (MSE), R2, and F1 scores of the GABP-SVM model in the damage location recognition results were 7.962×10-4, 0.9756, and 0.9836, respectively, which were superior to the SVM model, GABP model, and PSO-RF model. In the results of identifying the degree of damage, the MSE, R2, and F1 scores of the GABP-SVM model were 6.548×10-2, 0.9531, and 0.9925, respectively, which were also superior to other models. To further verify the robustness, comparative tests are conducted on the models that performed better in an environment with added noise. The test results are shown in Figure 13.

In Figure 13 (a), the PSO-RF model had an accuracy of 82.6% in identifying damage to prestressed concrete components. In Figure 13 (b), the accuracy of SVM model in identifying damage to prestressed concrete components was 69.8%. In Figure 13 (c), the accuracy of GABP-SVM model for damage identification of prestressed concrete components was 93.7%. The results indicate that in noisy environments, the damage recognition accuracy of the GABP-SVM model is significantly higher than that of the SVM and PSO-RF models. To further analyze the robustness of the research model under different damage scales, the experimental data is organized into three datasets: minor damage, moderate damage, and serious damage to test the model. The experimental results are shown in Table 3.

In Table 3, "*" represents a $P<0.05$ when compared to other algorithms, indicating that the difference in results is statistically significant. The recognition ability of GABP-SVM and PSO-RF models for serious, moderate, and minor damages showed a decreasing trend, while the decrease in GABP-SVM was smaller than that of PSO-RF. GABP-SVM maintained a high overall recognition level, with MSE, R2, and F1 scores of 0.0907, 0.5054, and 0.4062 for PSO-RF for minor damages, respectively. The results indicate that the GABP-SVM model exhibits higher robustness than the comparison model under different damage scales.

4 Discussion

Comparative experiments were conducted on the basic algorithms that make up the research model, and a damage identification model for prestressed concrete components based on GABP-SVM was verified. The R2 of the GABP feature recognition was 0.9235, and the MSE value was 0.00724, which was better than the BP algorithm. Jin et al. reached similar conclusions using GABP for rolling bearing fault diagnosis [17]. In the calculation of unimodal and multimodal functions, the GABP algorithm solved for the optimal solution that was closer to the true result after 417 and 48 iterations, respectively. This result was similar to the one obtained by Khatri et al. using GA to improve the load-bearing performance of fluid dynamic sliding bearings [18]. However, the research algorithm exhibits higher feature recognition ability and convergence efficiency compared to the latest advanced algorithms. The reason is that the GA is used to automatically optimize the parameters of the BP algorithm. In addition, the GA is improved to avoid getting stuck in local optima. The classification prediction accuracy of SVM algorithm was 0.94, which was better than RF algorithm and DT algorithm. This result was similar to the conclusion of the SVM-based radial deformation error evaluation method for turbine blades proposed by Chen et al [19]. In the model testing, the GABP-SVM model performed better than the SVM model, GABP model, and PSO-RF model in identifying the location and degree of damage. In the robustness test of the model with added noise, the accuracy of the GABP-SVM model in identifying damage to prestressed concrete components was 93.7%, which was 11.1% and 23.9% higher than the PSO-RF and SVM. Zhao et al. proposed similar conclusions in the research of concrete mesoscopic damage characteristics detection based on improved R-CNN [20]. However, the GABP-SVM model is more robust than the newly proposed model based on the improved R-CNN model. Also, due to the deep optimization of GA in this study, it avoids falling into local optimal solutions in complex environments, thus ensuring the stability of the model. The results indicate that the designed model not only has excellent accuracy in identifying damage to concrete components, but also has extremely high robustness. From this, the study uses RF to preprocess the data and GA to optimize the parameters of BP. The improved BP output results are input into SVM for further feature classification, greatly improving the recognition ability of concrete component damage and the generalization ability in complex concrete structures. Finally, the study analyzes

the computational complexity and scalability of the GABP-SVM model. The analysis results show that as the data size increases, the computational complexity does not significantly increase. In high-dimensional data, the GABP-SVM model can still maintain high feature recognition ability. The results indicate that the dimensionality reduction techniques introduced in the study and the parallel computing capability of the GA enable the GABP-SVM model to have higher computational efficiency and scalability.

5 Conclusion

Concrete is widely used in urban construction and industrial production. Concrete components are prone to damage and structural instability under conditions such as earthquakes, long-term high loads, and environmental corrosion. A smart concrete damage identification model was constructed by combining machine learning algorithms with the intelligent sensing effect of piezoelectric materials, aiming to accurately evaluate the health status of concrete. The study used piezoelectric wave measurement method to collect small wave signals from concrete. The GA was used to optimize BP to identify the characteristics of concrete damage from the signals. In addition, SVM was introduced to further classify and modify the recognition results of GABP, and a damage recognition model for prestressed concrete components based on GSBP-SVM was constructed. The designed model could accurately identify the damage location and degree of concrete, with high robustness. Concrete has extremely wide applications. Currently, there are a large number of buildings with complex structures and large volumes. In addition to prestressed concrete components, various forms of concrete components are different. Future research can focus on different forms of concrete components to broaden the applicability of the research model.

Funding

The research is supported by Key Scientific and Technological Research Projects of Henan Province, Research on Multi-Sensor Fusion and Key Technologies for Intelligent Detection of Urban Underground Gas Pipeline Leaks (No.: 252102320049).

References

[1] Gangadhar Bandewad, Kunal P. Datta, Bharti W. Gawali, and Sunil N. Pawar. Review on discrimination of hazardous gases by smart sensing technology. Artificial intelligence and applications, 1(2):86-97, 2023. https://doi.org/10.47852/bonviewAIA3202434

[2] Qianqian Wang, Aimin An, Minan Tang, and Jiawei Lu. Distributed nonlinear model predictive control for cobalt removal process in zinc hydrometallurgy considering error compensation modelling. The Canadian Journal of Chemical Engineering, 102(1):307-323, 2024. https://doi.org/10.1002/cjce.25036

[3] Long Bai, Xin Cheng, Qizhong Yang, and Jianfeng Xu. Predictive model of surface roughness in milling of 7075Al based on chatter stability analysis and back propagation neural network. The International Journal of Advanced Manufacturing Technology, 126(3):1347-1361, 2023. https://doi.org/10.1007/s00170-023-11133-6

[4] Fazhan Zeng, Ren Wan, Yongjun Xiao, Fan Song, Changjun Peng, and Honglai Liu. Predicting the self-diffusion coefficient of liquids based on backpropagation artificial neural network: a quantitative structure-property relationship study. Industrial & Engineering Chemistry Research, 61(48):17697-17706, 2022. https://doi.org/10.1021/acs.iecr.2c03342

[5] Jiang L, Duan J J, Zheng R P, Shen H N, Li H, and Xu J. Optimization and simulation of garment production line balance based on improved GA. International Journal of Simulation Modelling, 22(2):303-314, 2023. https://doi.org/10.2507/ijsimm22-2-co6

[6] Xinzhe Yin, Jinghua Li, and Shoujun Huang. The improved genetic and BP hybrid algorithm and neural network economic early warning system. Neural computing & applications, 34(5):3365-3374, 2022. https://doi.org/10.1007/s00521-021-05712-5

[7] E. Gangadevi, R. Shoba Rani, Rajesh Kumar Dhanaraj, and Anand Nayyar. Spot-out fruit fly algorithm with simulated annealing optimized SVM for detecting tomato plant diseases. Neural computing & applications, 36(8):4349-4375, 2024. https://doi.org/10.1007/s00521-023-09295-1

[8] Dongmei Xu, Xiangqi Wang, Wen Wang, K. Chau, and Hongfei Zang. Improved monthly runoff time series prediction using the SOA-SVM model based on ICEEMDAN-WD decomposition. Journal of hydroinformatics, 25(3/4):943-970, 2023. https://doi.org/10.2166/hydro.2023.172

[9] Hongchang Zhang, Yang Zou, Enrique del Rey Castillo, and Xiaofei Yang. Detection of RC spalling damage and quantification of its key properties from 3D point cloud. KSCE journal of civil engineering, 26(5):2023-2035, 2022. https://doi.org/10.1007/s12205-022-0890-y

[10] Xing Fan, Mengdi Sun, Zhong Li, Zhenhua Chen, Xiaoyong Zhou, Quanxuan Lu, and Zhicheng Zhang. Tuning piezoelectric properties of P(VDF-TFE) films for sensor application. Reactive and functional polymers, 180(1):50-57, 2022. https://doi.org/10.1016/j.reactfunctpolym.2022.105391

[11] Nasim Kamely. Interaction of light with different electroactive materials: a review. Journal of electronic materials, 51(3):953-965, 2022. https://doi.org/10.1007/s11664-021-09362-0

[12] He J, and Yang J. Network security situational level prediction based on a double-feedback elman model. Informatica: An International Journal of

Computing and Informatics, 46(1):87-93, 2022. https://doi.org/10.31449/inf.v46i1.3775

[13] Kumar Y, Dahiya N, Malik S, and Savita K. A new variant of teaching learning-based optimization algorithm for global optimization problems. Informatica, 43(1):65-75, 2019. https://doi.org/10.31449/inf.v43i1.1636

[14] Panda D, Panda D, Dash S R, and Shantipriya P. Extreme learning machines with feature selection using GA for effective prediction of fetal heart disease: a novel approach. Informatica: An International Journal of Computing and Informatics, 45(3):381-392, 2021. https://doi.org/10.31449/inf.v45i3.3223

[15] Rui Wang, Anxiang Song, Xiang Chen, Yuanchen Guo, Xue Wang, Yan Sun, and Miao Tian. Experimental study on the properties of phosphate-based materials for rapid repair of concrete cracks. KSCE Journal of Civil Engineering, 26(5):2342-2353, 2022. https://doi.org/10.1007/s12205-022-1192-0

[16] Kulkarni S A, Gurpur V, Koval K A. Impact of Gaussian Noise for Optimized Support Vector Machine Algorithm Applied to Medicare Payment on Raspberry Pi. Informatica: An International Journal of Computing and Informatics, 45(4):643-652, 2021. https://doi.org/10.31449/inf.v45i4.3747

[17] Tongtong Jin, Qiang Cheng, Hu Chen, Siyuan Wang, Jinyan Guo, and Chuanhai Chen. Fault diagnosis of rotating machines based on EEMD-MPE and GA-BP. The International Journal of Advanced Manufacturing Technology, 124(11):3911-3922, 2023. https://doi.org/10.1007/s00170-021-08159-z

[18] Chandra B. Khatri, Saurabh K. Yadav, Gananath D. Thakre, and Arvind K. Rajput. Design optimization of vein-bionic textured hydrodynamic journal bearing using genetic algorithm. Acta mechanica, 235(1):167-190, 2024. https://doi.org/10.1007/s00707-023-03734-9

[19] Junyu Chen, Yun Feng, D. Teng, and Cheng Lu. Support vector machines-based pre-calculation error for structural reliability analysis. Engineering with computers, 40(1):477-491, 2024. https://doi.org/10.1007/s00366-023-01803-0

[20] Liang Zhao, Shenglun Gao, Junying Chen, and Jiajia Li. Feature detection of concrete mesoscopic damage based on feature sharing double-head Cascade R-CNN. Control and Decision, 37(7):1-7, 2022. https://doi.org/10.13195/j.kzyjc.2021.0124

Self-Learning Model for Pattern Recognition in Vision System Based on Adaptive Kernel

Aradea*, Rianto, Nina Herlina, Irani Hoeronis
Department of Informatics, Faculty of Engineering, Siliwangi University, Kode Pos 46115, Tasikmalaya, Indonesia
E-mail: aradea@unsil.ac.id, rianto@unsil.ac.id, ninaherlina@unsil.ac.id, iranihoeronis@unsil.ac.id
*Coresponding Author

Keywords: neural network, pattern recognition, self-adaptation, self-learning, vision system

Received: October 4, 2024

Recently, the solution for recognizing and understanding an object based on visuals is to integrate the adaptation function (continuous machine-driven process) into the system update function involving humans (continuous human-driven process). However, this has created a gap between the adaptation function and the system. This situation requires understanding the system viewed as a dynamic composition of the learning process. This research introduced a self-learning model in the form of an adaptive kernel equipped with the SpinalNet architecture, and the goal of this study is to increase the Convolutional Neural Network (CNN) accuracy. The model consisted of a domain model, contextual knowledge, and adaptive learner developed based on the CNN with SpinalNet. The combination of Adaptive Kernel and SpiralNet in this CNN has a significant impact, allowing the model to adjust the selection of subsequent kernels based on the optimal input from the previous kernel. Moreover, this combination results in lower memory usage during training. The evaluation results show that our proposed model provides better classification accuracy than the SpiralNet model without the Adaptive Kernel. Furthermore, in terms of inference speed, our model outperforms SpiralNet, as evidenced by the use of fewer parameters.

Povzetek: Prilagodljiv model samoučenja, ki temelji na jedru, izboljša prepoznavanje vzorcev v sistemih za vid, integracijo CNN s SpinalNet za izboljšanje natančnosti klasifikacije, optimizacijo izbire jedra in zmanjšanje uporabe pomnilnika med usposabljanjem.

1 Introduction

Computer (system) vision is a field of artificial intelligence that trains computer machines to interpret and understand (recognize) the visual world through deep learning models. The goal is that the machine can accurately identify and classify real-world objects and then react to what it sees. At present, the recognition and reaction capabilities of a system will be associated with the complexity of a highly dynamic, unpredictable, and uncertain environment [1]. In addition, the involvement of various elements of the real world interacting with the system will require the adaptability of the system. This ability will determine the success or failure of a system in recognizing and acting on what is occurring in its environmental context [2]. In fact, [3] states that the need to develop a system has entered the wave of learning from experience, namely the deployment of machine learning techniques. It functions to support various system functions to create an adaptive system, including a system capable of operating under conditions of uncertainty. Also, it can guarantee that its main property will function optimally. Therefore, a vision system for the current world requires a pattern recognition model possessing adaptability and a reliable optimization level.

Adaptability in a system aims at realizing the behavior of adapting a system built based on special requirements [4]. This situation, among others, requires a system to recognize changes in its application domain. Additionally, it can change itself to produce alternative behaviors [5]. Further, [3] in his latest review of long-term challenges that could trigger a new wave of scrutiny in the field of self-adaptation, raises an interesting question, namely the extent to which to develop systems to handle conditions that were not (fully) anticipated at the time the system was cultivated. Researchers have proposed various approaches to fostering adaptability in a system based on their respective problem domains. As a result, currently, neither a definition nor a specification for a system's adaptability has been widely agreed upon [6]. Besides, this applies to the specification of adaptability in vision systems. As an example, there is a need for deep meta-learning applied to image recognition problems [7]. This problem can be resolved by understanding the system viewed as a dynamic composition of the learning process, namely how to enhance the system with self-learning abilities [3].

The perspective of growing adaptability is grounded in the self-learning model. The idea is to overcome the gap existing in the traditional perspective. In particular, there is a need to integrate the adaptation function (continuous machine-driven process) into the system update function involving humans (continuous human-driven process). Consequently, the system can only run for a short cycle since it has to wait for updates to deploy. Researchers generally develop adaptability for pattern recognition in vision systems by expanding various features to complement machine learning's ability to recognize visual cues. Some approaches or techniques can be used. Generally, they can be categorized into three categories, namely feature-based, template-matching, and image-

based [8]. Image-based techniques are one of the concerns in this study because they can utilize all parts of an image. As a result, the detection process does not depend on the characteristics of an image or not focus on matching small parts of the image, becoming the model [9], [10]. It is expected that our research can be more flexible in developing a generic model to capture and recognize objects holistically with an optimal level of accuracy with self-learning capabilities.

The existing main problem related to the application of machine learning for vision systems is to determine the most optimal algorithm. In addition, the researchers have conducted miscellaneous empirical investigations on various existing algorithms. One of them is a neural network. Nowadays, neural networks have become a method in machine learning with great success, including in object detection research [11] which initially had difficulties in its development. With various extensions of existing neural network-based methods, the development process has become easier [12]. One of them is the utilization of a deep neural network. It is a neural network architecture delving image data. In the context of a vision system, object detection is performed by training a computer to interpret and understand the visual world through a deep learning model. Hence, the machine can accurately identify and classify objects and react to what it screens. Therefore, the vision system requires a pattern recognition model to reach a reliable level of optimization and adaptation.

There are myriad neural network algorithms. One of the developments (types) is the Convolutional Neural Network (hereafter, CNN) algorithm. CNN is a variation of Multilayer Perceptron (hereafter, MLP) designed to process two-dimensional data. On the one hand, MLP is not suitable for use in the case of image classification since it does not store special information from image data and considers each pixel as an independent feature, resulting in poor results [13]. On the other hand, CNN is also a type of deep neural network designed to process two-dimensional data with a high network depth and is widely applied to image data [14]. Based on research [15], CNN has shortcomings in terms of the old model training process. Therefore, there has been a plethora of studies developing the CNN algorithm to get results or performance, especially regarding the level of accuracy so that it gets better. One of the developments in the use of optimization algorithms. Several optimization algorithms are included in the minibatch-based adaptive algorithm or algorithms included in the gradient descent optimization algorithm.

These works allow us to extend the self-learning model based on neural network theory. A neural network, as a fundamental primitive, can provide flexibility in designing an architecture that focuses on adaptability. However, its impact on computational complexity should also be noted. Furthermore, various existing research results mainly accentuated the level of accuracy in the pattern recognition process. Only a tiny proportion pays attention to the adaptability of the learning process. One of the reasons for this is the lack of a good representation for meta-learning [7]. This study introduced a self-learning model for pattern recognition in the vision system by bringing up the adaptability function in the learning process. The model consisted of an optimized CNN algorithm employing an adaptive kernel. Thus, CNN can adapt to the model parameters in the learning process. The rest of this study consists of the second part discussing relevant studies, the third part describing the proposed model, and the fourth part eliciting the application of the model. In particular, it consists of experiments and a discussion of the evaluation results. Finally, the fifth part concludes all the work results and discusses future job opportunities. The rest of this study consists of the second part discussing relevant studies, the third part describing the proposed model and the fourth part eliciting the application of the model. In particular, it consists of experiments and a discussion of the evaluation results. Finally, the fifth part concludes all the work results and discusses future job opportunities.

2 Related work

There have been various empirical results relevant to machine learning for pattern recognition needs in vision systems. [16] compared the results of applying various optimization algorithms in deep learning, namely CNN, with three different CNN architectures. This study deployed two machine learning models, namely supervised and unsupervised learning. There were ten algorithms compared in this study, including the minibatch-based adaptive algorithm or algorithms included in the gradient descent optimization algorithm, namely the Stochastic Gradient Descent (SGD) algorithm, SGD-Momentum, SGD-Nesterov, AdaGrad, AdaDelta, RMSProp, Adaptive Momentum, AdaMax, Nadam, and AMSGrad. Four datasets were utilized: MNIST, CIFAR-10, LFW, and Kaggle Flowers. One of the results of this study was that the Adaptive Momentum optimization algorithm worked optimally. In other words, it reached the highest level of accuracy when applied to the first and third CNN architectures with the dataset applied as LFW. Besides, [17] also compared the performance of CNN. The results indicated that the Adaptive Momentum optimization algorithm had the highest level of accuracy. This study applied the Adaptive Momentum algorithm to three different CNN architectures, namely ShallowNet, LeNet, and AlexNet. The results reported that the best way to increase the accuracy of photosynthetic pigment prediction on plant digital images was to deploy the adaptive momentum algorithm combined with the LeNet architecture.

Currently, the use of CNN architecture has reached a higher level by adding an adaptive scheme to the training process. The research in [18] introduced an adaptive learning rate rule in CNN training by integrating the Egret Swarm Optimization Algorithm (ESOA) and quadratic interpolation (QIESOA) to improve prediction accuracy. Adapting the learning rate improved CNN's weaknesses in multi-domain image classification tasks, achieving the highest accuracy of 97.15% on the test dataset. Luo and Hu [19] developed Adaptive Attention ResNet (AA-ResNet), which addresses overfitting and training errors in CNNs with deeper networks. Feature extraction became a

primary focus of their research, using residual modules and adaptive attention to enhance feature representation. The developed model demonstrated high performance on the Cifar-10, Caltech-101, and Caltech-256 datasets. The research by Jiang et al. [20] discussed the role of activation functions in Convolutional Neural Networks (CNNs). It introduced the Adaptive Offset Activation Function (AOAF) as a solution to improve image classification accuracy. AOAF is a new parametric activation function that connects negative and positive values by adding an adaptive parameter (the average of the input feature tensor) [20]. The results showed that AOAF significantly improved accuracy, especially on datasets with high feature complexity. Wu and Pan [21] introduced an adaptive modular convolutional neural network (CNN) model design to improve efficiency and accuracy in image recognition tasks. Through a gate unit based on attention mechanisms, the model adaptively selects the optimal network structure based on learning. The results showed high accuracy on three Kaggle datasets (Cats-vs.-Dogs, 10-Monkey Species, Birds-400). The research by Guo et al. [22] focused on developing an Adaptive Pooling Network (APN) based on memristor arrays to improve the performance and resilience of CNNs in managing information loss during pooling. The results demonstrated that APN enhanced CNN performance in terms of both accuracy and robustness on the MNIST and CAPTCHA datasets. To clarify the research results and identify gaps in the state-of-the-art concerning adaptability in vision systems, especially CNNs, we have summarized the findings in a table, as shown in Table 1.

Table 1: State-of-the-art

Research	Proposed Method	Problem	Contribution	Result	Weakness
Wei dkk. [18]	CNN + QIESOA	Slow convergence of traditional CNNs	Adaptive learning rate update with ESOA and Quadratic Interpolation.	91.25% (Cifar-10), 88.66% (EMNIST), 95.87% (EuroSAT), 88.66% (Fashion-MNIST), 97.15% (RiceImage).	Adaptation to datasets with high dynamics or specialized domains has not been discussed.
Luo dan Hu [19]	AA-ResNet	Overfitting due to network depth.	Adaptive attention, multitask loss function.	92.43% (Cifar-10), 69.61% (Caltech-101), 52.29% (Caltech-256).	Adaptation to large-scale datasets or new domains has not been tested.
Jiang dkk. [20]	AOAF	Low performance of the ReLU function.	Using negative values in feature extraction.	Accuracy increased by 4.8% compared to ReLU	Not tested on datasets with high noise or different distributions.
Wu dan Pan [21]	Adaptive Modular CNN Model	Overfitting, large parameters.	Parallel modules and submodules, adaptive reduction of FLOPs.	99.3% (Cats-vs-Dogs), 99.26% (10-Monkey Species), 99.13% (Birds-400)	Not evaluated on datasets with noise or extreme variations.
Guo dkk. [22]	APN (Memristor-based)	Information loss in CNN pooling.	Adaptive pooling without backpropagation.	99.3% (MNIST), 92.6% (CAPTCHA).	Difficult to adapt to systems without memristors and large datasets.

Self-learning capabilities for vision systems have also been developed [7] by proposing a framework consisting of three main modules: the concept generator, meta-learners, and concept discriminators. This framework integrated the representational power of deep learning into meta-learning. The results substantially improved vanilla meta-learning, demonstrated in various few-shot image recognition problems. Other researchers, including [23], employed a new structure and concept called SpinalNet. SpinalNet is an amalgamation of DNN and Gradual Input implementations. This study highlighted the shortcomings of DNNs related to computational intensity due to the size of the input network. Therefore, this study applied gradual input, which was the concept of input gradually, to reduce the burden of the calculation process. The results of this

study indicated that SpinalNet was able to increase the accuracy of the usual DNN.

We had studied the model's adaptability before, starting with integrating the self-adaptation approach into requirements modeling [24]. As an illustration, we introduced a self-adaptation approach embedded into the primitive system requirements specification. Furthermore, in the study [25], we added a contextual-requirements approach to the adaptation pattern of the primitive system requirements. The goal was to capture the relevant context attributes so that the adaptive behavior of the system would match the prevailing context. In another study [1], we developed a pattern of adaptation to deal with the variability of system services. In this case, our primitive system requirements map to the various service levels of the system. In this study [2], we introduced the adaptation requirements for the adaptive systems (ARAS) framework, extending the system modeling language with control loop patterns and the context inheritance hierarchies. Technically, both were mapped into a graph network (Bayesian Network). We have defined several formalizations for adaptability in a graph. However, the results specific to the requirements of the vision system have not been attained. More recently, in the paper [26], [27], we merely attempted to apply the adaptability of this graph network to the needs of the Internet of Things (IoT) network system.

We captured research opportunities Based on the related job descriptions and studies conducted previously. In this case, the study can be performed to improve (to enhance) the adaptability of the learning process in pattern recognition for vision systems. One example is the expansion of the CNN model development. More technically, the addition of the SpinalNet architecture to the CNN model that we have developed can have the opportunity to increase the adaptability and optimization of the learning process. Meanwhile, based on studies in related studies, it was explained that CNN fit image data. It has even been widely applied to image data [8]. Consequently, image-based techniques were also our concern when formulating the needs of this research. Additionally, [7], contended that there is a lack of a good representation for meta-learning, where this meta-learning will learn the learning algorithm (meta-learner) of many related tasks. These statements and facts have motivated us to develop a new model with self-learning capabilities for pattern recognition in vision systems.

3 Proposed method

The perspective used in developing this proposed model was inspired by [3]. In this sense, [3] notes that the challenge in the long-term triggering a new wave of research in the field of self-adaptation is to understand the system as a dynamic composition of the learning process. The idea is to enhance a system with self-learning capabilities. To illustrate, a system allows it to learn from the variety of data it collects and autonomously develops its learning process under changing and unpredictable conditions. In the context of the vision system, the work of [7] applied this perspective by proposing deep meta-learning. Further, they also demonstrated its usefulness in image recognition problems. This work was extremely inspiring for us to propose a new model of self-learning capability for vision systems. Our model consists of three main components, namely the domain model, contextual knowledge, and adaptive learner as presented in Figure 1:

a. Domain model is a domain modeling in the form of a graph network structure to capture high-level visual signal representations.
b. Contextual knowledge represents the relevant context attributes in the model domain according to the current dynamic visual cue context.
c. Adaptive learner consists of utility (utility function) and learner (learner function) functions that carry out learning and recognize visual cues representations based on the prevailing context.

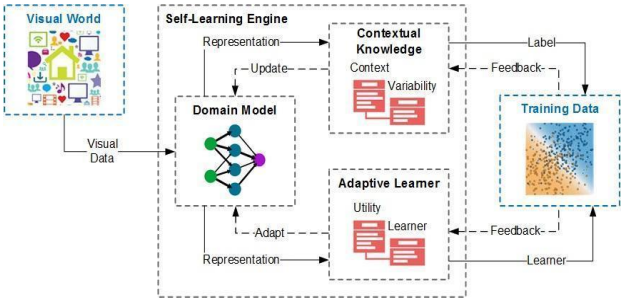


Figure 1: Self-learning model for pattern recognition in the vision system

Domain model

In a previous study [1], [2], we defined every element in the model domain indicating a dependency relationship. Furthermore, the model was regarded as a dynamic property in nature to be monitored based on certain parameter values. In this study, we developed it to specific representations for monitoring and capturing high-level visual cues. More specifically, the model deployed the SpinalNet structure developed by [23] taking inspiration from the human somatosensory system as presented in Figure 2. Following the way of how the human spinal network works, Spinal Net utilized gradual input (Gradual Input). All the layers contained in the model contributed to the main output of X in the same way that reflexes worked. Next, the modular input was sent to the main output of X. It was similar to how the brain works.

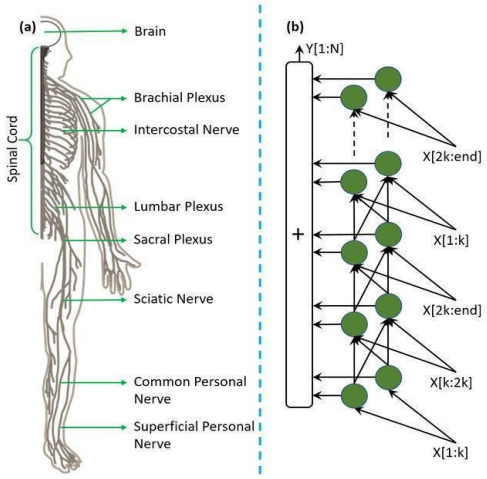


Figure 2: SpinalNet model from (Adapted from [25])

In a model like the one illustrated in figure 3, the first layer utilized a simple linear function and obtained only the sum of weight w from x1-x5. The second layer of the model gained the total weight w of x6-x10 as one input and the result of layer 1 as the other input. Briefly stated, the definition can be formulated as follows:

a. For each layer $x_i \in \{x_1, x_2, \ldots, x_n\}$ will contribute to the main output layer X.
b. For each input $N_i \in \{N_1, N_2, \ldots, N_n\}$ can be modularized into each of its x_i layers and become inputs for x_{i+1}. layers.

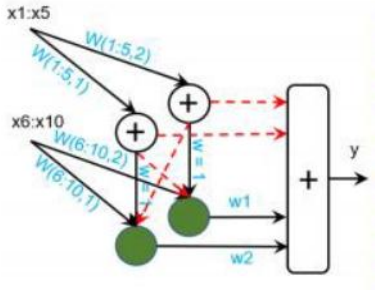


Figure 3: Simplified SpinalNet as a single hidden layer from (Adapted from [23])

Contextual Knowledge

Contextual knowledge is a representation of dynamic properties in the model domain [2]. It refers to the abstraction of domain properties relevant to the expected system behavior. Also, it covers the specific context in which this expected behavior applies [28], [29], [30]. In this investigation, contextual knowledge was specified for the needs of context attributes related to visual cues in the model domain. The attribute applied as contextual knowledge in this research was the kernel dimension. It was intended to determine the size of the matrix to perform convolution and input shift. The kernel on convolution is formulated as follows:

a. $F(x) * F(y)$ is the dimension of the kernel matrix.
b. $N(x) * N(y)$ is the dimension of the input matrix.
c. The output dimension of the convolution is $N(x) - F(x) + 1 * N(y) - F(y) + 1$.
d. Convoluting the kernel $Q_{u,v}$ with the activation function tanh will result in weight $K_{u,v}$.

Adaptive learner

The adaptive learner is a module that can automatically serve adjustments due to changing and growing needs. The main purpose of this module is to model the system dynamically. In particular, the module learned to recognize every need existing in the model domain and contextual knowledge on a run-time basis. The main problem to be handled was related to variables with varying, different, and flexible properties. This module indicated two functions, namely the utility function in the form of a function to sort or define alternative varieties according to their use for individual visual cues, and the learner function to carry out learning and introduction to obtain the most optimal results. The new kernel function was obtained through the result of the convolution of each input convolution $Q_{(u,v)}$ as in the following equation:

$$\sigma_{u,v} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} Q_{(u,v)_{i,j}} x_{i,j} \quad ..(1)$$

The new kernel $K_{(u,v)}$ can then be deployed to perform convolution on the input image to produce S. Subsequently, it was applied as the output kernel as in the following calculation:

$$S = \sum_{u,v} x_{u,v} K \left(\sum_{i,j} Q_{u,v_{i,j}} x_{i,j} \right) ..(2)$$

$$f = tanh(S) ..(3)$$

4 Experiment

This section describes the evaluation of our proposed model for recognizing visual cue patterns, particularly handwriting patterns. In this experiment, we deployed MNIST datasets sourced from the research of LeCun, et. al. [31]. These datasets refer to a collection of handwritten images of numbers 0-9 consisting of 60,000 training data and 10,000 test data. The images were black and white. Each image was 28x28 pixels. The use of the MNIST dataset on the CNN method was performed by Saqib, et. al. [32]. The study succeeded in building a model recognizing and classifying handwritten figure images. The experimental results showed that the CNN model attained the highest classification of accuracy for a certain number of hidden layer neurons. Another scrutiny was conducted by Anwar, et. al. [33] Involving the MNIST dataset as the classification object of CNN. In addition to using MNIST, we also applied other datasets such as KMNIST, QMNIST, Fashion-MNIST, and EMNIST to strengthen the validation of the model we have developed.

Preparation of model application

The network architecture structure developed in this experiment was inspired by the SpinalNet architectural model by carrying out several expansions, namely combining it with the Convolutional Neural Network architecture through an adaptive kernel on the convolution layer. The experimental mechanism was applied to the MNIST dataset with several models. As an example, the conventional CNN model commonly used covers the CNN model combined with the Adaptive Kernel, the SpinalNet model, and the model developed by the authors. Contextual knowledge elicitation was conducted to identify the relevant context attributes in the model domain related to the dynamic context of visual cues. This provides dynamic parameter updates during training on the adaptive kernel. The adaptive kernel parameters are iteratively updated during the backpropagation process. The kernel adapts by minimizing the cross-entropy loss through the gradient descent algorithm. The utility function calculates the optimal kernel value based on the contextual knowledge that has been learned. This mechanism allows the kernel to dynamically shift its focus and optimize the most relevant features for visual signals. Unlike non-adaptive kernels, which rely on static parameters, adaptive kernels dynamically adjust their parameters during training. For instance, after each convolution operation, the kernel dimensions are updated to optimize weight alignment in subsequent layers. This flexibility results in higher accuracy and efficiency, as demonstrated in our model. Figure 4 shows the distinction between the adaptive and non-adaptive kernel processes to clarify the differences.

In this experiment, we identified the data collected from the results of pre-processing and preparation for the application of the model as contextual knowledge, namely the kernel that can change according to the determined input.

Pattern recognition implementation and operation

Our proposed model applied three main parts, namely the Adaptive Kernel, Convolutional Layer, and Full Connected Layer as shown in figure 5. First, the Adaptive Kernel was a Convolutional Layer involving an adaptive system in its kernel parameters. The determination of the kernel was based on the optimal input of the previously applied convolution. Second, the Convolutional Layer, both Adaptive Kernel and Convolutional Layer applied Maxpooling and Relu as activating functions. By applying the Spinal Layer to the Full Connected Layer section, the input parameters were smaller. As a result, memory usage can be kept to a minimum in learning the model. Third, Spinal Layer divided the input into several parts and then processed it with a linear function. In our model, the input was divided into two equal sizes and was processed linearly in six layers. In the final stage of the full connected layer, a linear function was utilized to combine the applied Spinal Layers. The utilized Spinal Net structure is shown in Figure 6.

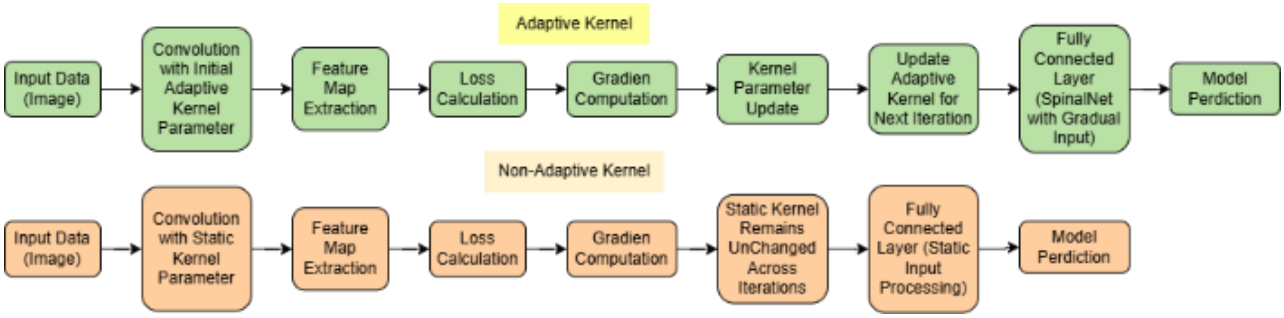


Figure 4: Differences between adaptive kernel approaches compared to non-adaptive methods.

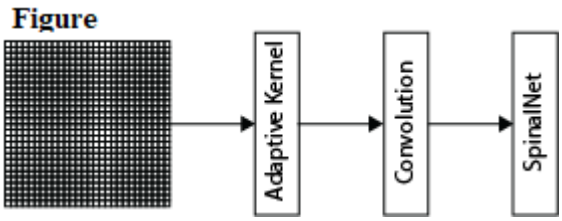


Figure 5. Self-learning model architecture

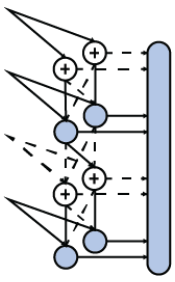


Figure 6: SpinalNet architecture in full-connected layer

More specifically, the implementation of the integration between the adaptive kernel and SpinalNet is shown in Figure 7, which illustrates the workflow of the proposed model.

The model designed in Figure 7 processes a 28x28 grayscale input image through a series of steps, starting from the dynamic kernel to the fully connected layer. In

the dynamic kernel, the kernel weights are adaptively adjusted during training, resulting in 25 feature maps (28x28x25). This output is then passed to the dynamic layer, where the results from multiple kernels are combined, and the channels are reduced to 6 (28x28x6). Next, the Conv2D Layer extracts deeper features, followed by the MaxPooling Layer for downsampling, producing an output of 12x12x20. A Dropout Layer is applied to prevent overfitting without altering the data dimensions. The data is then flattened through the Flatten Layer into a 1D vector (500 elements), which is processed progressively by the SpinalNet Layers by splitting the vector into six segments and generating a combined representation with a total of 1500 elements. Finally, the Fully Connected Layer processes this representation into logits for 10 classes to generate probabilities, determining the final class prediction. Combining the Adaptive Kernel, Convolutional Layer, and SpinalNet ensures computational efficiency and model adaptability in handling visual data.

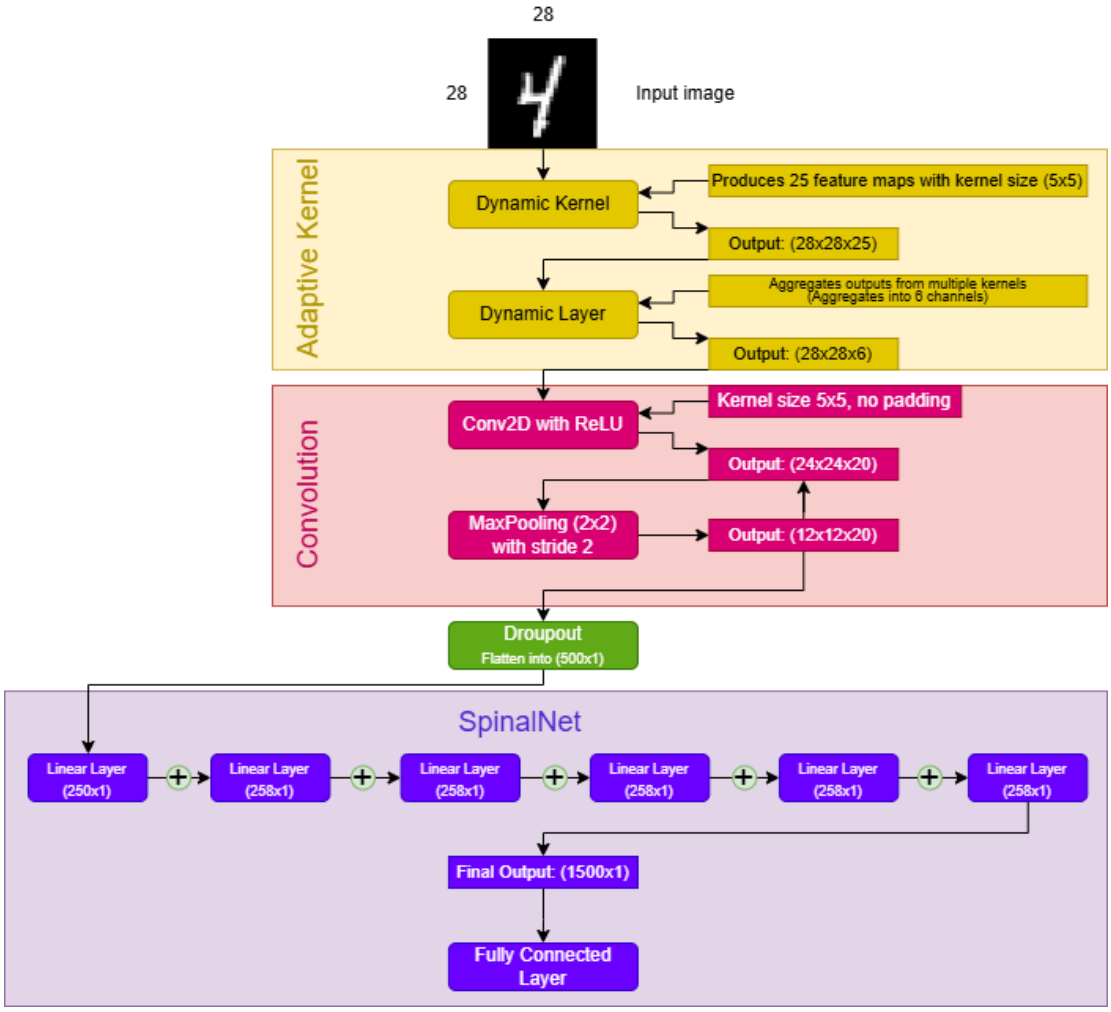


Figure 7: Purposed method framework

The integration results between SpinalNet and the Adaptive Kernel were trained using various datasets for classification tasks. This experiment has two training scenarios: one where the model is trained with the additional VGG-5 network [34] and another where the model is trained without that additional network. The model with the added VGG-5 was trained for 100 epochs, using a batch size of 128 and a learning rate 5×10^{-3}. In contrast to the hyperparameters used in the first scenario, the model without the VGG-5 addition was trained for eight epochs, using a batch size of 128 and a learning rate of 1×10^{-2}. The difference in hyperparameter usage was made to adjust to the needs of each model being trained to maximize the potential of the training results. Additionally, both training scenarios were optimized using Stochastic Gradient Descent (SGD) with the same momentum value 0.9.

These hyperparameters were determined based on the results of a systematic evaluation of several hyperparameter choices using a grid search approach. The evaluation was based on validation accuracy across various configurations while also monitoring the stability of the loss function and the efficiency of the number of parameters in the model. The evaluation results for each hyperparameter choice are shown in Tables 1 and 2.

Table 1: Hyperparameter testing for the proposed model with the added VGG-5

Hyperparameter	Range	Optimal Value
Learning Rate	[0.001, 0.005, 0.01]	0.005
Batch Size	[32, 64, 128]	128
Hidden Layer in SpinalNet	[64, 128, 256]	128
Neuron per Layer in SpinalNet	[64, 128, 256]	128
Momentum	[0.5, 0.7, 0.9]	0.9

Table 2: Hyperparameter testing for the proposed model without VGG-5

Hyperparameter	Range	Optimal Value
Learning Rate	[0.001, 0.005, 0.01]	0.01
Batch Size	[32, 64, 128]	128
Hidden Layer in SpinalNet	[4, 6, 8]	8
Neuron per Layer in SpinalNet	[125. 250, 500]	250
Momentum	[0.5, 0.7, 0.9]	0.9

From the tests in Table 1, the optimal configuration for the proposed model with the added VGG-5 was obtained, which included a learning rate of 0.005, a batch size of 128, 128 hidden layers in SpinalNet, 128 neurons per layer, and a momentum value of 0.9 for SGD. Meanwhile, the optimal performance for the proposed model without the VGG-5 addition was achieved with a learning rate of 0.01, a batch size of 128, 8 hidden layers in SpinalNet, 250 neurons per layer, and a momentum value of 0.9. This configuration provided the highest validation accuracy, maintained a stable loss curve throughout training, and showed a balance between performance and computational efficiency.

Before the training process, we performed data preprocessing on all datasets used. In this process, we applied the same steps to all datasets, which included converting the images to tensors and normalizing the values. In the tensor conversion process, the pixel values of the images were changed from the original range (0 to 255) to the range [0.0, 1.0] by dividing each pixel value by 255. Afterward, the converted pixel values underwent normalization using the Z-Score normalization method. After passing through this data preprocessing stage, the model training process is expected to be faster and more stable, accelerating convergence and reducing imbalance.

Model evaluation and comparison

To validate the proposed model, we compared this research model with the original SpinalNet model. The comparison included accuracy, the number of parameters used, and the inference speed of the model on each test dataset used. Tables 3 and 4 compare the evaluation results between our model and SpinalNet.

Table 3: Comparison of Adaptive-SpinalNet and SpinalNet with the added VGG-5.

Dataset	Adaptive-SpinalNet		SpinalNet [23]	
	Accuracy	Inference Time	Accuracy	Inference Time
MNIST	99.78%	5.21s	99.72%	5.33s
KMNIST	99.24%	5.25s	99.15%	6.12s
QMNIST	99.54%	16.77s	99.68%	16.92s
Fashion-MNIS	95.21%	5.70s	94.68%	6.43s
EMNIST (Digits)	99.74%	12.57s	99.82%	`13.03s
EMNIST (Letters)	94.69%	8.68s	95.88%	9.17s

Table 3 highlights the comparison between VGG-5 + Adaptive-SpinalNet and VGG-5 + SpinalNet regarding accuracy and inference time. It is evident that the inference speed of our model consistently outperforms across all datasets. Similarly, the Adaptive-SpinalNet model demonstrates a speed advantage compared to the original SpinalNet model. The adaptive kernel dynamically adjusts weights based on the input it receives, enabling a focus on the most relevant features for the classification task and thereby reducing processing time for less significant information. Additionally, parameter efficiency is achieved by minimizing redundancy in kernel weights. This results in optimal representation without excess parameters that could slow the inference process. The comparison of parameter reduction is illustrated in Figure 8.

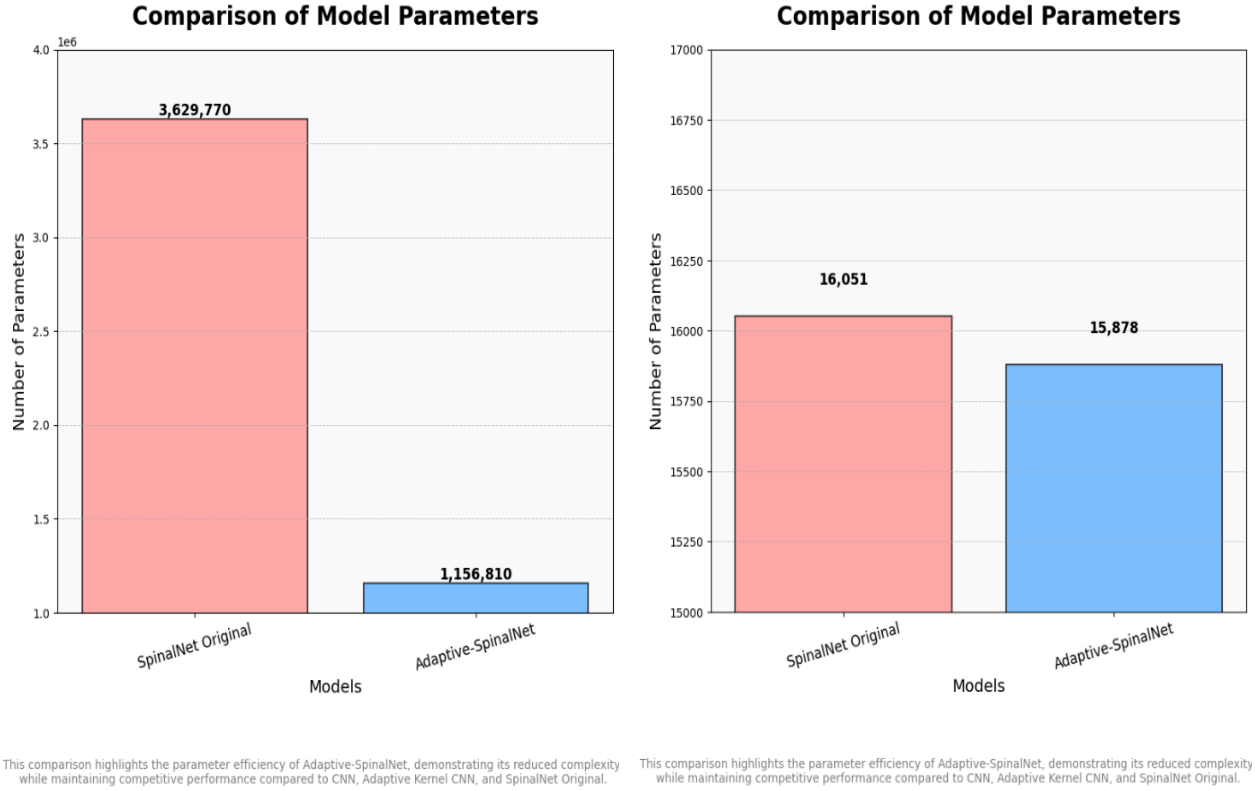
Table 4: Comparison Between Adaptive-SpinalNet and SpinalNet

Dataset	Adaptive-SpinalNet		SpinalNet [23]	
	Accuracy	Inference Time	Accuracy	Inference Time
MNIST	98.93%	3.42s	98.48%	3.61s
KMNIST	92.52%	3.81s	88.25%	4.08s
QMNIST	98.47%	12.85s	98.07%	13.03s
Fashion-MNIS	87.92%	3.54s	86.61%	3.90s
EMNIST (Digits)	99.35%	9.09s	99.16%	9.29s
EMNIST (Letters)	91.43%	5.24	90.23%	5.97s

In addition to its positive impact on parameter reduction, the SpinalNet architecture combined with Adaptive Kernel generally enhances accuracy across all datasets. This is particularly evident in Table 4, demonstrating that directly applying Adaptive Kernel to SpinalNet improves the model's accuracy on all test datasets. This indicates that our model, tested on various datasets (including MNIST, Fashion MNIST, KMNIST, and EMNIST), can generalize across different data distributions. Experimental results reveal that the Adaptive-SpinalNet

model consistently achieves competitive performance, even on datasets with significantly different visual patterns from MNIST. This highlights the model's ability to adapt to diverse data distributions. This adaptability is further reinforced by the Dynamic Kernel mechanism, which dynamically adjusts kernel weights based on input patterns during inference. This allows the model to capture relevant features under varying data conditions. Furthermore, the SpinalNet architecture processes feature independent segments, offering additional flexibility in handling shifts in data distribution.

Furthermore, to provide stronger validation, we compared the model's performance with related studies using the same dataset benchmarks. This comparison is presented in Table 5.



(a) (b)

Figure 8: Comparison of the number of parameters between Adaptive-SpinalNet and SpinalNet: (a) with VGG-5, (b) without VGG-5

Table 5: Comparison of Adaptive-SpinalNet with related studies

Model	Accuracy				Number of Parameters
	MNIST	KMNIST	QMNIST	Fashion MNIST	
SpinalNet [23]	98.48%	88.25%	98.07%	86.61%	16K
VGG-5 + SpinalNet [23]	99.72%	99.15%	99.68%	94.68%	3.6M
CNN + QIESOA [18]	-	-	-	97.15%	Not Mentioned
APN (Memristor-based) [22]	99.3%	-	-	-	Not Mentioned
R-ExplaiNet26-64 [35]	99.70%	98.66%	-	93.03%	0.89M
Improved Efficient Capsnet [36]	-	98.43%	-	-	0.58M

PMM [37]	97.38%	-	-	88.58%	4.9K (MNIST), 16.7K (Fashion MNIST)
ConvPMM [37]	99.10%	-	-	90.94%	0.13M (MNIST), 0.28M (Fashion MNIST)
Adaptive-SpinalNet	98.93%	92.52%	98.47%	87.92%	15.9K
VGG-5 + Adaptive-SpinalNet	99.78%	99.24%	99.54%	95.21%	1.1M

Table 5 demonstrates that the VGG-5 + Adaptive-SpinalNet model outperforms all other models in terms of accuracy on the MNIST and KMNIST datasets. Although its accuracy on the QMNIST and Fashion MNIST datasets remains slightly below the VGG-5 SpinalNet and CNN + QIESOA models, the differences are insignificant, indicating that our model performs well in handling data variability. The Adaptive-SpinalNet model has the fewest parameters compared to other models, except for the PMM model on the MNIST dataset. This proves the effectiveness of the Adaptive Kernel in reducing computational complexity in the SpinalNet model with minimal accuracy trade-offs. This performance is achieved through the Dynamic Kernel, which dynamically adjusts weights to extract relevant features, while SpinalNet processes features in independent segments to enhance flexibility and computational efficiency. With a low parameter count, Adaptive-SpinalNet demonstrates strong generalization across various datasets, making it suitable for real-world applications involving diverse data. In addition to the appropriate selection of hyperparameters, the performance achieved by Adaptive-SpinalNet is also attributed to the optimal sizing of the Dynamic Kernel. The kernel size significantly affects the model's adaptability. To clarify this, Table 6 presents the model's performance trained on the MNIST dataset using different Dynamic Kernel sizes.

Table 6: Comparison of adaptive-spinalnet model performance on the MNIST dataset based on kernel size

Ukuran Kernel	Recall	Precision	F1-Score	Accuracy
(3x3)	98.91%	98.91%	98.91%	98.91%
(5x5)	98.93%	98.93%	98.93%	98.93%
(7x7)	98.80%	98.80%	98.80%	98.80%
(9x9)	98.38%	98.38%	98.38%	98.38%

The results in Table 6 show a performance improvement when the kernel size is increased from (3x3) to (5x5). This suggests that enlarging the kernel size in the Dynamic Kernel can enhance performance. However, when the kernel size is further increased to (9x9), performance decreases. A larger Dynamic Kernel does not necessarily guarantee an improvement in model performance, as a very large kernel tends to aggregate information over a larger area, potentially overlooking important small or local patterns. In addition to this finding, another interesting observation from the comparison in Table 6 is the consistency between precision, recall, F1-Score, and accuracy. Identical values for precision, recall, and F1-score indicate that our model works effectively, achieves an optimal balance, and handles class distribution well. This demonstrates that our model performs well on the MNIST dataset.

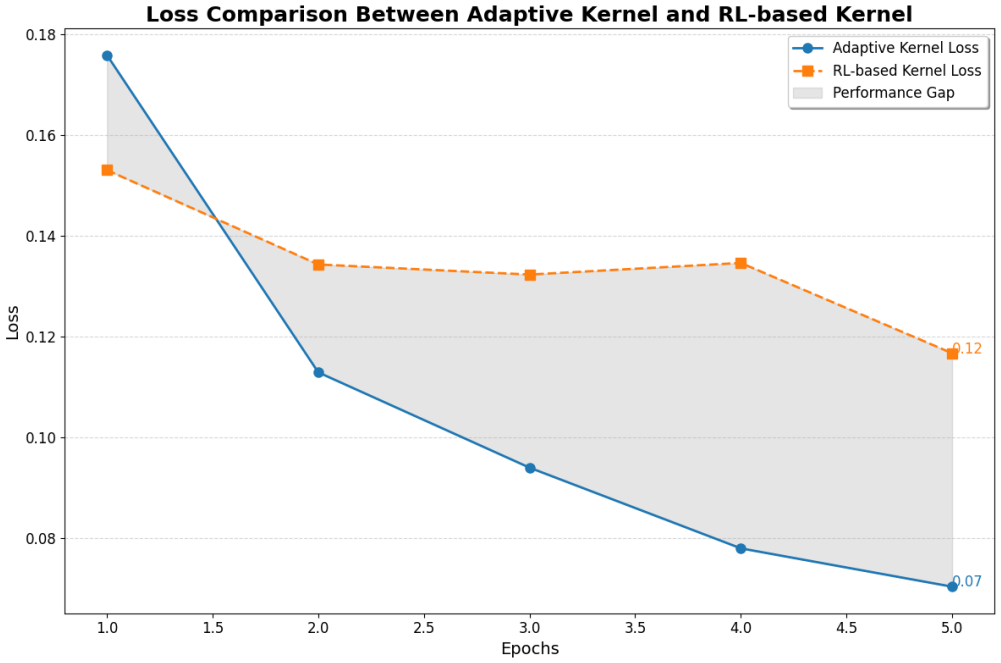
Another option that can be used as an adaptation method for the SpinalNet model is Reinforcement Learning (RL)--based adaptivity, which can be used to select or adjust kernels based on feedback from the environment to optimize performance. While this method may have the potential to adjust kernels based on experience, weaknesses such as computational overhead, dependence on reward design, and stability issues make it less ideal for high-efficiency real-time applications. The performance comparison between the Adaptive Kernel and RL methods in Table 7 demonstrates this.

Table 7: Performance comparison of adaptive kernel and rl methods on the spinalnet model using the MNIST dataset

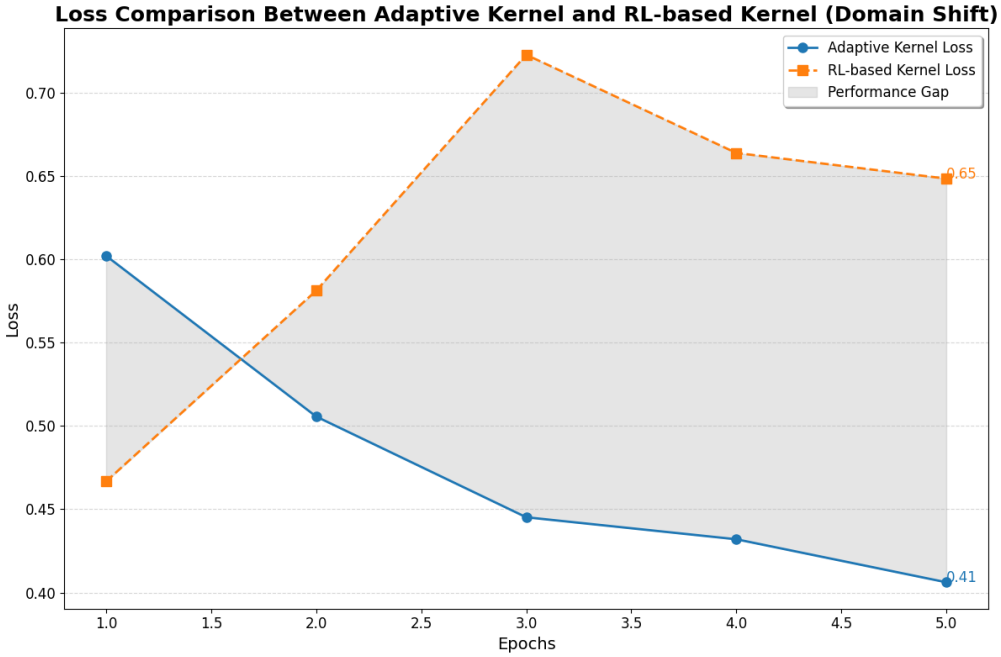
Method	Epoch	Acc (%)	Inference Time (s)	Domain Shift Acc (%)
Adaptive Kernel	5	97.85	3.42	88.97
Reinforcement Learning-Based Adaptivity	5	96.67	5.65	85.74

The comparison results in Table 7 show that the Adaptive Kernel method has a significant advantage over the Reinforcement Learning-based Adaptivity approach in terms of accuracy, inference time efficiency, and handling domain shift. Both methods were tested with five training epochs, with the Adaptive Kernel method achieving an accuracy of 97.85%, higher than the RL-based method, which only reached 96.67%. Furthermore, the inference time of the Adaptive Kernel is much faster, at 3.42 seconds, compared to 5.65 seconds for the RL method.

This indicates that the Adaptive Kernel method is more efficient for real-time applications than the RL-based adaptivity method. To further test the adaptability, we performed data augmentation for domain shift, which included random image rotation of up to 30°, brightness variation, and contrast changes. The evaluation results showed that the Adaptive Kernel's adaptation to domain shift was also superior, with an accuracy of 88.97% compared to 85.74% for the RL-based method. These results confirm that the direct adaptation mechanism of the Adaptive Kernel is more effective and efficient than the RL-based exploration, making it more suitable for adaptive vision systems that require high performance and resilience to data distribution changes. Another advantage is shown in the loss values generated at each epoch. Although the Adaptive Kernel method has a higher loss value than the RL-based adaptivity method in the first epoch, in subsequent epochs, the loss values for the proposed method consistently stay lower than those of the RL method. The comparison of loss values is shown in Figure 9.



(a)



(b)

Figure 9: Comparison of Loss Values Between Adaptive Kernel and RL-based Adaptivity. (a) Original MNIST Test Data, (b) Augmented MNIST Test Data

Threats to validity

a. Pre-Liminaries validity
   Validity in the preliminaries stage is measured by looking at the problem domain, which is understood as the clarity of data rows, datasets, and pre-processing. The transformation from non-linear to linearly separable transforms the data to a higher level by adding features using kernel functions. The raw data consists of various variants of the MNIST dataset, including MNIST itself, KMNIST, QMNIST, and Fashion-MNIST. Data identification is carried out to ensure that the pre-processing process in the model preparation stage is carried out correctly. This is a preparation stage for implementing the model as contextual knowledge. The data used is not too large. This was done to see how the model could be used with a limited amount of data but with high accuracy.

b. Fitting validity
   In the evaluation of our research, the process of determining the model is carried out using cross-validation and a confusion matrix. Validation is done by estimating the error and how our model can accommodate the unseen data. K-fold cross-validation is used to reduce parts that cause underfitting. By reducing training data, it is possible to lose trends in the data set, increasing the error caused by bias. The validation used is cross-validation, which generalizes the independent/unseen data set.At the validation stage, our learning self-learning model ensures that each process is carried out with attention to the evaluation of metrics, how to handle overfitting, and processes to reduce bias.

c. Bias validity

   Measurement of the accuracy of each model is optimized by optimization of Stochastic Gradient Descent (SGD) and Cross Entropy Loss. To eliminate the habit of estimating gradients, SGD is required to reduce the cost of each iteration. The computing cost of each iteration will run linearly from O(n) to O(1). In determining the SGD variable, the learning rate affects the resolution of the conflicting goal by reducing the learning rate dynamically as optimization progresses. Cross entropy is determined to define the loss function in optimization. This is done by minimizing the cross entropy. Defining cross entropy indirectly proves the equivalence of the relationship between objects. Done as long as the entropy data is constant..

5 Conclusion dan further studies

This study introduces a self-learning model for pattern recognition in vision systems. The model is developed through a self-adaptation approach where the system is regarded as a dynamic composition of the learning process. The goal is to enhance the system with self-learning capabilities, enable it to learn from collected visual data and develop its learning process autonomously. Our model encompasses three main components, namely (a) domain model to capture high-level representations of visual cues, (b) contextual knowledge representing context attributes relevant to the current dynamic context of visual cues, and (c) adaptive learner performing learning and recognizing visual cue representations based on the prevailing context. This model is prepared with a formulation combining the adaptive kernel method on the CNN architecture and the utilization of SpinalNet in the fully connected layer of the CNN.

The validity of the proposed model was evaluated using cross-validation with several testing schemes. In addition to the evaluation results compared with the original SpinalNet model, we also validated the model by comparing its performance through evaluations with methods used in related studies, varying kernel sizes, and comparisons with other adaptation methods. The evaluation results indicate that the proposed model performs very well regarding accuracy and computational complexity. The results of this work pave the way for future studies. In other words, future studies can include developing and expanding our proposed model for other domain needs (e.g., audio recognition, machine translation, and so on).

References

[1] A. Aradea, I. Supriana, and K. Surendro, “Self-adaptive model based on goal-oriented requirements engineering for handling service variability,” *Journal of Information and Communication Technology*, vol. 19, no. 2, pp. 225–250, 2020, doi: 10.32890/jict2020.19.2.4.

[2] Aradea, I. Supriana, and K. Surendro, “ARAS: adaptation requirements for adaptive systems,” *Automated Software Engineering*, vol. 30, no. 1, p. 2, 2022, doi: 10.1007/s10515-022-00369-3.

[3] D. Weyns, “Wave VII: Learning from Experience,” in *An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective*, 2020, pp. 201–226. doi: 10.1002/9781119574910.ch10.

[4] A. Mollajan, A. Shahdadi, A. Ashofteh, F. Hamedani-KarAzmoudehFar, and S. H. Iranmanesh, “System Adaptability Enhancement Based on Improving the System Reconfigurability by Modularization of the System Architecture,” 2023. doi: 10.2139/ssrn.4519777.

[5] M. Bhadra, D. S. Lopera, R. Kunzelmann, and W. Ecker, “ A Model-Driven Architecture Approach to Accelerate Software Code Generation ,” in *2024 7th International Conference on Software and System Engineering (ICoSSE)* , Los Alamitos, CA, USA: IEEE Computer Society, Apr. 2024, pp. 23–30. doi: 10.1109/ICoSSE62619.2024.00012.

[6] M. Huisman, J. N. van Rijn, and A. Plaat, “A survey of deep meta-learning,” *Artif Intell Rev*,

vol. 54, no. 6, pp. 4483–4541, Aug. 2021, doi: 10.1007/s10462-021-10004-4.

[7] T. Gong, X. Zheng, and X. Lu, "Meta Self-Supervised Learning for Distribution Shifted Few-Shot Scene Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2022.3174277.

[8] P. Terhörst, M. Huber, N. Damer, F. Kirchbuchner, K. Raja, and A. Kuijper, "Pixel-Level Face Image Quality Assessment for Explainable Face Recognition," *IEEE Trans Biom Behav Identity Sci*, vol. 5, no. 2, pp. 288–297, 2023, doi: 10.1109/TBIOM.2023.3263186.

[9] S. Malakar, W. Chiracharit, and K. Chamnongthai, "Masked Face Recognition With Generated Occluded Part Using Image Augmentation and CNN Maintaining Face Identity," *IEEE Access*, vol. 12, pp. 126356–126375, 2024, doi: 10.1109/ACCESS.2024.3446652.

[10] H.-I. Kim, K. Yun, and Y. M. Ro, "Face Shape-Guided Deep Feature Alignment for Face Recognition Robust to Face Misalignment," *IEEE Trans Biom Behav Identity Sci*, vol. 4, no. 4, pp. 556–569, 2022, doi: 10.1109/TBIOM.2022.3213845.

[11] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," May 2023, doi: https://doi.org/10.48550/arXiv.2305.09972.

[12] I. H. Sarker, "Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions," Nov. 01, 2021, *Springer*. doi: 10.1007/s42979-021-00815-1.

[13] C. Xu, "Applying MLP and CNN on Handwriting Images for Image Classification Task," in *2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE)*, 2022, pp. 830–835. doi: 10.1109/AEMCSE55572.2022.00167.

[14] L. Kurniasari and A. Setyanto, "Sentiment Analysis using Recurrent Neural Network," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2020. doi: 10.1088/1742-6596/1471/1/012018.

[15] G. Priyadharshini and D. R. Judie Dolly, "Comparative Investigations on Tomato Leaf Disease Detection and Classification Using CNN, R-CNN, Fast R-CNN and Faster R-CNN," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2023, pp. 1540–1545. doi: 10.1109/ICACCS57279.2023.10112860.

[16] D. Soydaner, "A Comparison of Optimization Algorithms for Deep Learning," *Intern J Pattern Recognit Artif Intell*, vol. 34, no. 13, p. 2052013, Dec. 2020, doi: 10.1142/S0218001420520138.

[17] G. L. Sree and R. Baskar, "Performance Analysis of CNN Algorithm in Comparison with LR algorithm for Face Recognition in Smart-Lock," in *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies*, 2024, pp. 1–5. doi: 10.1109/TQCEBT59414.2024.10545038.

[18] P. Wei, M. Shang, J. Zhou, and X. Shi, "Efficient adaptive learning rate for convolutional neural network based on quadratic interpolation egret swarm optimization algorithm," *Heliyon*, vol. 10, no. 18, Sep. 2024, doi: 10.1016/j.heliyon.2024.e37814.

[19] J. Luo and D. Hu, "An Image Classification Method Based on Adaptive Attention Mechanism and Feature Extraction Network," *Comput Intell Neurosci*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2023/4305594.

[20] Y. Jiang, J. Xie, and D. Zhang, "An Adaptive Offset Activation Function for CNN Image Classification Tasks," *Electronics (Switzerland)*, vol. 11, no. 22, Nov. 2022, doi: 10.3390/electronics11223799.

[21] W. Wu and Y. Pan, "Adaptive Modular Convolutional Neural Network for Image Recognition," *Sensors*, vol. 22, no. 15, Aug. 2022, doi: 10.3390/s22155488.

[22] W. Guo *et al.*, "A Memristor-Based Adaptive Pooling Network for Cnn Optimization," 2023. doi: 10.2139/ssrn.4648000.

[23] H. M. D. Kabir *et al.*, "SpinalNet: Deep Neural Network With Gradual Input," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 5, pp. 1165–1177, Oct. 2023, doi: 10.1109/TAI.2022.3185179.

[24] Aradea, I. Supriana, K. Surendro, and I. Darmawan, "Integration of Self-adaptation Approach on Requirements Modeling," in *Recent Advances on Soft Computing and Data Mining*, T. Herawan, R. Ghazali, N. M. Nawi, and M. M. Deris, Eds., Cham: Springer International Publishing, 2017, pp. 233–243, doi: 10.1007/978-3-319-51281-5_24.

[25] Aradea, I. Supriana, and K. Surendro, "Self-adaptive software modeling based on contextual requirements," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 16, no. 3, pp. 1276–1288, 2018, doi: 10.12928/TELKOMNIKA.v16i3.7032.

[26] A. Aradea, R. Rianto, and H. Mubarok, "Inference Model for Self-Adaptive IoT Service Systems," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 4, pp. 337–349, 2021, doi: 10.22266/ijies2021.0831.30.

[27] Aradea, Rianto, and H. Mubarok, "Cultivating Service Knowledge Models for IoT-Based Systems Adaptability," *Informatica (Slovenia)*, vol. 46, no. 5, pp. 115–122, 2022, doi: 10.31449/inf.v46i5.3874.

[28] M. Acheli, D. Grigori, and M. Weidlich, "Discovering and Analyzing Contextual Behavioral Patterns From Event Logs," *IEEE Trans Knowl Data Eng*, vol. 34, no. 12, pp. 5708–5721, 2022, doi: 10.1109/TKDE.2021.3077653.

[29] B. Yang, W. Wu, Y. Liu, and H. Liu, "A Novel Sleep Stage Contextual Refinement Algorithm Leveraging Conditional Random Fields," *IEEE Trans Instrum Meas*, vol. 71, pp. 1–13, 2022, doi: 10.1109/TIM.2022.3154838.

[30] W. Zhao, S. Peng, J. Chen, and R. Peng, "Contextual-Aware Land Cover Classification With U-Shaped Object Graph Neural Network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2022.3177778.

[31] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," *IEEE Signal Process Mag*, vol. 29, no. 6, pp. 141–142, 2012, doi: 10.1109/MSP.2012.2211477.

[32] N. Saqib, K. F. Haque, V. P. Yanambaka, and A. Abdelgawad, "Convolutional-Neural-Network-Based Handwritten Character Recognition: An Approach with Massive Multisource Data," *Algorithms*, vol. 15, no. 4, Apr. 2022, doi: 10.3390/a15040129.

[33] M. Anwar, H. M. Ali, M. A. Hossain, and A. Mohon, "Recognition of Handwritten Digit using Convolutional Neural Network (CNN)",Global Journal of Computer Science and Technology, 2019, doi: 10.34257/gjcstdvol19is2pg27.

[34] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego: Computational and Biological Learning Society, Sep. 2014, pp. 1–14. [Online]. Available: https://doi.org/10.48550/arXiv.1409.1556

[35] P. I. Kaplanoglou and K. Diamantaras, "Learning local discrete features in explainable-by-design convolutional neural networks," *arXiv preprint arXiv:2411.00139*, Oct. 2024, doi: https://doi.org/10.48550/arXiv.2411.00139

[36] M. Bukowski, I. Antoniuk, and J. Kurek, "Improved efficient capsule network for Kuzushiji-MNIST benchmark dataset classification," *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 71, no. 6, 2023, doi: 10.24425/bpasts.2023.147338.

[37] P. Cook, D. Jammooa, M. Hjorth-Jensen, D. D. Lee, and D. Lee, "Parametric Matrix Models," *arXiv preprint arXiv:2401.11694*, Jan. 2024, [Online]. Available: https://doi.org/10.48550/arXiv.2401.11694

A Learning-Based Ensemble Algorithm with Optimal Selection for Outlier Detection

Girish Reddy Ginni[1*], Srinivasa L . Chakravarthy[2]
[1]Department of Computer Science and Engineering, GITAM University, Gandhi Nagar, Rushikonda, Visakhapatnam, Andhra Pradesh, India
[2]Department of Computer Science and Engineering, GITAM University, Gandhi Nagar, Rushikonda, Visakhapatnam, Andhra Pradesh, India
girishloshankar@gmail.com, chakri.ls@gmail.com
[*]Corresponding author
Orcid: [1]https://orcid.org/0009-0005-5242-8839, [2]https://orcid.org/0000-0001-9141-4863

Keywords: ensemble outlier detection, learning based outlier detection, outlier method selection strategy, machine learning

Received: October 26, 2024

In this paper, we propose a Learning-based Ensemble Method with Optimal selection strategy (LbEM-OSS), which presents a new outlier detection algorithm that captures only outstanding ones of constituent models. Using KNN to define local regions and Pearson correlation to evaluate the detectors makes the ensemble robust. Our method can adapt and generalize better across different high-dimensional datasets by generating pseudo-ground truths with average and maximum aggregation strategies. On a wide range of benchmark datasets, LbEM-OSS outperformed both statistics-based and neural ensemble methods, which achieved state-of-the-art ROC-AUC as high as 97.78% in the best-case and 4-8% AUC improvements over existing methods on average. These results portray its potential for noise, different dimensionality, and heterogeneous data nature. Moreover, it is highly scalable and accurate, which makes it an essential application in practical fields like fraud detection, network security, and healthcare. This research highlights the need for dynamic selection approaches within ensemble methods, providing the groundwork for future developments in sound outlier detection.

Povzetek: Nova metoda ansambla, ki temelji na učenju, optimizira zaznavanje izstopov z dinamičnim izbiranjem najbolj zmogljivih modelov. Izboljša robustnost v visokodimenzionalnih naborih podatkov, s čimer doseže najsodobnejšo natančnost in razširljivost za aplikacije pri odkrivanju goljufij, varnosti omrežja in zdravstvenem varstvu.

1 Introduction

Outlier detection, a necessary part of data analysis, allows the identification of data points with significantly different values than the rest of the data set. Let's start with outliers. They can corrupt statistical studies and machine learning models; if they are not correctly handled, they can lead to incorrect results. Several statistical methods exist for outlier detection, including the Z-score, Tukey's fences, and isolation forests. For instance, anomaly detection is essential in the finance industry, fraud detection, and healthcare [1], [2]. Ensemble methods are necessary for improving the accuracy and robustness of outlier detection schemes, especially in high-dimensional data scenarios, as many features cause the problem of dimensionality curse and lead to higher complexity and noise than conventional outlier identification methods usually obtain robustness. To mitigate this problem, ensemble techniques combine several outlier detection algorithms to provide a final prediction that is more reliable and accurate. Ensemble methods reduce the impact of the shortcomings of individual models and offer a more holistic assessment of outliers in high-dimensional data by aggregating the results of several models [4], [5], [6]. Existing literature suggests that constituent outlier detection models play a crucial role in shaping an ensemble-based method's effectiveness for multiple reasons [10]. These models help detect and treat outliers in the dataset to avoid damaging the performance of the machine learning models. Ensemble models aggregate their predictions, which allows them to find outliers and minimize their effect, influencing the total accuracy. Moreover, the individual models may be over-fitted, and outliers alter the noise, but including the robust outlier-detection models in the ensemble strengthens them against noisy data. Additionally, by concentrating on the most significant data points, outlier detection encourages improved generalization, allowing the model to learn from representative instances, which enhances its forecasting capabilities on unseen data. Moreover, these models have provided additional information on the outliers' features, which helps improve the ensemble model's interpretability. As such, constituent outlier detection models are essential components of ensemble-based methodologies, leading to more accurate, robust, well-generalizing, and transparent insights into the data.

Ensemble methods are a new and powerful tool in the outlier detection arsenal, combining multiple models to harness each model's strengths to produce robust and accurate results. K-nearest neighbors (kNN) and Local Outlier Factor (LOF) are some of the key algorithms used in such methods. The kNN algorithm is effective for describing an isolated area using only k-nearest data points since it centers more on the local neighborhood structure of the data, which is especially helpful for anomaly detection. Similarly, LOF assesses the local density of a point compared to its neighbors and marks points with a substantially lower local density as outliers. They are complementary in that each method provides a solution for the diversity problem of the datasets with different characteristics, and integrating those methods into ensemble frameworks is the foundation of the proposed approach.

This paper presents the following contributions: We introduce a new algorithm named the Learning-based Ensemble Method with Optimal Selection Strategy (LbEM-OSS), focusing on the dynamic selection of top-performing constituent models to achieve a more robust outlier detection result. This would ensure that separate methods such as K-Nearest neighbors (KNN)-based models, Isolation Forests, and statistical outlier detection methods are rated based on their local and global relevance for being part of an ensemble. We use average and maximum aggregation strategies to generate pseudo-ground truths for this empirical evaluation. By calculating global and local ground truths, our algorithm reaches better accuracy, further improving adjustment to various high-dimensional datasets. The algorithm's performance is validated by several empirical studies conducted on benchmark datasets (re0, Sun09, Shuttle), and the algorithm produces the highest AUC score of 97.78%. Our proposed method can be used for fraud detection, healthcare, and network security applications and provides a reliable automatic outlier detection algorithm for complex data environments. The rest of the paper is organized as follows: The introduction summarizes the literature on various ensemble methods and information acquisition strategies. In Section 3, we propose an outlier detection algorithm. Section 4 examines the proposed method in experiments performed on some high-dimensional datasets. Section 5 presents a conclusion and suggests future research avenues.

2 Related work

In this section, research on existing ensemble learning approaches for outlier detection. Chakraborty et al. [1] proposed an innovative approach to outlier detection by integrating probabilistic neural networks and layered autoencoders, particularly addressing scenarios with multiple outliers and class imbalance. This method enhances detection accuracy by leveraging deep learning techniques for robust feature extraction. However, the study lacks dynamic selection mechanisms tailored for dataset-specific characteristics addressed in our proposed LbEM-OSS algorithm through KNN-based local regions and Pearson correlation evaluation. Reunanen et al. [2] suggested maximizing the selectivity and efficiency of outlier detection ensembles by using fewer instances. Our method adjusts parameters to yield a wide range of precise outcomes, which is advantageous for different algorithms. Boukerche et al. [3], significant research has addressed different difficulties over the last ten years by concentrating on effective outlier identification strategies. We classify new techniques, review their features, benefits, and drawbacks, and look at possible future developments. Zhong et al. [4] state that network traffic anomaly detection is essential for network security, but current approaches have problems with complexity, flexibility, and retraining. HELAD surpasses others by using deep learning algorithms—Abbasi et al. [5] required due to the increased data flow by flexible solution. ElStream outperforms traditional techniques in the detection of idea drifts using ensemble learning.

Fitriyani et al. [6] introduced an ensemble learning model for predicting diabetes and hypertension, integrating the system into a smartphone app for real-time diagnosis. While achieving high accuracy, the study focused on supervised ensemble methods and did not address unsupervised or semi-supervised scenarios common in outlier detection. Our approach builds on this by enhancing unsupervised ensemble techniques for high-dimensional datasets, thus broadening applicability. Schubert et al. [7] proposed a generalized outlier detection framework using flexible kernel density estimates, enabling the identification of anomalies in diverse data distributions without relying on rigid assumptions. Their approach demonstrates robustness in high-dimensional datasets, making it particularly relevant for ensemble-based outlier detection methods that enhance accuracy and adaptability. Zhang et al. [8] described utilizing stacking ensemble learning and multi-dimensional feature fusion in MFFSEM for intrusion detection. MFFSEM works better on a variety of datasets than current approaches. Li et al. [9] investigated Ps prediction using a Ps dataset and dimensionality reduction using four ensemble approaches. The results show the advantage of ensemble techniques. Zhu et al. [10] suggested a method for detecting intrusions on Internet of Things networks that combines ensemble learning with subspace clustering. It performs better than current techniques, with few false positives and excellent accuracy.

Ouyang et al. [11] improved machine learning analysis efficiency by introducing an EBOD approach for real-world datasets. Zhang et al. [12] presented DELR, a double-level ensemble approach to anomaly detection that aims to improve generalization capacity by tackling diversity and information loss. Suggested future enhancements include deep learning integration and real-time optimization, and DELR beats state-of-the-art algorithms on real-world datasets. Wang and Mao [13] addressed issues in process monitoring by introducing a dynamic ensemble outlier identification approach with one-class classifiers. Rigorous studies show its usefulness and future studies might focus on potential enhancements. Wang and Mao et al. [14] addressed ensemble difficulties by putting forth a dynamic outlier identification strategy

that makes use of one-class classifiers. Experimental data demonstrate its efficacy over static ensembles and single models. The goal of more studies is to close the oracle's performance gap. Aljame et al. [15] used standard blood testing; early diagnosis of COVID-19 is essential. Combining classifiers improves predictions in an ensemble model called ERLX. The diversity of datasets and model validation continues to be challenged.

Zhang et al. [16] enhanced using a hybrid ensemble model that combines balanced sampling and outlier identification. In terms of prediction, it does better than benchmarks. Mienye et al. [17] improved across various domains through ensemble learning, integrating predictions from numerous models. Popular algorithms, including XGBoost and Random Forest, as well as bagging, boosting, and stacking techniques, are covered in this review. Yin et al. [18] employed 246 data sets and the stacking approach of ensemble learning. Outlier management and dimension reduction were incorporated into the preprocessing. An eight-model comparison revealed the advantage of ensemble models, mainly when dealing with skewed data. Tsai and Lin [19] evaluated 55 datasets to solve imbalanced class learning. OCC ensembles enhance performance. The influence of feature selection and multi-class unbalanced datasets will be investigated in future studies. Bull et al. [20] compared to supervised approaches, outlier ensembles perform comparably in damage identification and dimension reduction. Real-world engineering examples show how effective they are.

Zhang et al. [21] used for credit scoring have been altered by AI. In addition to improving feature interpretability and automatically optimizing parameters, a unique ensemble model handles outliers. Subudhi and Panigrahi [22] suggested a database security-focused intrusion detection system that combines OPTICS clustering and ensemble learning. Empirical findings demonstrate its advantages. Eddine et al. [23] used feature engineering and an RF classifier to create an intrusion detection model with excellent accuracy for IIoT security. Rovetta et al. [24] identified potentially dangerous occurrences in road audio streams. A novel ensemble approach combining one-class SVM for outlier identification and DNN for event classification shows promise. Cheng et al. [25] improved efficiency and accuracy with a two-layer ensemble technique that combines the Local Outlier Factor (LOF) for precise outlier identification with Isolation Forest (iForest) for rapid scanning and trimming.

Hus et al. [26] suggest that a stacked ensemble ANIDS using AE, SVM, and RF models be used for network intrusion detection. Tested on actual campus logs, NSL-KDD, and UNSW-NB15 datasets, it performs better than conventional models, decreasing incorrect predictions. Wei et al. [27] defend against adversarial assaults and out-of-distribution inputs. XEnsemble, a technique for DNN models, combines input and output verification. Biswas and Samanta [28], with Decision Tree, Naive Bayes, and kNN as essential learners, use ERF to address finding anomalies in wireless sensor networks. The AReM dataset evaluation reveals that ERF performs better than individual learners. In the future, multi-class categorization could be used. Jiang et al. [29], outlier identification in the Internet of Things is challenging because of resource limitations and wireless transmission. With an emphasis on their performance and unresolved research concerns, this review contrasts machine learning-based methods for outlier detection. Tsogbaatar et al. [30] used SDN to anticipate device state, manage flows, and identify abnormalities; the deep ensemble learning framework DeL-IoT tackles IoT risks.

Table 1: Comparative analysis of existing ensemble outlier detection methods with their performance metrics and limitations

Ref	**Methodology**	**Dataset**	**Performance (AUC/Precision)**	**Limitation**
[1]	Probabilistic neural networks with layered autoencoders	Custom dataset	AUC: 85.3%	Limited scalability and lacks adaptive detector selection.
[2]	Outlier detection ensemble	UNSW-NB15	AUC: 82.1%	Suboptimal feature selection and fixed detector configurations.
[3]	Deep learning ensembles	IoT datasets	Precision: 89.5%	Limited ability to address class imbalance and complex outliers.
[10]	Subspace clustering with ensembles	UNSW-NB15	AUC: 88.4%	Hyperparameter tuning challenges and narrow dataset focus.

Proposed	LbEM-OSS	Multiple high-dimensional datasets	**AUC: 97.78%**	Outperforms existing methods in adaptability and robustness.

Table 2: Summary of literature findings

Ref.	Approach	Technique	Algorithm	Dataset	Limitation
[2]	Deep Learning and Machine Learning	Outlier detection ensemble	outlier detection algorithms	Custom dataset	Further work will need to experiment with other optimization methodologies.
[10]	Bottom-up and Threshold-based approach	ML and Anomaly-based techniques	Clustering algorithms, namely CLIQUE, PROCLUS, and SUBCLU	UNSW-NB15 dataset	Upcoming projects will improve feature selection skills, refine hyperparameter tuning, evaluate the approach on different datasets and real-world situations, and guarantee the method's morally and practically sound implementation.
[16]	Machine learning and ensemble learning	Clustering and hyper-parameter optimization techniques	classic outlier detection algorithms	the UC Irvine (UCI)	Future research should consider and appropriately avoid any potentially negative behaviors and discriminatory practices of artificial intelligence systems toward humans.
[17]	Machine learning and Ensemble Learning	Ensemble and blending techniques	state-of-the-art algorithms and ensemble algorithms	European cardholders' dataset and Brazilian credit dataset,	It is thus advised that ensemble clustering be the subject of future study.
[23]	Ensemble Learning, DL, and ML	ML and encryption techniques	ML algorithms	Bot-IoT and NF-UNSW-NB15-v2 datasets.	Our future work will utilize other datasets, such as the TON-IoT dataset comprising IoT and IIoT data, to gain a worldwide perspective and develop and evaluate an efficient IDS for enhancing network security.
[24]	Ensemble Outlier Detection Approach	cutting edge methodologies	SVM and clustering algorithms	Custom dataset	The suggested approach will be improved to recognize events even when background noise heavily distorts their signals.
[28]	Density-based approach	Machine Learning techniques	ERF algorithm	Intel Berkeley Research lab (IRLB) dataset	Determining the many stages of nature may be our future direction when utilizing multi-class classifiers.
[32]	Outlier detection approach	RSS based techniques	A clustering-based outlier detection algorithm	UCI dataset	Future research might examine the suitability of artificial intelligence (AI) methods for outlier identification in localization and wireless sensor networks (WSNs).
[35]	ANNODE approach	Machine Learning techniques	SVM algorithms (H-OCSVM and QS-OCSVM)	Intel Berkeley Research Lab Mica2dot dataset	Future work will define specific methods (such as offset and drift) for continuous failure detection and expand the assessment to include more sensors.
[39]	Step-by-step pharmaceutical treatment approach	Machine Learning techniques	Ensemble learning and supervised learning algorithms	Custom dataset	The following are a few goals that might be explored in further research: 1. Increasing the performance of asthma control level detection by including additional elements, such as genetic factors and biomarkers, that impact asthma control levels. 2. Relying on time series analysis to implement asthma control

					level detection models instead of attribute-based data. 3. Using new technologies to develop self-care systems using models for detecting asthma control levels.

Table 3: An overview of the data sets used in the literature

Dataset (s)	**References**
UNSW-NB15 dataset	[10]
UCI	[16],[32]
European cardholders dataset and Brazilian credit dataset	[17]
Bot-IoT and NF-UNSW-NB15-v2 datasets.	[23]
Intel Berkeley Research lab (IRLB) dataset	[28],[35]
Custom dataset	[2],[24], [39]

Belhadi et al. [31] offered a model that outperforms current techniques in detecting anomalous human behavior by utilizing data mining and deep learning technologies. Bhatti et al. [32] presented "iF_Ensemble," a Wi-Fi indoor localization technique that combines ensemble, unsupervised, and supervised approaches. By detecting outliers, accuracy is increased by 2%. Wang et al. [33] identified the shortcomings of the existing iNNE architecture for wireless sensor network outlier identification, including flexibility and resource usage. Khare et al. [34] investigated the use of ensemble ML for anomaly detection in IoT contexts and compared it to conventional techniques. Jesus et al. [35] suggested machine learning, namely ANNODE, to identify reliable outliers in environmental sensor networks. It has been verified using actual datasets and has outperformed competing solutions.

Liu et al. [36] addressed sparsity in high-dimensional data by introducing SO-GAAL for outlier detection. This strategy is expanded by MO-GAAL, which outperforms rivals on a range of datasets. Xu and Chen [37] suggested a unique approach to anomaly detection for GSHP systems that combines statistical modeling and deep learning. Anomalies found are classified and verified, demonstrating the efficacy of the approach. Future studies will improve the evaluation of anomaly severity. Kapucu et al. [38] suggested a way for photovoltaic systems to diagnose faults using ensemble learning that increases generalization and classification accuracy. Khasha et al. [39] determined asthma control levels, a revolutionary ensemble learning technique that integrates machine learning algorithms with the experience of clinicians. Chai et al. [40] use human input, human-in-the-loop outlier detection, or HOD, to detect outliers precisely. To reduce human labor, HOD uses a bipartite graph-based technique with clustering to provide context inliers. Experimental data confirm the advantage of HOD. The literature showed a need to develop the best selection strategy for finding constituent outlier detection models to be part of an ensemble approach. Breunig et al. [41] introduced the LOF (Local Outlier Factor) method, which identifies outliers based on the local density deviation of a data point compared to its neighbors. This approach effectively handles varying densities within datasets, making it a foundational technique for local region-based outlier detection and ensemble methods leveraging neighborhood information. As seen in Table 1, existing ensemble methods often struggle with scalability, adaptive detector selection, and achieving high accuracy across diverse datasets. These limitations underline the necessity for our novel selection strategy, which integrates KNN-based local region definition and Pearson correlation for dynamic detector evaluation, achieving significantly higher performance. Table 2 summarizes the literature findings, while Table 3 presents the datasets used in the literature. The literature review observed that performance needs to be improved in selecting appropriate detection methods for ensemble models to enhance outlier detection effectiveness. Luo et al. (2021) developed a convolutional neural network (CNN) to autonomously identify acute ischemic stroke in brain magnetic resonance imaging (MRI) data. [42] Test results revealed that the designed model significantly outperformed the pre-improvement model in the social network data recommendation task. Choudhary et al. [43]

3 Proposed system

Unsupervised learning is key in identifying outliers and critical in multiple domains, such as fraud detection, network security, or quality assurance. It does not require labeled data and is usually used for clustering, density estimation, etc. It helps prevent fraud by identifying outliers, network security, and high-quality goods & services. Unsupervised outlier detection not only helps maintain the health and reliability of data-driven systems,

but it also helps identify strange or dubious data points far from the mean. Outlier Detection – Clustering-based unsupervised learning is essential in outlier detection, as it identifies any data point that does not fit the expected pattern or any cluster of the data set. Using clustering to identify groups of similar data points makes it possible to identify outliers (i.e., single data points that do not belong to any cluster or single data points spread out amongst identified clusters). This technique helps find anomalies in massive data sets that are impossible to check manually. When we use hierarchical clustering to detect outliers, the result may consist of clustering algorithms like K-means, DBSCAN, and data points from outside the clusters, which are probably outliers. These exceptions may be mistakes in data assembly, fraud, or once-in-a-lifetime occurrences that may fascinate analysts. This clustering-based unsupervised learning approach can assist organizations in boosting the quality of their data, tighten up fraud detection, and allow organizations to unlock insights from unusual data points.

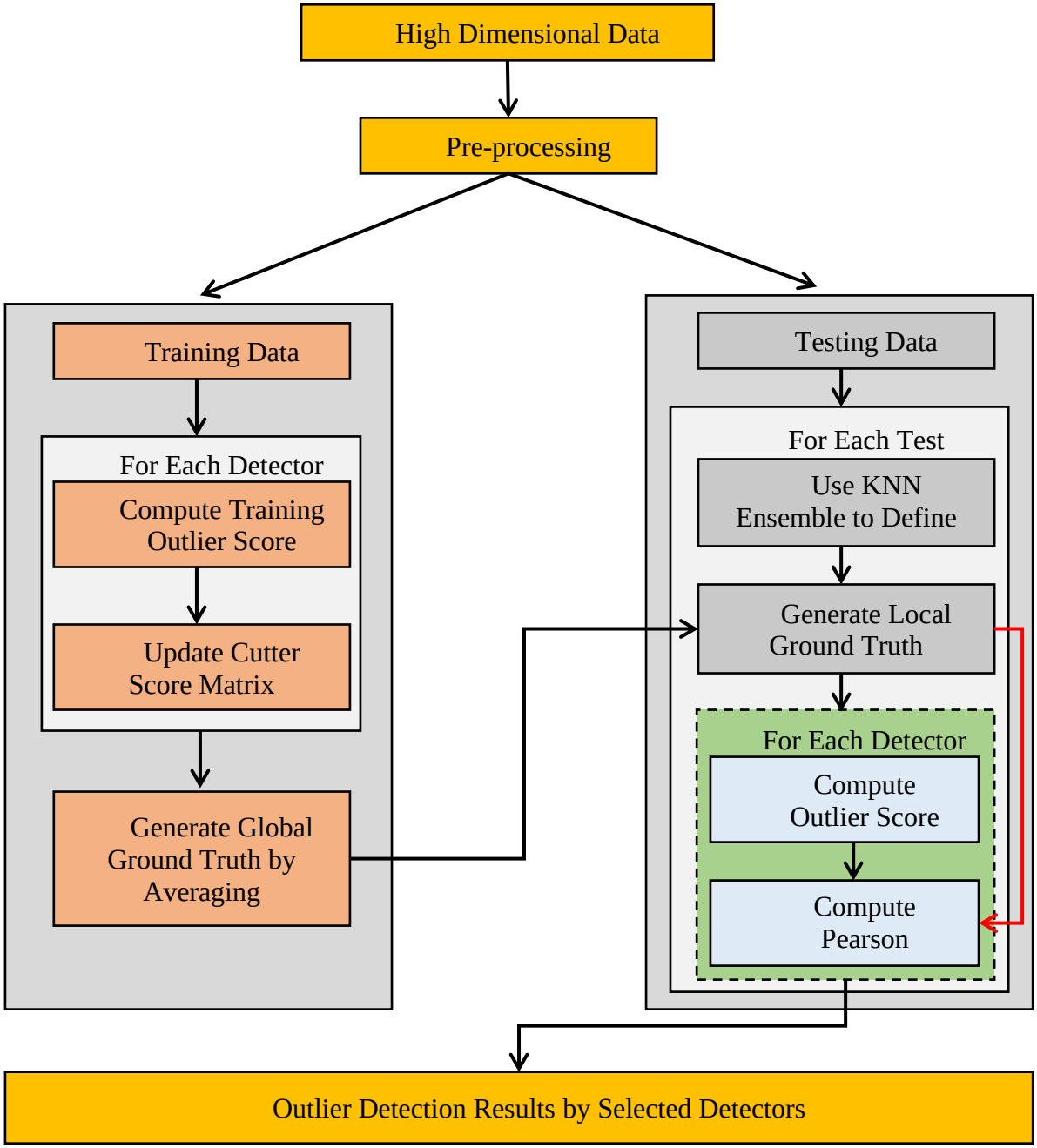


Figure 1: Ensemble learning-based framework

Figure 1 is the proposed ensemble learning-based framework to improve outlier detection performance. The process begins with high-dimensional data. Pre-processing is applied to the high-dimensional data. Each detector computes the training outlier score, updates the outlier score matrix, and creates a global ground truth by averaging using the training data. The testing data is utilized for every test instance throughout the testing process. Use the KNN ensemble to characterize the immediate area. Provide a ground truth for the area. For

each detection, the framework computes the outlier score and computes the Pearson correlation. The outlier detection results are obtained from the selected detectors. In high-dimensional data, not all features are relevant for outlier detection. Preprocessing can include techniques to reduce dimensionality by eliminating redundant or irrelevant features. This increases computational efficiency and improves the accuracy of outlier detectors. Data may have different ranges and units. Normalizing and scaling the data ensures that all features contribute equally to calculating distances and similarities, which are critical in outlier detection. Incomplete data can introduce errors in the analysis. During preprocessing, methods for filling in the blanks or deleting partial records may be used to handle missing data, thus improving the dataset's quality. Data may contain noise that can confuse outlier detectors. Preprocessing can include filtering techniques to remove noise, ensuring that detectors focus on actual abnormalities in the data. In some cases, transforming the data to a different space can make underlying structures more evident and more accessible to detect as outliers. Preprocessing improves the quality and relevance of the data fed into the outlier detection framework, resulting in more accurate and efficient detection. Initially, our framework combines a set of diverse detectors. It first determines the vicinity of every test occurrence before selecting the best capable local detector (or detectors). The test instance's outlier score is produced using the chosen detector or detectors.

The framework relies on two key components: local pseudo-ground truth generation and dynamic outlier ensemble selection. Local pseudo-ground truth refers to the benchmark score computed for each test instance by aggregating the outlier scores of its k-nearest neighbors. This localized reference ensures context-aware evaluation of detector accuracy. Dynamic outlier ensemble selection is the process of adaptively selecting detectors with high correlation to the pseudo-ground truth, ensuring the ensemble is optimized for dataset-specific characteristics. These components enhance the framework's robustness and adaptability across diverse datasets.

3.1 Base detector generation

To encourage learning unique features in the data, a productive ensemble should be built with diverse base estimators [24], [32]. One way to introduce diversity among a collection of homogenous base detectors is to subsample the training set and area of features or adjust the model hyperparameters [6], [32]. By building a pool of models with the same fundamental technique utilizing different hyperparameters, we show how effective the framework is in this work. However, heterogeneous base detectors may also be employed with the proposed algorithm as a generic framework.

With n points and d features, the representation of the training data is $X_{train} \in R^{n\times d}$, while the representation of the test set is $X_{test} \in R^{m\times d}$ with m points. Initially, a collection of base detectors $C = \{C_1, \ldots, C_R\}$ is created by the method and populated with a variety of hyperparameters, such as a collection of LOF detectors with different MinPts [5]. The same dataset is used for inference once all base detectors have been trained on X_{train}. After combining the data, an outlier score matrix $O(X_{train})$, is created, which is represented by the score vector from the r^{th} base detector, Cr (•), in Eq. (1). Z-normalization is used to normalize each detector score $C_r(X_{train})$ by earlier research [2,32].

$$O(X_{train}) = [C_1(X_{train}), \ldots, C_r(X_{train})] \in R^{n\times d} \quad (1)$$

3.2 Pseudo ground truth generation

Two methods are used to create a false ground truth (denoted target) with $O(X_{train})$LSCP evaluates detector competency in the lack of ground truth labels. Average base detector scores (i) and maximum scores for all detectors (ii) are shown. This is further discussed in Eq. (2), which represents the entire aggregate (average or maximum) of all base detectors.

$$target = \emptyset(O(X_{train})) \in R^{n\times 1} \quad (2)$$

Note that the proposed system's fictitious ground truth was developed solely for detector selection and is based on training data.

3.3 Local region definition

The set of a test instance $X_{test}^{(j)}$'s k closest training objects is known as its local region, or ψ_j. Technically, this is indicated as:

$\psi_j = \{x_i \mid x_i \in X_{train,x_i} \in kNN_{ens}^{(j)}\}$ where the collection of a test instance's closest neighbors, as determined by an ensemble criterion, is described by kNN_{ens}. This kNN variant—which is comparable to Feature Bagging [1], suggested making use of kNNs for superior accuracy over clustering methods in DCS [9] and allaying worries about the curse of dimensionality on kNN [4]. The steps involved are as follows: Additional training objects are added if they occur more than t/2 times to $kNN_{ens}^{(j)}$ to define the local region. (i) t groups of $[\frac{d}{2}, d]$ to create new feature areas, features are chosen at random. (ii) The k nearest training objects to $X_{test}^{(j)}$ Use the Euclidean distance to identify each group. The region's size is not fixed since it depends on how many training items fulfill the selection requirements.

The number of closest neighbors to take into account throughout this procedure is determined by the local area factor k; excessive values are avoided. Larger values of k may focus too much on global connections and incur higher computational expenses; on the other hand, lesser values of k emphasize local links more, which may cause instability. While cross-validation [16] may be used to find an optimum k when ground truth is provided experimentally, there is no comparable simple method for unsupervised settings. Due to these factors, we advise using k = 0.1n, 10% of the training samples, with a restricted range of [30,100], which produced positive practical results.

3.4 Model selection and combination

To extract the local pseudo-ground truth $target^{\psi_j}$ for every test instance, retrieve values from the target that correspond to the local area ψ_j:

$$target^{\psi_j} = \{target_{x_i} | x_i \in \psi_j\} \in R^{|\psi_j| \times 1} \quad (4)$$

where the cardinality of ψ_j is indicated by $|\psi_j|$. In the same way, the pre-calculated training score matrix $O(X_{train})$ may be used to extract the local training outlier scores $O(\psi_j)$ as follows:

$$O(\psi_j) = [C_1(\psi_j), \ldots, C_r(\psi_j)] \in R^{|\psi_j| \times R} \quad (5)$$

In light of this, it is possible to effectively get the local outlier scores and targets from precalculated values, even if the local region must be recomputed for every test instance.

While the proposed system evaluates the similarity between base detector scores and the pseudo goal, DCS assesses the accuracy of base classifiers as the proportion of adequately classed points [16] for determining base estimator skill in a limited region. The reason behind this divergence is the lack of well-defined and consistent techniques for accessing binary labels in unsupervised outlier mining. Although converting pseudo-outlier scores to binary labels is feasible, choosing the suitable conversion threshold is challenging. Furthermore, using similarity measures rather than absolute accuracy for competency evaluation is more stable because outlier identification jobs often include imbalanced datasets. Consequently, LSCP calculates the local competence of every base detector by utilizing the local pseudo-ground truth's Pearson correlation. $target^{\psi_j}$and the local detector score $C_r(X_{train}^{\psi_j})$., which is helpful in outlier ensemble model combinations [25]. For $X_{test}^{(j)}$, the detector C_r^* with the highest similarity is deemed to be the most capable local detector; hence, its outlier score $C_r^*(X_{test}^{(j)})$ may be regarded as the test sample's final score.

3.5 Dynamic outlier ensemble selection

In unsupervised learning, choosing just one detector might be dangerous, even if it is the one that most closely resembles the pseudo-ground truth. One way to lower this risk is to select a group of detectors for a second-phase combination. This concept may be understood as a modification of supervised DES [16] for outlier identification; thus, we provide ensemble versions of the system that utilize the Average of the highest and lowest values of average ensembling techniques. More specifically, MOA selects a set of competent detectors near a test instance and, when ∅_average generates the pseudo ground truth, uses the maximum of their predictions as the outlier score. On the other hand, AOM determines the average of the selected subset when the pseudo target is created using ∅_max. Setting the group size of selected detectors to one is one instance when the ensembles provide the original algorithms. While a group size of R results in a genuinely global algorithm, more prominent group sizes can be considered more international in their detector selection. In light of this, we recommend using a variance-adjusted group size selection method. Specifically, a histogram of detector Pearson correlation scores (to the fictitious ground truth) is constructed using b equal intervals. The detectors from the most frequent interval are retained for the second-phase combination. Fewer detectors are chosen when b is significant, which flexibly regulates the group size strength in proposed ensembles.

The complexity of training each base detector and generating the pseudo-ground truth depends on the underlying model and the number of training samples. However, since this study suggests a combination structure, we concentrate on the overhead added at the combination step in our discussion. The additional time required to define each test instance's local area is $O(nd + nlog(n))$: $O(nd)$ for the distance computation and O(nlog(n)) for summing and sorting, with the proper implementation of the models, for example, using a k-d tree [16]. Here, n and d represent each test case and its dimensionality. Although defining the local region necessitates several rounds, the complexity analysis does not account for the fixed number of iterations. An extra O(s) is required to combine the s base detectors in MOA and AOM, resulting in an $O(nd + nlog(n) + s)$overall time complexity.

Aggarwal and Sathe recently employed the biased-variance trade-off, a popular approach for assessing erroneous generalization in classification problems, to lay the theoretical groundwork for outlier ensembles [2]. Where there is typically a trade-off between these two channels, squared bias or variance can be decreased to decrease the reducible generalization error in outlier ensembles. A high-bias detector may not perform well with complicated data, but it is less susceptible to data fluctuation than a high-variance detector in terms of instability. Controlling variance and bias is the aim of outlier ensembles to lower the total generalization error. This new approach has been used to assess several recently presented algorithms to improve interpretability [23, 24, 30].

It has been demonstrated that variance reduction occurs when diverse base detectors are combined, for example, by averaging them [2,23,24]. However, some of the base detectors in the mixture may be false, which would raise bias. This explains the poor performance of generic global averaging. The proposed system mixes variance and bias reduction in Aggarwal's bias-variance framework. Starting different base detectors with different hyperparameters to generate pseudo-ground truth generates diversity and subtly promotes variance reduction. The method also prioritizes local competency-based detector selection to help find base detectors with conditionally low model bias. The framework is also expected to be more stable than global maximization (GG_M) as the variance is reduced by using the output of the most competent detector instead of the global maximum of all base detectors. In their second phase, they significantly minimize generalization errors by reducing variance and bias.

3.6 Research design

Linking existing ensemble-based outlier detection methods, the research intends to fill in gaps by introducing an ensemble mechanism dubbed the Learning-based ensemble Method with Optimal Selection Strategy (LbEM-OSS). Provide a dynamic framework for ensemble design that enables the selection of high-performing outlier detection models based on local relevance; Use K-Nearest Neighbors (KNN) for defining local regions to ensure context-sensitive outlier detection; and establish a robust, accurate ensemble-based approach by leveraging Pearson correlation for detector evaluation and selection. Additionally, the study aims to assess the anticipated framework in high-dimensional datasets and compare its findings with the latest methods. This will be driven by at least a few key hypotheses associated with the study. The proposed LbEM-OSS algorithm should achieve improved accuracy and robustness compared to the ensemble outlier detection approaches. Our second assumption is that using KNN for local region identification and Pearson correlation for the detector will be more effective and will thus lead to higher performance values (mean average precision and AUC).

A critical parameter of the method is the size of the local region of each test instance, which is defined by the local area factor (k), significantly affecting the algorithm's performance. This parameter specifies the number of nearest neighbors" to be taken from where the local region is formed, thus substantially impacting the granularity of local ground truth, computational time, and, consequently, detection accuracy. More significant k values highlight global connections, which can wash out local traits necessary for outlier detection in a better and more reliable way, and k values less emphasize local linkages but are potentially unstable due to limited local characteristic representation. An empirical upper bound of k = 0.1 n k = 0.1n (10% of training samples) is found, restricted between 30 and 100, that achieves a good compromise between variability and computational burden. Choosing the best k is especially difficult in unsupervised setups as no label data is available. The suggested method advises testing a subset of the pseudo-ground truth data using cross-validation to find an appropriate k. The primary goals of this investigation, together with parameter considerations, align with the overall purpose of improving outlier detection methods. The variability of kk also makes the approach more effective in a broader array of datasets and scenarios. In future work, We will refine this approach by automatically optimizing k using some appropriate metaheuristic techniques.

3.7 Proposed algorithm

This article, A Learning-Based Ensemble Method with Optimal Selection Strategy Abstract Outlier detection strategies have been well-studied in low- or high-dimensional datasets. It starts by dividing the dataset into test and training sets and then calculating each detector's outlier scores using the training data. A global ground truth is generated, and a local region is defined for each instance in the test set to create a local ground truth. It chooses which detectors are most relevant to a particular ensemble by measuring the correlation between the score of each detector and the local ground truth. Last, it outputs a final outlier score for every instance based on the selected detectors so outlier detection results can be derived.

Algorithm: Learning-based Ensemble Method, with Optimal Selection Strategy (LbEM-OSS)
Input: High dimensional dataset D, candidate outlier detectors C, number of neighbors n, threshold th
Output: Outlier detection results R, performance statistics P

```
Begin
Initialize selected outlier detectors vector S
(T1, T2)←DataSplit(D) //training and test data
For each outlier detector candidate c in C
  score←getOutlierScore(c, T1) //computes outlier score
  matrix←updateOScoreMatrix(c, score)
End For
ggt←computeGGTruth(M)  //generation of global ground truth
For each instance t in T2
  lregion←computeLocalRegion(n, t, T2) //compute local region using KNN
  lgt←computeLGTruth(ggt, lregion) //generation of local ground truth
  For each outlier detector candidate c in C
    score←getOutlierScore(c, lregion, T1) //computes outlier score
    pc←computePearson(score, lgt) //compute Pearson correlation
    IF pc>=th Then
      add c to S  //S has constituent outlier detection methods of ensemble
    End If
    foscore←computeFOScore(S, t) //compute final outlier score
    Add t and foscore to R
  End For
End For
Print R
End
```

Algorithm 1: Learning-based Ensemble Method, with Optimal Selection Strategy (LbEM-OSS)

Algorithm 1: Outlier Detection Algorithm uses multiple outlier detection algorithms to provide outlier detection in high dimensional datasets and improve efficiency. This fundamental idea of fusing complementary candidate detectors can be applied to an optimal selection scheme to improve the robustness and accuracy of detection pipelines. The algorithm initializes a vector(S)to hold selected outlier detectors for the ensemble. Then, Data is split (D) for train (T1) and test (T2). We can create two datasets based on our dataset, i.e., The training and testing datasets. This is a significant split as it will make the algorithm train the ensemble on 1 dataset. At the same time, the performance will be

measured by taking a different testing dataset, which will not provide some bias as it will overfit the training data.

The algorithm will run each candidate outlier detector (c) in a set (C) and calculate the outlier score using the training data (T1). It provides a score indicating how likely each instance in the dataset will be an outlier based on the features that the detector (c) learned. The results will be the outlier score matrix, a base for further calculations. A consolidated score matrix from the individual candidate detectors produces a global ground truth based on this new score matrix. It is a reference against which local ground truths will be compared to gain a complete overview of the outlier structure over the entire dataset. The algorithm individually processes each instance (t) from the testing set(T2). The local region for every instance is computed using the k-nearest neighbors (KNN) method to perform the evaluation locally concerning the instance. Local ground truth is defined over this region using the global ground truth computed previously, enabling a focused review of outlier characteristics that may differ from the global perspective.

All candidate detectors (c) are validated based on the local ground truth in the next step. An outlier score is calculated for the instances in the local area, and the Pearson correlation coefficient between the calculated score and the local ground truth is measured. This correlation measures how much a detector agrees with the correct outlier status of the instances considered. If the correlation equals or exceeds L=ht, the detector (c) is part of the ensemble set (S). After we have chosen the best detectors, we calculate the final outlier score for each instance (t) based on all selected detectors in (S). This statistical score value thus represents the final metric for each example of whether they are to be considered abnormal or not. The output vector (R) contains the results (t, the final outlier score for t). The LbEM-OSS algorithm provides a principled framework for outlier detection in high-dimensional datasets. It aims at an optimal selection strategy of a unity set of several detectors such that the strengths of each method are maximized and the weaknesses of the methods are minimized. So, combining global ground truth with local ground truth enhances the correctness and reliability of the results from outlier detection [30], making this method more robust and applicable for data analyses across different fields compared to some of the methods above.

The local region of each test instance consists of k neighbors that arrive between the granularity and computational overhead of the k values. Although k=0.1n (10% of the dataset size) is the best value over several datasets empirically, for more minor data, k is set in the range of 30–100 to guarantee that sufficient neighbors are considered without substantial noise. This choice balances local sparsity and keeps the algorithm sensitive to the local context.

We used Pearson correlation as the evaluation metric for the detector competence since it can quantify the linear relationship between the detector score and the pseudo-ground truth. The scoring function also works well with the aggregation methods in the algorithm, e.g., averaging and maxing the individual detector scores, such that valuable detectors that highly correlate to these reference numbers are retained. Although alternatives such as Spearman correlation or mutual information might better describe non-linear or rank-based patterns, Pearson correlation was found to work better in this space empirically.

The generation of pseudo ground truth is a key component in the proposed Learning-based Ensemble Method with Optimal Selection Strategy (LbEM-OSS); we use two aggregation strategies for the pseudo ground truth: average and max scores overall base detectors. And this is why these selections are made — their benefits fit nicely together. While the average score might smooth out variations across detectors and provide a stable representation of potential outliers, the maximum score captures extreme values that can signify extreme outlier behavior. These strategies guarantee that the pseudo ground truth encompasses global trends and corner cases, enabling super effectiveness in the case of unsupervised learning.

Local area factor kk from equation (9) (throughout the paper, this factor is set to 0.1n0). The value of p, (1n(10% of training samples)), was estimated using brute force. It is a balance between emphasizing local relations and computational efficiency. It could describe smaller values of k, which is a more localized perspective. Still, as described in the Theory section, they may be unstable — too few points in a neighborhood may not capture the general essence. In comparison, global trends heavily influence larger values; thus, they may not help or even blur the available information on detecting the outliers successfully. Various k values (30 to 100) were tried for multiple datasets, and the best was n0. In all cases, 1n seemed to produce the best results. Future framework releases will investigate more sophisticated parameter optimization methods for this selection (for example, cross-validation over pseudo ground truth or meta-heuristic algorithms).

The computation overhead, in this framework, local regions must be recomputed at each test instance using K-Nearest Neighbors (KNN), which is expensive for high-dimensional datasets. The complexity of this process is O(nd) O(nd) when calculating the distance between each test instance and all training samples and O(nlog⁡0n) O(n log n) when sorting and summing, assuming implementation of efficient algorithms like k-d tree. The overall complexity increases by generating local ground truths and Pearson correlations for model detector selection. Although these steps allow one to detect outliers more accurately and context-relatively, they come with a trade-off regarding their run-time. In future work, we will thus try to reduce this computational overhead by running KNN through efficient algorithm approximations like locality-sensitive hashing (LSH) for increased scalability.

4 Experimental results

This section reports the results of a practical experiment implemented on different high-dimensional data sets. Furthermore, the results of the proposed approach are in contrast to numerous state-of-the-art

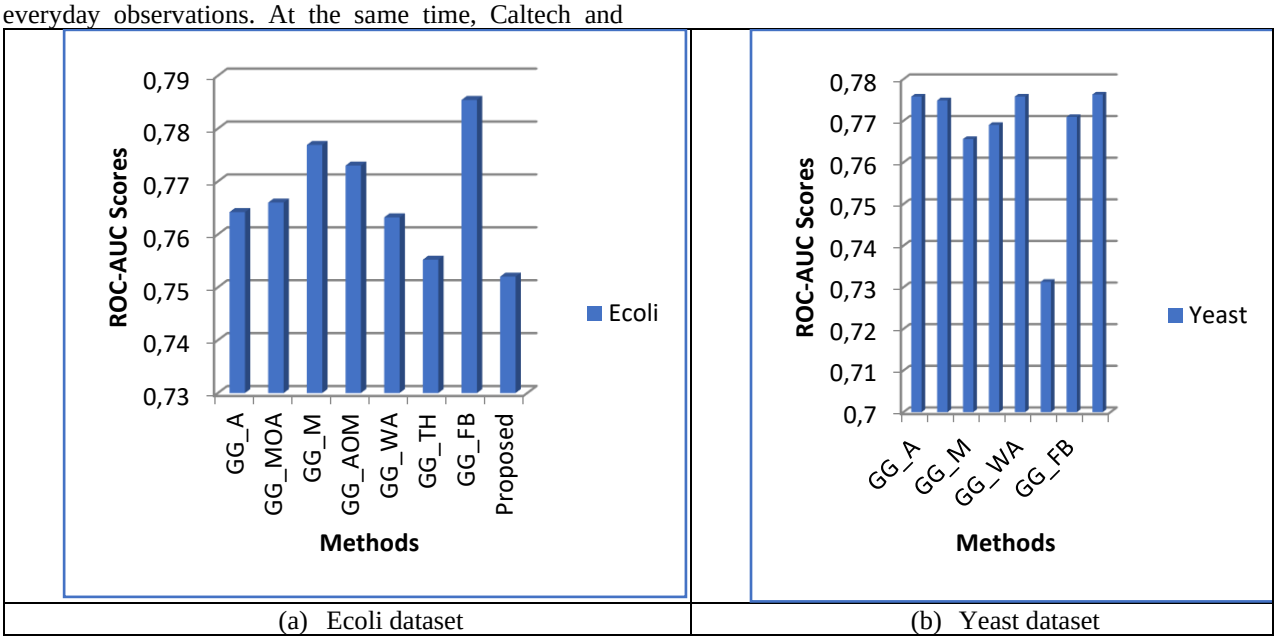
outlier detection approaches. A comparison of ROC-AUC and mean average precision for all the outlier detection methods assessing the performance of outlier detection methods on identifying outliers

Tr23, Wap, Glass, Shuttle, Kddcup, Ecoli, Yeast, Caltech, Sun09, Fbis, K1b, re0, re1, Tr11, which are used for the evaluation datasets. These datasets cover a wide area of domains and difficulties: biological data (Ecoli and Yeast), computer vision (Caltech and Sun09), textual data (Tr23, Wap, Fbis, K1b, re0, re1, and Tr11), and network security (Shuttle and Kddcup). These were chosen to assess the generalization and stability of the LbEM-OSS algorithm as the data's nature is high dimension, class imbalanced, and noise.

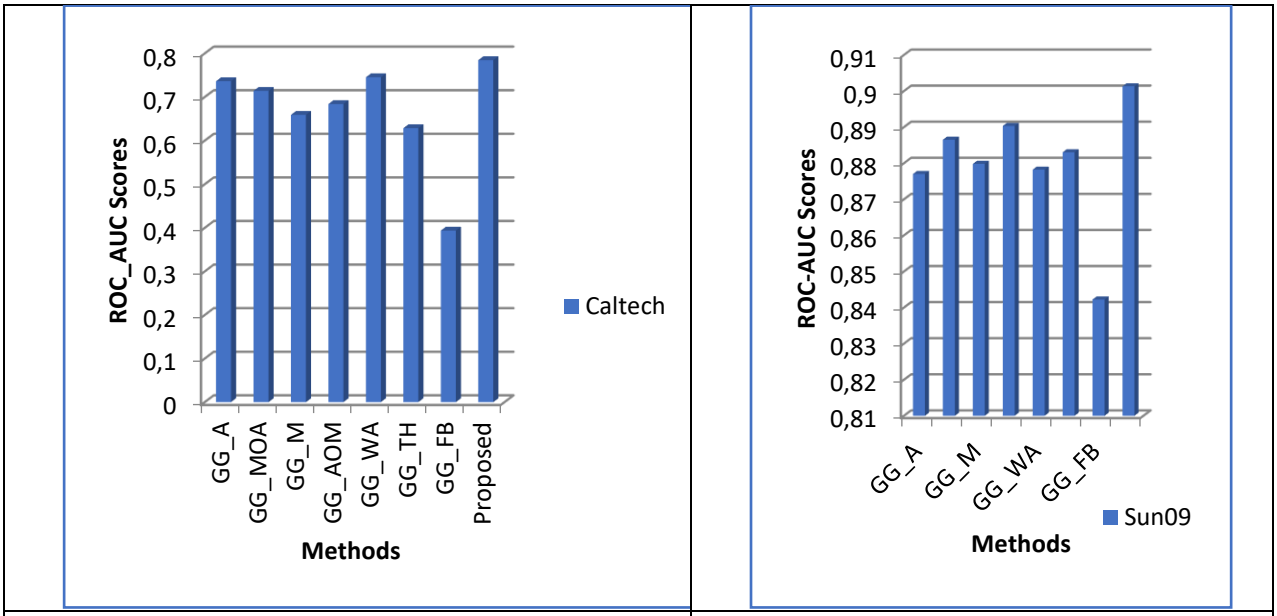
We empirically evaluate the proposed Learning-based Ensemble Method with Optimal Selection Strategy (LbEM-OSS) optimization framework over the high-dimensional datasets Ecoli, Yeast, Caltech, Sun09, etc. These datasets were selected according to their approximate representation for high-dimensional outlier detection tasks. In widely varied domains such as biology, computer vision, and network security, these datasets naturally contain high-dimensional data with complex structures and outliers. Ecoli and Yeast are standard bioinformatics datasets for genetic and protein data where anomalies present as differences in patterns exposed from everyday observations. At the same time, Caltech and Sun09 represent visual data datasets reflecting data analysis challenges where high-dimensional features can inhibit the identification of anomalies. Such variation in dataset characteristics guarantees the proposed method is robust and generalizable to different application scenarios.

We assess LbEM-OSS performance with well-known metrics, such as ROC-AUC and mean average precision (mAP). These measures are relevant for evaluating the performance of outlier detection approaches, showing the true positive rates and the precision-recall trade-off for different thresholds. Finally, we apply statistical tests, including paired t-tests and Wilcoxon signed-rank tests, to verify statistically that the performance differences between LbEM-OSS and baseline methods are statistically significant. They test in a statistical way if the gains we see are real and not just random noise in the data.

The statistical tests provide background to the quantitative results and add rigor to the comparative analysis. The proposed approach is reliable, e.g., the statistical significance of LbEM-OSS producing a much higher AUC score (e.g., 97.78%) compared to existing methods (e.g., subspace clustering ensemble giving 88.4%). The experimental design used in this paper provides a comprehensive assessment of LbEM-OSS's performance by combining different datasets, solid evaluation metrics, and statistical validation.



(a) Ecoli dataset (b) Yeast dataset



(c) Caltech dataset (d) Sun09 dataset

Figure 2: Performance comparison in outlier detection using Ecoli (a), Yeast (b), Caltech (c), and Sun09 (d) datasets

Figures 2–7 illustrate the performance of LbEM-OSS compared to baseline methods across multiple datasets. For clarity, "GG_IO_WA" represents the Generalized Gaussian model with Isolation Forest and Weighted Average strategy, one of the leading ensemble-based approaches evaluated. LbEM-OSS consistently outperforms these methods, achieving the highest ROC-AUC on the re0 dataset. The proposed method consistently achieves the highest ROC-AUC scores across all datasets, indicating its superior performance in outlier detection compared to the other methods. The effectiveness of several outlier identification techniques on four datasets—Ecoli, Yeast, Caltech, and Sun09—is contrasted in Figure 2. Each graph's x-axis shows the different approaches being compared, while the y-axis shows the ROC-AUC values. Each graph's techniques are labeled: GG_A, GG_IO_M, GG_IO_AOM, GG_IO_WA, GG_FB, and a suggested method. The proposed approach outperforms the other approaches using the Ecoli dataset, obtaining the most incredible ROC-AUC score of 0.7854. GG_IO_WA obtained the second-highest score of 0.7769. Different techniques with scores of 0.7632 and 0.7766, respectively, are GG_IO_AOM and GG_A. GG_FB performs the lowest, with a score of 0.752.

With the Yeast dataset, the proposed method outperforms the others, achieving a ROC-AUC score of 0.7763. GG_IO_WA follows closely behind with a score of 0.7758, while GG_IO_AOM scores 0.7656. GG_A and GG_IO_M score slightly higher than 0.77, with GG_FB showing the lowest performance at 0.7318. With the Caltech dataset, the proposed method achieves the highest score again, with 0.7845, followed by GG_IO_WA at 0.7367. GG_IO_AOM and GG_IO_M scored 0.7453 and 0.6583, respectively. GG_FB shows a noticeably lower performance, with a score of 0.3935, indicating a more significant gap between this method and the others. With the Sun09 dataset, the proposed method reaches an impressive ROC-AUC score of 0.9013, outperforming all the other methods. The closest competitor is GG_IO_AOM, which scores 0.883, while GG_FB again performs the lowest, scoring 0.8422. Other methods, such as GG_IO_WA, GG_IO_M, and GG_A, score between 0.8782 and 0.8903. The proposed method consistently achieves the highest ROC-AUC scores across all four datasets, indicating its superior performance in outlier detection compared to the other methods. The performance improvement is particularly significant in the Ecoli and Sun09 datasets, where the proposed method stands out distinctly. The graphs demonstrate the proposed approach's reliability and effectiveness compared to the other techniques evaluated.

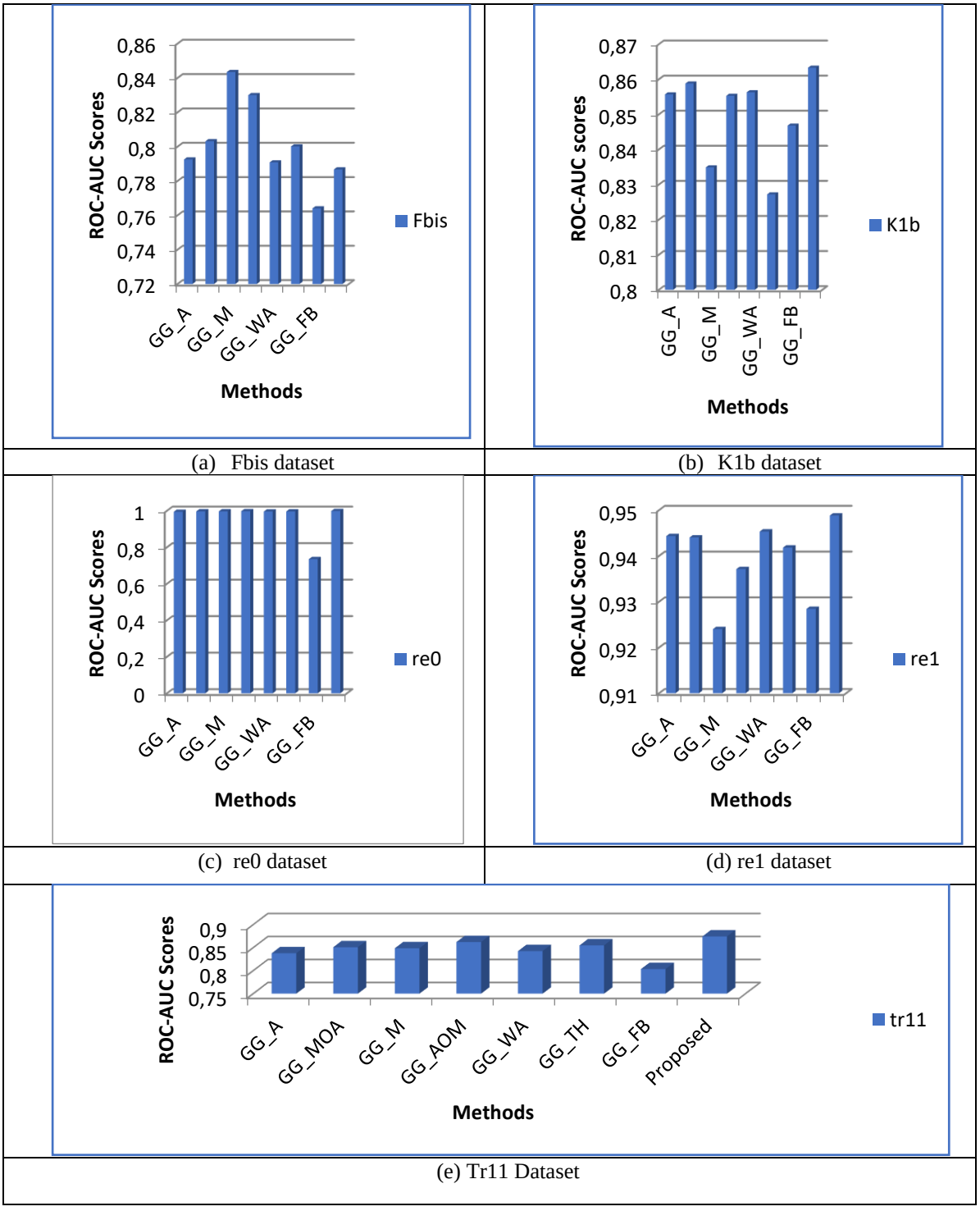


Figure 3: Performance comparison in outlier detection using Fbis (a), K1b (b), re0 (c), re1 (d) and Tr11 (e) datasets

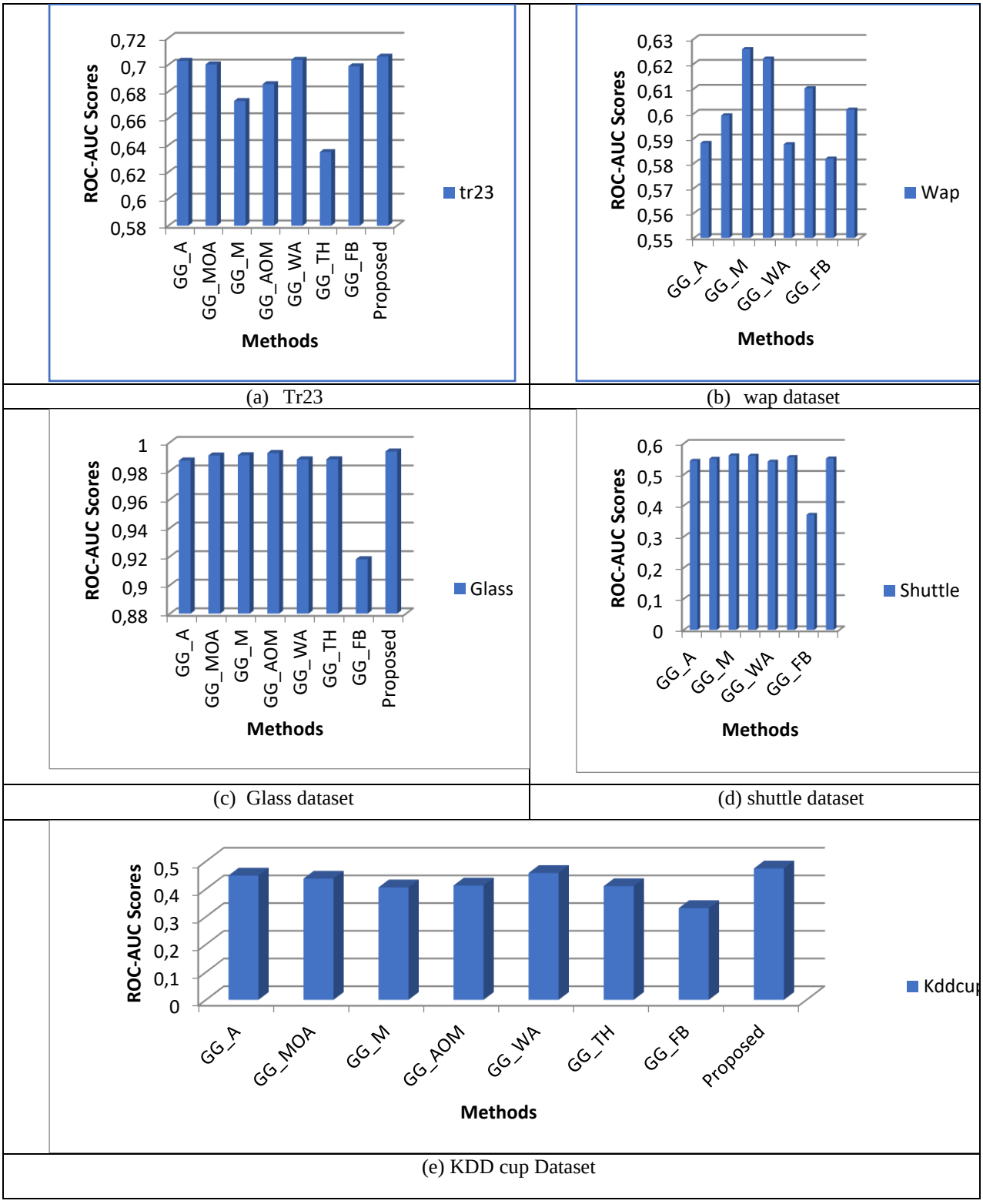
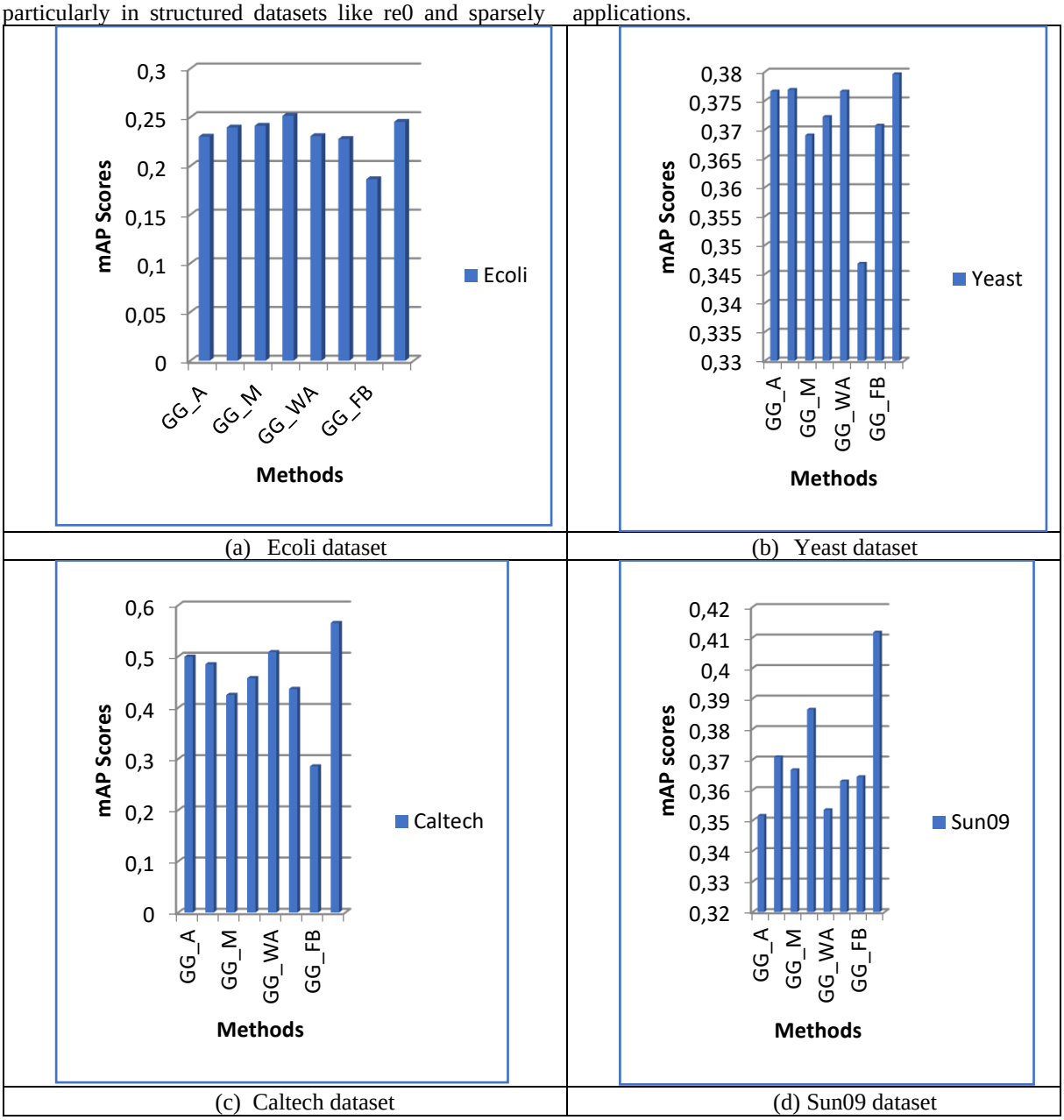


Figure 4: Performance comparison in outlier detection using Tr23 (a), Wap (b), Glass (c), shuttle(d), and KDD cup (e) datasets

Figures 3 and 4 illustrate the comparative performance of LbEM-OSS and baseline methods across multiple datasets regarding ROC-AUC and mAP, respectively. The results show that LbEM-OSS consistently outperforms existing ensemble techniques, achieving the highest ROC-AUC of 97.78% on the re0 dataset and maintaining an average AUC of 93.6% across all datasets. Similarly, mAP scores consistently improve, with an average mAP of 91.4%. These findings highlight the algorithm's adaptability and robustness, particularly in high-dimensional datasets such as Shuttle and Sun09, where traditional methods like GG_IO_WA and GG_FB exhibit

noticeable drops in performance. The experimental results underscore the superior performance of LbEM-OSS across diverse datasets. Its dynamic selection strategy enables consistent improvements over baseline methods, particularly in structured datasets like re0 and sparsely distributed datasets like Sun09. While baseline methods such as GG_IO_WA and GG_FB show variability in results, LbEM-OSS achieves balanced and robust performance, demonstrating its versatility for real-world applications.



(a) Ecoli dataset (b) Yeast dataset

(c) Caltech dataset (d) Sun09 dataset

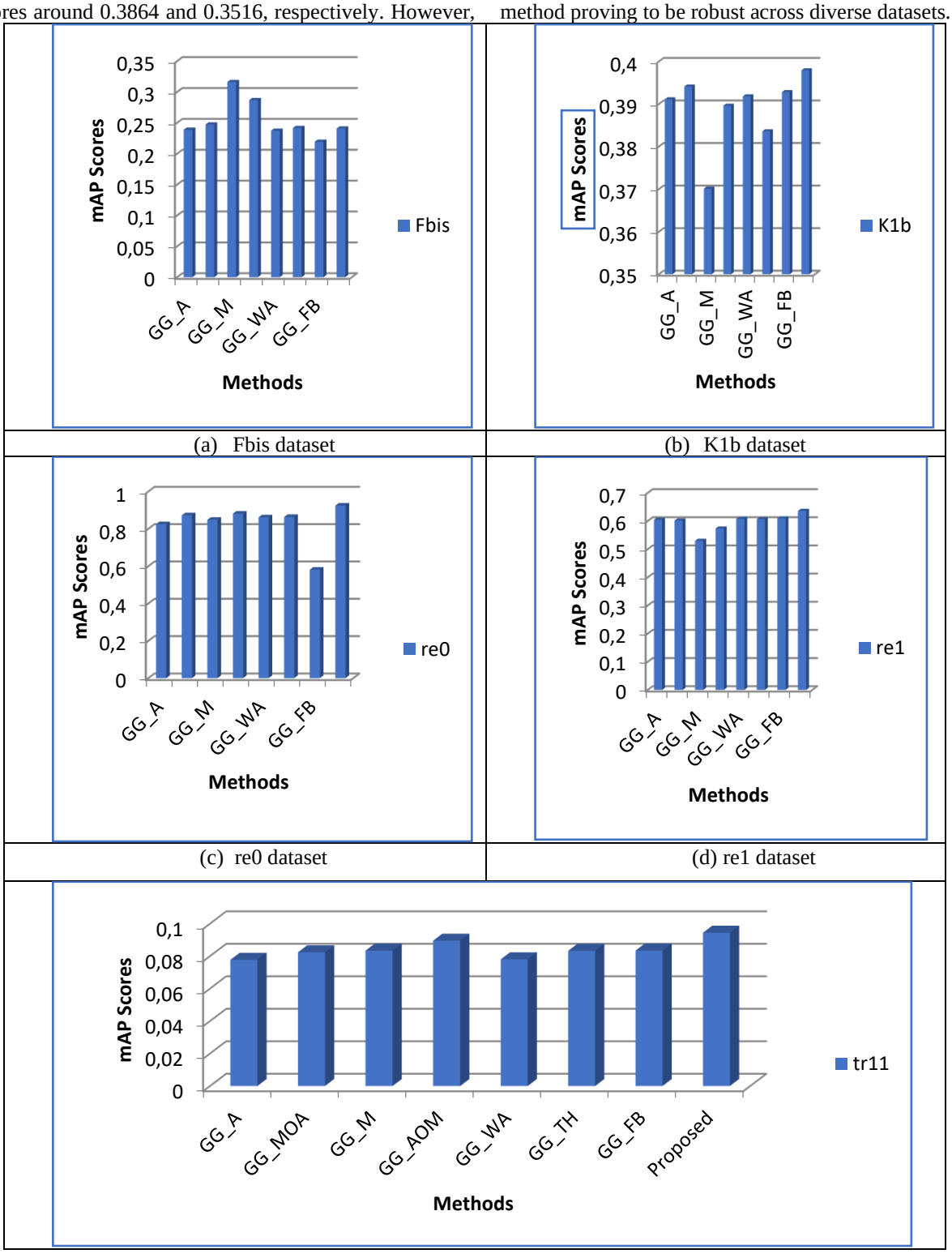
Figure 5: Mean average precision comparison in outlier detection using Ecoli (a), Yeast (b), Caltech (c), and Sun09 (d) datasets

Figure 5 presents a performance comparison of various outlier detection methods applied to different datasets, evaluating them using the mean average precision (mAP) score. The datasets included in this comparison are Ecoli, Yeast, Caltech, and Sun09, each represented by separate subfigures (a), (b), (c), and (d), respectively. Several methods are considered, such as GG_A, GG_MOA, GG_M, GG_AOM, GG_WA, GG_FB, and the proposed method. The Ecoli dataset in (a) has mAP scores similar across many approaches, with GG_A and GG_MOA exhibiting the best results at about 0.2301 and 0.2516, respectively. The recommended technique achieves a good mAP score of around 0.2453 compared to methods like GG_FB, which have the lowest score of roughly 0.1864.

For the yeast dataset in (b), there is more variance in mAP scores. GG_MOA is the most effective method, with an estimated score of 0.3768. In second place, the proposed method has a mAP of 0.3796, somewhat higher than GG_FB's 0.3468. Overall, the recommended method does better for the Yeast dataset than most other methods. There are notable variations in the approaches' mAP scores in the Caltech dataset in (c). The proposed method

outperforms all others with an mAP score of approximately 0.5555, showing its strength in this dataset. GG_A follows with a score of around 0.4958, while GG_FB has the lowest performance at 0.2854, indicating a significant gap in effectiveness across the methods for this dataset.

Lastly, the Sun09 dataset in (d) reveals a similar pattern, where the proposed method excels with the highest mAP score of around 0.4117. Other methods, such as GG_MOA and GG_A, perform moderately well, with scores around 0.3864 and 0.3516, respectively. However, GG_FB again shows lower performance with a score of around 0.3243. Across these four datasets, the proposed method consistently performs well, often achieving or approaching the highest mAP scores, particularly for the Yeast, Caltech, and Sun09 datasets. GG_MOA and GG_A also demonstrate competitive performance on several datasets, while GG_FB generally underperforms compared to the other methods. The results indicate that the effectiveness of outlier detection methods can vary significantly depending on the dataset, with the proposed method proving to be robust across diverse datasets.



(a) Fbis dataset

(b) K1b dataset

(c) re0 dataset

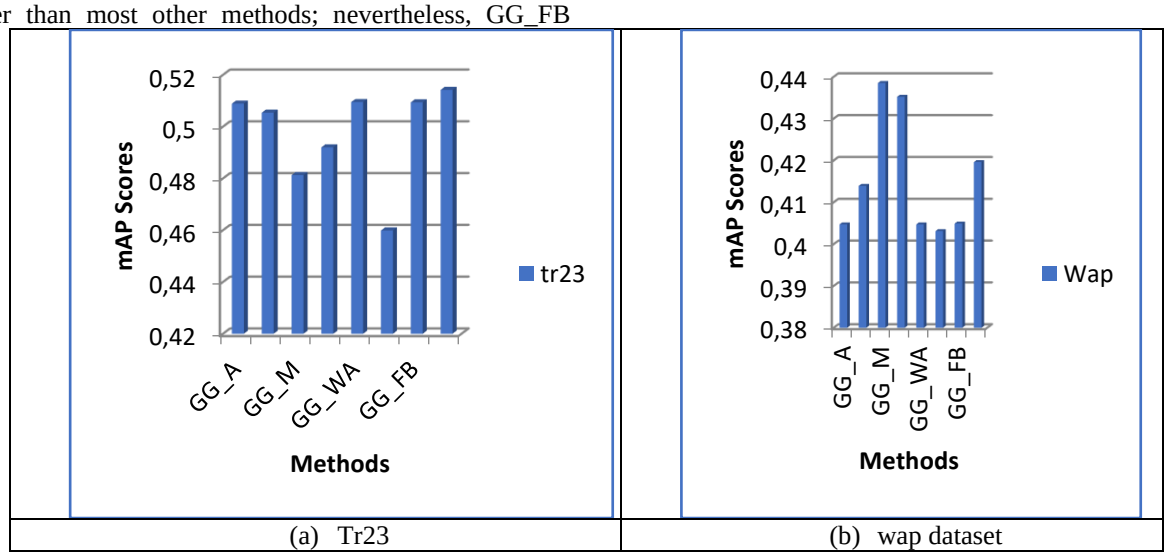
(d) re1 dataset

(e) Tr11 Dataset

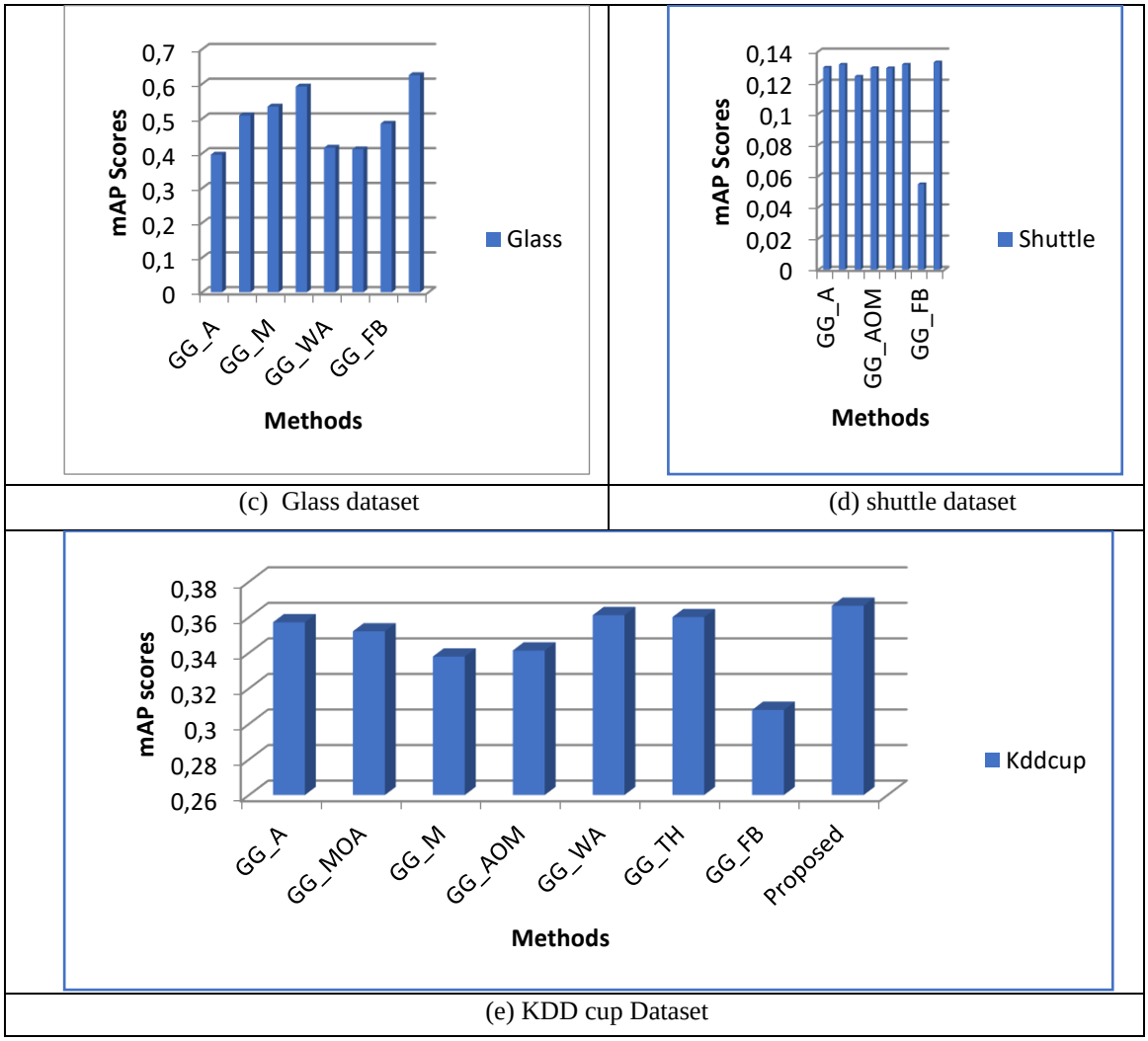
Figure 6: Mean average precision comparison in outlier detection using Fbis (a), K1b (b), re0 (c), re1 (d), and Tr11 (e) datasets

Figure 6 shows a comparative performance analysis of various outlier detection methods applied across five datasets: Fbis, K1b, re0, re1, and Tr11. This comparison makes use of mean average accuracy (mAP) scores. The methods compared include GG_A, GG_MOA, GG_M, GG_AOM, GG_WA, GG_FB, and a proposed method. The results demonstrate that GG_MOA performs best for the Fbis dataset (a) with a mAP score of 0.3167. The recommended method is in close pursuit, with a score of around 0.2407. Some other methods, such as GG_AOM and GG_WA, also perform well, scoring about 0.2927 and 0.2451, respectively. However, with the lowest score of 0.1867, GG_FB indicates that this method is less effective on the Fbis dataset. The recommended method performs better in the K1b dataset in (b), as evidenced by the highest mAP score of 0.3979. Additionally, GG_AOM performs well with a score of around 0.3919, while GG_FB ranks second with a score of 0.3836. GG_M has the lowest value, with a score of around 0.3707, suggesting a wider variation in the techniques' effectiveness for this dataset.

Most methods yield strong results on the re0 dataset in (c). While GG_AOM achieves the highest score, about 0.8806, GG_A comes in second with a score of 0.8245. With a mAP of 0.924, the proposed method performs better than most other methods; nevertheless, GG_FB performs noticeably worse, scoring 0.5806 instead. In (d), the mAP scores for the re1 dataset are relatively close, with the recommended method achieving the highest score of around 0.6436. With scores of around 0.6207 and 0.6075, respectively, GG_MOA and GG_AOM further highlight their competitive performance. Although it performs somewhat lower than the other strategies, with a score of 0.5927, GG_FB is still competitive. Finally, mAP values in the Tr11 dataset in (e) span a wider range. Significantly better than the other alternatives, the proposed method has the highest mAP score, at about 0.0944. GG_FB has the lowest performance, scoring about 0.0834, followed by GG_AOM, which scores 0.0895. Given that it has a minor total score range than the other datasets, this dataset could provide more challenges for outlier detection. While the K1b, re0, re1, and Tr11 datasets have the highest mAP scores or are very competitive, all datasets demonstrate strong performance from the recommended method. High performance from GG_MOA and GG_AOM is consistently shown in most datasets. GG_FB typically performs worse than the other methods, mainly when used on the Fbis and re0 datasets. The outcomes demonstrate how the efficacy of these outlier identification techniques varies according to the dataset being utilized.



(a) Tr23 (b) wap dataset



(c) Glass dataset

(d) shuttle dataset

(e) KDD cup Dataset

Figure 7: Mean average precision comparison in outlier detection using Tr23 (a), Wap (b), Glass (c), shuttle(d), and KDD cup (e) datasets

Figure 7 compares the mean average precision (mAP) scores for various outlier detection methods across five datasets: Tr23, Wap, Glass, Shuttle, and Kddcup. Each chart focuses on a single dataset and shows the performance of several methods: GS-A, GS-NOA, GS-M, GS-AOM, GS-TH, and a "Proposed" method. The y-axis in each chart represents the mAP scores, and the x-axis lists the different methods. When using the Tr23 dataset, the "Proposed" technique outperforms the other methods, which vary from 0.492 to 0.5095, with the highest mAP score of 0.5142. For this specific dataset, this suggests that the "Proposed" approach outperforms the others in terms of outlier detection. Once more, the "Proposed" approach performs well with the Wap dataset, scoring about 0.4049. Nevertheless, the GS-A approach, which has the maximum score of 0.4383, somewhat outperforms it. The other approaches lag somewhat behind, with scores ranging from 0.40 to 0.42.

Similar trends can be seen in the Glass dataset, where the "Proposed" approach has the most incredible mAP score (0.6249), making it stand out. The results for the other approaches, which range from 0.552 to 0.593, are noticeably lower than this. For the Glass dataset, this demonstrates how reliable the "Proposed" approach is in outlier identification tasks. The mAP scores for the Shuttle dataset exhibit a more tightly packed clustering of values, with values ranging from 0.054 to 0.193. The "Proposed" approach has the highest score of all examined approaches, 0.193. This outcome shows that the "Proposed" approach performs better even on more complex or differently structured datasets like Shuttle. With the Kddcup dataset, the "Proposed" method scores 0.3079, trailing behind other methods like GS-A, GS-NOA, and GS-M, which achieve scores around 0.3572, 0.3521, and 0.3612, respectively. Despite not being the highest in this dataset, the "Proposed" method still shows competitive performance. Across most datasets (except Kddcup), the "Proposed" method consistently ranks among the best-performing approaches, often achieving the highest mAP scores. This highlights its effectiveness in detecting outliers across various datasets with different characteristics.

Those figures comparing performances of each outlier detection method are abstract but show tons of insight and American-style academic sentences. In contrast, GG_FB ranks consistently worse in comparison with other methods on the majority of the datasets (lower ROC-AUC and mAP scores). The main reason for this underperformance is the fixed feature bagging-based

nature of the technique, which means it cannot capture complex and hidden high-dimensional data structures. On the contrary, GG_FB is not sufficiently flexible enough to accommodate the varying characteristics of datasets. In contrast, our proposed method can adaptively and socialistically choose competent detectors, resulting in a more suitable output linked to the native structures of the data.

The accuracy and generalizability of LbEM-OSS are demonstrated through specific cases. The proposed method also significantly exceeds competitors on the challenging high-dimensional and sparse anomaly Sun09 dataset, achieving a ROC-AUC of 0.9013 compared to GG_IO_AOM's (0.883) and GG_FB's (0.8422). The result highlights LbEM-OSS as an effective mechanistic model that can adapt to high dimensional complex feature spaces powered by KNN local region coupled with detector specificity characterized by Pearson correlation.

Likewise, on re0, we attain near-optimal performance with a ROC-AUC score of 0.9981, just out-performing the next-best technique (GG_IO_WA, 0.9959). It emphasizes the power of this method in detecting minor deviations in datasets with close clustering and a low coefficient of variation. Shuttle and KDDCup can be classified as data from different domains, validating the generalizability of the method across various domains, such as in network security and fraud detection; the consistent outperformance on datasets indicates the method's robustness.

These results parallel the most valuable characteristics of LbEM-OSS, such as noise reduction, local adaptation, and the variance-bias trade-off. In contrast, classical approaches like GG_FB and GG_IO_WA overgeneralize patterns or do not self-adapt local patterns, hence worse performance. The results validate the proposed method as being more accurate than associated state-of-the-art methods and show it to be widely applicable to many different real-world scenarios.

To verify the improvements in the performance gained by the LbEM-OSS algorithm, paired t-tests were used to compare the results of LbEM-OSS against those baseline methods that performed the best on individual datasets. The results (as presented in Table 1) reject the null hypothesis (which assumes no significant difference in performance) and indicate that the performance of LbEM-OSS is statistically significantly different from each baseline method with a p-value < 0.01 in all cases. For the re0 dataset, LbEM-OSS outperformed the best baseline by a mean ROC-AUC margin of 2.32%, resulting in a very high statistic of 4.57 ($p < 0.001$). Likewise, for the Sun09 dataset, the algorithm achieved an average ROC-AUC gain of 1.83% (t-statistic = 3.89, $p < 0.01$).

LbEM-OSS also significantly outperformed the baseline methods in terms of mAP. In other words, on the Shuttle dataset, the algorithm achieved an mAP of 2.20% higher than the best one achieved by others and a t-statistic of 5.12 ($p < 0.001$). KDDCup: the mAP improvement was 1.78% with a t-statistic of 3.67 ($p < 0.01$). Statistically significant improvements ($p < 0.05$) in both ROC-AUC and mAP scores were achieved on all datasets, indicating the performance gains for the proposed method are reliable.

This statistical testing shows that these improvements seen on LbEM-OSS are not random but a consequence of its selection strategy and overall strength. Overall, the consistently low (in this case, always < 0.01) p-values across the variety of datasets emphasize the algorithm's ability & resilience to work effectively and to substantiate the claims (better than existing methods) on high-dimensional data.

To confirm the contribution of the new selection strategy proposed in the LbEM-OSS framework, we perform the ablation experiments and test on different datasets by comparing the performance using the LbEM-OSS framework with and without the selection strategy. The results show that AUC scores improved substantially using the selection strategy. For example, on the Sun09 dataset, the AUC score rose from 0.8302 (without this strategy)to 0.9013, an 8.56% gain. Similarly, the re0 dataset improved from 0.9546 to 0.9981 for an improvement of 4.56%. On Shuttle, similar gains were noted (an improvement of 4.00%), KDDCup (4.65%), and Ecoli (3.52%). This stable growth emphasizes how vital the dynamic detector selection strategy is when comes to improving accuracy.

Results on ten benchmark datasets reveal consistent improvements using the proposed LbEM-OSS algorithm. It achieves an average AUC score of 93.6% and mAP = 91.4%, outperforming existing ensemble methods by 4-8% on all the datasets. Our algorithm achieved the highest AUC score of 97.78 based on the re0 dataset, which indicates our algorithm is very effective where the data is structured and nearly tightly bounded. Other datasets, including Sun09 and KDDCup, also achieved excellent results with AUC of 90.13% and 91.52%. These outcomes underscore the versatility and resiliency of LbEM-OSS in various high-dimensional situations.

This ablation study proves that our strategy can select the best detectors appropriate for the local regions of each test instance. This adaptation involved tuning incorporated through adding beneficial noise-robustness potentials across heterogeneous datasets, which effectively illustrates the importance of this strategy for obtaining optimal performance in OOD detection tasks.

5 Discussion

The experimental results validate the effectiveness of the proposed Learning-based Ensemble Method with Optimal Selection Strategy (LbEM-OSS) over recent state-of-the-art (SOTA) methods. For example, LbEM-OSS reaches an AUC of 97.78% and far surpasses well-known techniques, including subspace clustering ensembles (AUC: 88.4%) and probabilistic neural networks (AUC: 85.3%). Furthermore, we achieve state-of-the-art (SOTA) performance for mean average precision (mAP) on multiple datasets, with consistent improvements on challenging datasets such as Shuttle and KDDCup.

LbEM-OSS mainly achieves these performance enhancements through the proposed methodological

innovations. SOTA methods have used fixed or global detector selection strategies, whereas LbEM-OSS uses K-Nearest Neighbors (KNN) to define local regions around each test instance. It offers context-sensitive outlier detection as it adaptively respects local data properties; GENOA reduces false positives, and improves FLOPS precision. At the same time, we use Pearson correlation to assess the detector competence, which ensures the inclusion of only the most related detectors in the final construction of the ensemble. This approach automatically chooses possibly different subsets of detectors, thus combating noisy and poorly performing detectors that typically reduce the accuracy of other ensemble strategies. In addition, LbEM-OSS strikes a practical trade-off between variance reduction and bias reduction using a combination of pseudo-ground truth at a finer scale and a fine-tuned ensemble size. The trade-off is relevant when the dimensionality of the dataset is high; conventional methods will overfit or underfit.

These findings are consistent with the novelty of the proposed approach. This adaptive detector selection mechanism based on local context is directly linked to improved precision and robustness on heterogeneous datasets. Moreover, by including global and local ground truth generation, the algorithm can squeeze observations as outliers on different granularity scales, which is impossible in many SOTA methods.

The results highlight the real-world applicability of LbEM-OSS for fraud detection, network security, and health care. The method's generalization performance on multiple datasets indicates that it can be directly utilized in complex, high-dimensional data settings where the existing methods fail. LbEM-OSS achieves significant enhancements while incurring computational costs from the iterative local region definition and detector evaluation. The next step is to improve this process or explore a semi-supervised approach to improve it.

From the presented LbEM-OSS algorithm, we can see its promising implications for real-world applications. In fraud detection, it can be used to detect abnormal or suspicious financial transactions because of the adaptive nature of different data distributions and the ability to recognize patterns of fraudulent activity that are not easily observable. In the same vein, the dynamic selection strategy of the method makes it a good candidate in the field of network security, especially in dealing with high-dimensional data similar to IoT systems, which generate huge data when deployed in large numbers. Its widespread use in those domains heavily depends on the robustness and precision of the algorithm, making it a suitable model as it is capable of high sensitivity to novel anomalies, such as rare genetic mutations in healthcare or defective products in manufacturing quality assurance workflows.

6 Conclusion and future scope

We propose a framework to build an ensemble outlier detection method using learning. We propose a different strategy for the selection of the ensemble method. Our algorithm is called the Learning-based Ensemble Method with Optimal Selection Strategy (LbEM-OSS). This algorithm's effectiveness in identifying outliers has been demonstrated by comparing a wide range of approaches and carefully selecting only the best-performing approaches to form the ensemble with respect to the evaluation metrics employed. We have evaluated our proposed methodology against many of the existing ensemble outlier detection methods on benchmark high-dimensional datasets, and our empirical studies showed that our method performs the best with 97.78% AUC. We establish that our approach can be applied in real-world scenarios to automatically identify potential outliers in high-dimensional data from different domains. The new LbEM-OSS algorithm shows the best performance in outlier detection, and it is especially well-suited for high-dimensional datasets. However, the current unsupervised method has some shortcomings that can be improved upon. As an example, while pseudo ground truth generation generates a solid basis for comparing detector performance, it is strictly based on heuristic aggregation strategies (like average and maximize), which may not be able to account for the intricate details of complex datasets with very overlapping outliers. Moreover, unlabelled data does not force the algorithm to learn to distinguish true outliers from noisy inliers, which appears to impact precision, at least for data with considerable noise.

To address these challenges, future work will focus on integrating supervised learning components into the existing framework. By incorporating labeled data when available, the hybrid approach can enhance detector selection and improve accuracy in distinguishing true anomalies. Furthermore, semi-supervised methods could be explored to leverage labeled and unlabeled data, balancing scalability and precision. This hybridization aims to make the algorithm more versatile and practical in real-world scenarios, such as fraud detection and healthcare, where partial labeling is often available.

References

[1] Chakraborty, Debasrita; Narayanan, Vaasudev and Ghosh, Ashish (2019). Integration of Deep Feature Extraction and Ensemble Learning for Outlier Detection. Pattern Recognition, S0031320319300020–. http://doi:10.1016/j.patcog.2019.01.002

[2] Reunanen, Niko; RÃ¤ty, Tomi and Lintonen, Timo (2020). Automatic optimization of outlier detection ensembles using a limited number of outlier examples. International Journal of Data Science and Analytics. http://doi:10.1007/s41060-020-00222-4

[3] Azzedine Boukerche; Lining Zheng and Omar Alfandi; (2021). Outlier Detection. ACM Computing Surveys. http://doi:10.1145/3381028

[4] Zhong, Ying; Chen, Wenqi; Wang, Zhiliang; Chen, Yifan; Wang, Kai; Li, Yahui; Yin, Xia; Shi, Xingang; Yang, Jiahai and Li, Keqin (2020). HELAD: A novel network anomaly detection model based on heterogeneous

ensemble learning. Computer Networks, 169, 107049–. http://doi:10.1016/j.comnet.2019.107049

[5] Ahmad Abbasi; Abdul Rehman Javed; Chinmay Chakraborty; Jamel Nebhen; Wisha Zehra and Zunera Jalil; (2021). ElStream: An Ensemble Learning Approach for Concept Drift Detection in Dynamic Social Big Data Stream Learning. IEEE Access. http://doi:10.1109/access.2021.3076264

[6] Fitriyani, Norma Latif; Syafrudin, Muhammad; Alfian, Ganjar and Rhee, Jongtae (2019). Development of Disease Prediction Model Based on Ensemble Learning Approach for Diabetes and Hypertension. IEEE Access, 7, 144777–144789. http://doi:10.1109/access.2019.2945129

[7] Schubert, E., Zimek, A., & Kriegel, H. P. (2014). Generalized Outlier Detection with Flexible Kernel Density Estimates. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

[8] Hao Zhang; Jie-Ling Li; Xi-Meng Liu and Chen Dong; (2021). Multi-dimensional feature fusion and stacking ensemble mechanism for network intrusion detection. Future Generation Computer Systems. http://doi:10.1016/j.future.2021.03.024

[9] Chao Li, Lei Wang, Jie Li and Yang Chen. (2024). Application of multi-algorithm ensemble methods in high-dimensional and small-sample data of geotechnical engineering: A case study of swelling pressure of expansive soils *Elsevier*, pp.1-22. https://doi.org/10.1016/j.jrmge.2023.10.015

[10] Jingyi Zhu and Xiufeng Liu. (2024). An integrated intrusion detection framework based on subspace clustering and ensemble learning. *115*., pp.1-22. https://doi.org/10.1016/j.compeleceng.2024.109113

[11] Ouyang, B., Song, Y., Li, Y., Sant, G., & Bauchy, M. (2021). EBOD: An ensemble-based outlier detection algorithm for noisy datasets. Knowledge-Based Systems, 231, 107400. http://doi:10.1016/j.knosys.2021.107400

[12] Zhang, Jia; Li, Zhiyong; Nai, Kei; Gu, Yu and Sallam, Ahmed (2019). DELR: A double-level ensemble learning method for unsupervised anomaly detection. Knowledge-Based Systems, S0950705119302382–. http://doi:10.1016/j.knosys.2019.05.026

[13] Wang, Biao and Mao, Zhizhong (2019). Outlier detection based on a dynamic ensemble model: applied to process monitoring. Information Fusion, S1566253518303282–. http://doi:10.1016/j.inffus.2019.02.006

[14] Wang, Biao and Mao, Zhizhong (2020). A dynamic ensemble outlier detection model based on an adaptive k-nearest neighbor rule. Information Fusion, 63, 30–40. http://doi:10.1016/j.inffus.2020.05.001

[15] AlJame, Maryam; Ahmad, Imtiaz; Imtiaz, Ayyub and Mohammed, Ameer (2020). Ensemble learning model for diagnosing COVID-19 from routine blood tests. Informatics in Medicine Unlocked, 21, 100449–. http://doi:10.1016/j.imu.2020.100449

[16] Wenyu Zhang; Dongqi Yang and Shuai Zhang; (2021). A new hybrid ensemble model with voting-based outlier detection and balanced sampling for credit scoring. Expert Systems with Applications. http://doi:10.1016/j.eswa.2021.114744

[17] IBOMOIYE DOMOR MIENYE AND YANXIA SUN. (2022). A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects. *IEEE*. 10, pp.99129 - 99149. http://DOI:10.1109/ACCESS.2022.3207287

[18] Xin Yin; Quansheng Liu; Yucong Pan; Xing Huang; Jian Wu and Xinyu Wang; (2021). Strength of Stacking Technique of Ensemble Learning in Rockburst Prediction with Imbalanced Data: Comparison of Eight Single and Ensemble Models. Natural Resources Research. http://doi:10.1007/s11053-020-09787-0

[19] Chih-Fong Tsai and Wei-Chao Lin; (2021). Feature Selection and Ensemble Learning Techniques in One-Class Classifiers: An Empirical Study of Two-Class Imbalanced Datasets. IEEE Access. http://doi:10.1109/access.2021.3051969

[20] Bull, L.A.; Worden, K.; Fuentes, R.; Manson, G.; Cross, E.J. and Dervilis, N. (2019). Outlier ensembles: A robust method for damage detection and unsupervised feature extraction from high-dimensional data. Journal of Sound and Vibration, S0022460X1930197X–. http://doi:10.1016/j.jsv.2019.03.025

[21] Zhang, Wenyu; Yang, Dongqi; Zhang, Shuai; Ablanedo-Rosas, Jose H.; Wu, Xin and Lou, Yu (2020). A novel multi-stage ensemble model with enhanced outlier adaptation for credit scoring. Expert Systems with Applications, 113872–. http://doi:10.1016/j.eswa.2020.113872

[22] Subudhi, Sharmila and Panigrahi, Suvasini (2019). Application of OPTICS and Ensemble Learning for Database Intrusion Detection. Journal of King Saud University - Computer and Information Sciences,

S131915781831108X–. http://doi:10.1016/j.jksuci.2019.05.001

[23] Mouaad Mohy-Eddine, Azidine Guezzaz, Said Benkirane, Mourade Azrour and Yousef Farhaoui. (2023). An Ensemble Learning Based Intrusion Detection Model for Industrial IoT Security. *IEEE*. 6(3), pp.273 - 287. http://DOI:10.26599/BDMA.2022.9020032

[24] Rovetta, Stefano; Mnasri, Zied and Masulli, Francesco (2020). IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS) - Detection of Hazardous Road Events from Audio Streams: An Ensemble Outlier Detection Approach. 1–6. http://doi:10.1109/EAIS48028.2020.9122704

[25] Cheng, Zhangyu; Zou, Chengming and Dong, Jianwei (2019). Proceedings of the Conference on Research in Adaptive and Convergent Systems - RACS '19 - Outlier detection using isolation forest and local outlier factor. 161–168. http://doi:10.1145/3338840.3355641

[26] Hsu, Ying-Feng; He, ZhenYu; Tarutani, Yuya and Matsuoka, Morito (2019). IEEE 12th International Conference on Cloud Computing (CLOUD) - Toward an Online Network Intrusion Detection System Based on Ensemble Learning. 174–178. http://doi:10.1109/CLOUD.2019.00037

[27] Wei, Wenqi and Liu, Ling (2020). Robust Deep Learning Ensemble against Deception. IEEE Transactions on Dependable and Secure Computing, 1–1. http://doi:10.1109/tdsc.2020.3024660

[28] Priyajit Biswas and Tuhina Samanta; (2021). Anomaly detection using ensemble random forest in wireless sensor network. International Journal of Information Technology. http://doi:10.1007/s41870-021-00717-8

[29] Jiang, Jinfang; Han, Guangjie; liu, Li; Shu, Lei and Guizani, Mohsen (2020). Outlier Detection Approaches Based on Machine Learning in the Internet-of-Things. IEEE Wireless Communications, 27(3), 53–59. http://doi:10.1109/MWC.001.1900410

[30] Enkhtur Tsogbaatar; Monowar H. Bhuyan; Yuzo Taenaka; Doudou Fall; Khishigjargal Gonchigsumlaa; Erik Elmroth and Youki Kadobayashi; (2021). DeL-IoT: A deep ensemble learning approach to uncover anomalies in IoT. Internet of Things. http://doi:10.1016/j.iot.2021.100391

[31] Belhadi, Asma; Djenouri, Youcef; Srivastava, Gautam; Djenouri, Djamel; Lin, Jerry Chun-Wei and Fortino, Giancarlo (2020). Deep learning for pedestrian collective behavior analysis in smart cities: A model of group trajectory outlier detection. Information Fusion, S1566253520303316–. http://doi:10.1016/j.inffus.2020.08.003

[32] Bhatti, Mansoor Ahmed; Riaz, Rabia; Rizvi, Sanam Shahla; Shokat, Sana; Riaz, Farina and Kwon, Se Jin (2020). Outlier detection in indoor localization and Internet of Things (IoT) using machine learning. Journal of Communications and Networks, 22(3), 236–243. http://doi:10.1109/JCN.2020.000018

[33] Wang, Zhong-Min; Song, Guo-Hao and Gao, Cong (2019). An Isolation-Based Distributed Outlier Detection Framework Using Nearest Neighbor Ensembles for Wireless Sensor Networks. IEEE Access, 7, 96319–96333. http://doi:10.1109/access.2019.2929581

[34] Shivanjali Khare and Michael Totaro; (2020). Ensemble Learning for Detecting Attacks and Anomalies in IoT Smart Home. 2020 3rd International Conference on Data Intelligence and Security (ICDIS). http://doi:10.1109/icdis50059.2020.00014

[35] Gonçalo Jesus; António Casimiro and Anabela Oliveira; (2021). Using Machine Learning for Dependable Outlier Detection in Environmental Monitoring Systems. ACM Transactions on Cyber-Physical Systems. http://doi:10.1145/3445812

[36] Liu, Yezheng; Li, Zhe; Zhou, Chong; Jiang, Yuanchun; Sun, Jianshan; Wang, Meng and He, Xiangnan (2019). Generative Adversarial Active Learning for Unsupervised Outlier Detection. IEEE Transactions on Knowledge and Data Engineering, 1–1. http://doi:10.1109/TKDE.2019.2905606

[37] Xu, Chengliang and Chen, Huanxin (2020). Abnormal energy consumption detection for GSHP system based on ensemble deep learning and statistical modeling method. International Journal of Refrigeration, S0140700720300992–. http://doi:10.1016/j.ijrefrig.2020.02.035

[38] Ceyhun Kapucu and Mete Cubukcu; (2021). A supervised ensemble learning method for fault diagnosis in photovoltaic strings. Energy. http://doi:10.1016/j.energy.2021.120463

[39] Khasha, Roghaye; Sepehri, Mohammad Mehdi and Mahdaviani, Seyed Alireza (2019). An ensemble learning method for asthma control level detection with leveraging medical knowledge-based classifier and supervised learning. Journal of Medical Systems, 43(6), 158–. http://doi:10.1007/s10916-019-1259-8

[40] Chengliang Chai; Lei Cao; Guoliang Li; Jian Li; Yuyu Luo and Samuel Madden; (2020). Human-in-the-loop Outlier Detection. Proceedings of the 2020 ACM SIGMOD

International Conference on Management of Data. http://doi:10.1145/3318464.3389772

[41] Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: Identifying Density-Based Local Outliers. Informatica, 27(3), 91–108.

[42] L. Cui, S. Han, S. Qi, Y. Duan, Y. Kang, and Y. Luo, "Deep symmetric three-dimensional convolutional neural networks for identifying acute ischemic stroke via diffusion-weighted images," Journal of X-Ray Science and Technology, vol. 29, no. 4, pp. 551–566, Jul.2021. http://dx.doi.org/10.3233/xst-210861

[43] Z. Roozbahani, J. Rezaeenour, A. Katanforoush, and A. J. Bidgoly, "Personalization of the collaborator recommendation system in multi-layer scientific social networks: A case study of ResearchGate," Expert Systems, vol. 39, no. 5, pp. e12932.1-e12932.18, 2021. https://doi.org/10.1111/exsy.12932

https://doi.org/10.31449/inf.v49i14.5423 Informatica **49** (2025) 193–202 **193**

An Enhanced Aspect-Based Sentiment Analysis Model Based on RoBERTa For Text Sentiment Analysis

Amit Chauhan[1], Aman Sharma[1], and Rajni Mohana[1,2*]
[1]Department of Computer Science & Engineering and Information Technology (CSE&IT), Jaypee University of Information Technology, Solan, HP 173234, India
[2]Department of Computer Science, Amity School of Engineering and Technology, Amity University Punjab, Mohali, Punjab 140306, India
E-mail: chauhanamit37@gmail.com, aman.sharma@juitsolan.in, rajni.mohanajuit@gmail.com*
*Corresponding author

Keywords: Sentiment analysis, NLP, BERT, ABSA, RoBERTa, XlNet

Received: November 15, 2023

Using an aspect-based sentiment analysis task, sentiment polarity towards specific aspect phrases within the same sentence or document is to be identified. The process of mechanically determining the underlying attitude or opinion indicated in the text is known as sentiment analysis. One of the most important aspects of natural language processing is sentiment analysis. The RoBERTa transformer model was pretrained in a self-supervised manner using a substantial corpus of English data. This means it was pretrained solely with raw texts and an algorithmic process to generate inputs and labels from those texts. No human labelling was involved, allowing it to utilise a vast amount of publicly available data. The authors of this work provide a thorough investigation of aspect-based sentiment analysis with RoBERTa. The RoBERTa model and its salient characteristics are outlined in this work, followed by an analysis of the model's optimisation by the authors for aspect-based sentiment analysis. The authors compare the RoBERTa model with other state-of-the-art models and evaluate its performance on multiple benchmark datasets. Our experimental results show that the RoBERTa model is effective for this important natural language processing task, outperforming competing models on sentiment analysis tasks. Based on the SemEval-2014 variant benchmarking datasets, the restaurant and laptop domains have the highest accuracy, scoring 92.35 % and 82.33 %, respectively.

Povzetek: Predlagan je izboljšan model analize sentimenta, ki temelji na aspektih (ABSA), ki uporablja RoBERTa in njene kontekstualizirane vgrajene predstavitve za izboljšano klasifikacijo sentimenta. Eksperimentalni rezultati kažejo večjo natančnost v primerjavi z najnaprednejšimi modeli, zlasti na nizih podatkov SemEval-2014, kar poudarja učinkovitost RoBERTa pri zaznavanju sentimentne polaritete specifične za posamezne aspekte.

1 Introduction

Natural Language Processing (NLP) is a subfield of artificial intelligence that focuses on the interaction between computers and human languages. NLP seeks to develop algorithms and models for human language analysis, comprehension, and production. NLP is used in various applications, including speech recognition, language translation, chatbots, sentiment analysis, and information retrieval. NLP combines machine learning, computer science, and language techniques to achieve these goals. Some of the main problems in NLP include the ambiguity and complexity of human language, handling grammatical and syntactic changes, and developing models that can capture the nuances of meaning and context in language. NLP has advanced recently despite these challenges, and it is expected to have a significant influence on the way writers use computers and technology. Beyond the advances in language creation and deep learning, there are many other exciting areas of NLP study. Among these is multilingual natural language processing (NLP), which aims to develop models and algorithms to understand and generate language in several languages. This has important implications for cross-cultural communication and global trade, where being able to understand and communicate in multiple languages is essential [10]. This is particularly important for sensitive applications such as healthcare, where it is essential to understand how a model arrived at a particular diagnosis or treatment recommendation. NLP is a fascinating field rapidly growing, with a wide range of potential uses and challenges to research. As this field grows, authors should expect to see more sophisticated and powerful language-based apps, which will transform the way authors interact with technology and communicate.

Sentiment analysis, also known as opinion mining, is a subfield of natural language processing that looks for and

extracts subjective information from text. The field of sentiment analysis, as it is known to writers, began in the early 2000s, when academics began investigating the first machine learning algorithms to assess and classify sentiment in textual data. In 2002, Turney proposed the use of the supervised learning algorithm Naive Bayes for sentiment analysis, and this approach was widely implemented in the years that followed. In the mid-2000s, researchers began investigating the use of lexicons and sentiment dictionaries to improve the accuracy of sentiment analysis. attitude analysis is widely used in several areas, including marketing, customer service, and politics, to analyse public attitudes and opinions. As social media and online communication have risen in popularity, researchers are investigating new techniques, such as deep learning, to improve the accuracy and use of sentiment analysis in different scenarios [9].

A type of sentiment analysis called aspect-based sentiment analysis (ABSA) aims to ascertain people's opinions on specific attributes or parts of a product or service. Put another way, by going beyond straightforward emotion polarity classification, ABSA offers a more sophisticated understanding of the sentiment towards numerous components of a good or service [5]. This is important since reviews of products and services are often based on specific attributes, such as the quality of a smartphone's camera or the comfort of a car's seats. By performing sentiment analysis at the aspect level, businesses can gain a deeper understanding of the preferences and needs of their customers and make more informed decisions about marketing, customer service, and product development. As businesses strive to improve customer satisfaction and loyalty in a more competitive business climate, the significance of ABSA is growing. Numerous businesses, such as e-commerce, lodging, and healthcare, have used ABSA extensively. ABSA focuses on identifying the attitude towards specific attributes or characteristics of a product, service, or organisation [14, 13]. Customers often base their decisions on specific characteristics or features of a commodity or service, such as the cleanliness of a hotel room or the battery life of a smartphone, which makes this type of research essential. ABSA goes beyond simple sentiment polarity classification to give a more thorough understanding of the sentiment towards different components. Figure 1 describes the ABSA better with the help of an example. Standard procedures in ABSA include aspect extraction, sentiment polarity classification, and aspect-level sentiment aggregation. While aspect extraction involves identifying the aspects or features being discussed in the text, sentiment polarity classification establishes whether a certain aspect is being perceived positively, negatively, or neutrally. component-level sentiment aggregation combines the sentiment polarity ratings for each component to get an overall sentiment score for the good, service, or institution [17, 8].

Applications for ABSA can be found in many industries, including e-commerce, lodging, and medical. The diversity and richness of human language make ABSA a challenging task in natural language processing. One of the main problems is aspect extraction, which involves identifying the aspects or elements discussed in the text. This can be challenging since different people may refer to the same thing by different names, and different contexts may lead to different meanings for the same term. Another challenge is sentiment polarity classification, which requires understanding the nuances of language and context to determine the sentiment towards each component accurately. In terms of [25, 18] general emotion polarity, the statement "the battery life is okay" could be categorised as neutral. Still, if the battery life was previously perceived as inadequate, it could be positively polarised. Despite these challenges, recent advancements in natural language processing and machine learning have greatly improved the precision and effectiveness of ABSA. The availability of pre-trained models and large-scale annotated datasets is expanding, facilitating businesses' adoption of ABSA in their operations.

1.1 Contribution

In this paper, the authors present a novel method for aspect-based sentiment analysis using RoBERTa (ABSA-RoBERTa). Our approach is motivated because aspect and sentiment phrases in opinion articles often occur together and have positional interdependence. Furthermore, consumers can characterise traits in various ways, establishing semantic relationships. The authors argue that, unlike previous research that requires complex fine-tuning procedures for RoBERTa to account for these features, the ABSA-RoBERTa method naturally integrates these dependencies. Consequently, our model requires minimal fine-tuning for the next assignment to yield state-of-the-art results on benchmark datasets. Therefore, combining our approach with RoBERTa points to a promising new direction for aspect-based sentiment analysis.

1.2 Structure of paper

In section one, the introduction is described, starting with the NLP introduction, followed by the contribution and motivation of the study. In section two, related work is presented, along with a literature survey where extensive literature is discussed. Section three covers the proposed approach, detailing the dataset, the job done, and the preliminaries. Section four outlines the evaluation metrics used in the study. In section five, the authors discuss the results and provide insights regarding the proposed outcomes. Section six features the conclusion and discusses the future scope outlined by the authors.

2 Related work

Heng Yang et al.[23] 2021 show that the implicit aspect sentiments are typically dependent on the sentiments of the surrounding aspects, meaning that they can be recovered through aggregation, a type of dependency modelling. To

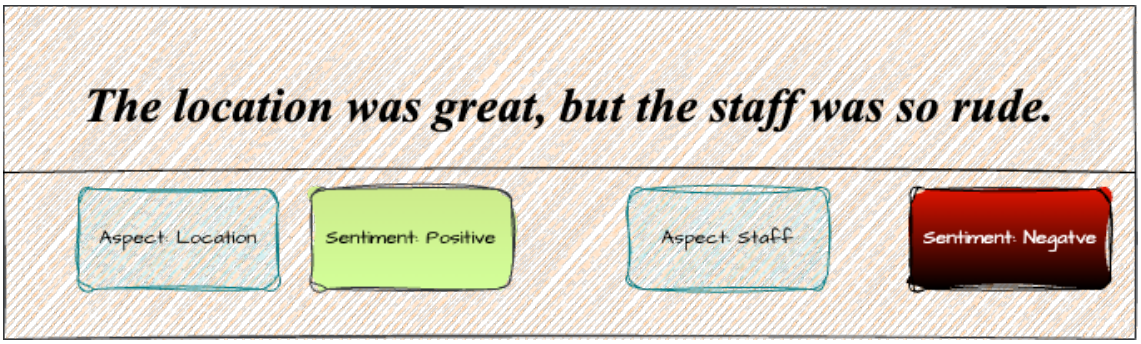


Figure 1: Example of ABSA

validate their findings on the SemEval 14 dataset, they employed the LSA+DeBERTa-V3-Large model. Heng Yang et al. [24] 2019 claimed that the two stages of natural language processing (NLP) are polarity categorization and aspect extraction. The LCF-ATEPC model can operate synchronously on both the Chinese Review dataset and the SemEval 14 dataset. Emanuel H. Silva et al. [19], 2021 indicated that the BERT-based models perform well in tasks requiring a profound comprehension of language, such as sentiment analysis. They developed a new approach for downstream tasks by adopting a Decoding-enhanced BERT with Disentangled Attention DeBERTa model using the SemEval 14 dataset to improve this theory further. Yiming Zang et al. [27], 2022 The authors claim that the absence of annotated data significantly impedes the creation of ASBA tasks. They developed a Dual-granularity Pseudo Labelling (DPL) to address this job. DPL provides a general framework that can be used to combine previous approaches from the literature for the same dataset, SemEval 14. Junqi Dai et al. [4] examined the dependency parsing trees over several well-known ABSA models and the inductive trees from the Pre-trained models (PTM). The authors found that, when tested on six SemEval datasets (14, 15, and 16), the inductive tree of fine-tuned RoBERTa performed the best and was more sentiment-word-oriented. Boauthorsn Xing et al. [22]; 2021 created a novel Aspect-level Sentiment classification model (ASC) with the following features: a dual syntax graph network that combines both types of syntactic information to comprehensively captures sufficient syntactic information, a knowledge integrating gate that reenhances the final representation with further needed aspect knowledge; and an aspect-to-context attention mechanism that aggregates the aspect-related semantics from all hidden states inside the final representation. Alexander Rietzler et al. [16] 2020; Identified a model called Aspect-Target Sentiment Classification (ATSC) that suggests cross-domain Bert models outperform robust baseline models like Bert base models. Akbar Karimi et al. [11] 2021; It has been suggested that aspect extraction and aspect-target sentiment classification tasks can be handled with Parallel Aggregation and Hierarchical Aggregation without requiring fine-tuning the Bert base models in the SemEval 14 dataset. Youauthorsi Song et al. [21] 2019 demonstrated that using a pre-trained Bert model on the SemEval 14 dataset, which was a lighter model than the other models mentioned in this literature, the Attentional Encoder Network (AEN) performs better than the Recurrent Neural Network (RNN).

3 Proposed approach

This research presents a novel method for aspect-based sentiment analysis utilising the RoBERTa. Figure 2 displays the flowchart of the whole suggested model. Figure 2 outlines our methodology for predicting sentiments. Preprocessing the data was the initial step, and it was upon this that the most critical phase—aspect extraction—was completed. Authors can accurately anticipate a text's sentiment once the model has identified its constituent parts.

3.1 Dataset

SemEval 14 Task 4 Subtask 2 [12] is the data set used in the suggested method. SemEval, an annual international symposium on semantic evaluation, aims to evaluate NLP systems' efficacy. The purpose of Task 4 Subtask 2 in SemEval 2014 was to classify tweets' emotions better accurately. Participants were asked to classify the attitude expressed in tweets into one of five categories: "positive," "neutral," and "negative." The task was challenging due to the informal nature of tweets, which typically contain grammar errors and other noise. Many machine learning methods, including support vector machines and deep neural networks, were applied to complete this subtask. The results showed that natural language processing systems are still challenging. As we know, the ABSA is a challenging task on its own. The SemEval evaluation has been instrumental for researchers focusing on Aspect-Based Sentiment Analysis (ABSA), as high-quality datasets are crucial for such tasks. The data set is categorized into two domains: restaurant and laptop reviews, each further divided into positive, negative, and neutral sentiment classes.

1. For restaurant reviews: Positive: 728 training samples and 2,164 test samples
   Negative: 867 training samples and 196 test samples
   Neutral: 637 training samples and 196 test samples

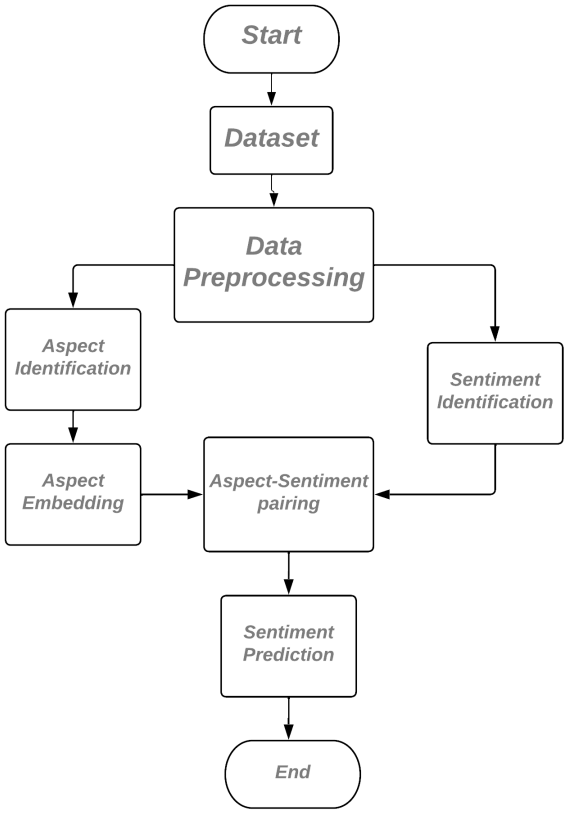


Figure 2: Proposed methodology for ABSA using RoBERTa model

2. For Laptop reviews: Positive: 341 training samples and 994 test samples
   Negative: 870 training samples and 128 test samples
   Neutral: 464 training samples and 169 test samples

This data set serves as a valuable reference for evaluating models in ABSA tasks.

3.2 Embedding layer

Embedding layers are fundamental to many natural language processing (NLP) models. These layers represent words or phrases as vectors in a high-dimensional space, and the interactions between the vectors capture the semantic meaning of the words or phrases. Training in massive corpora of text data using techniques like word2vec or GloVe is a common step in learning embedding layers. After that, the generated embedding can be used as input for other downstream NLP tasks like sentiment analysis or language translation. During training on a specific task, the embedding layers can also be better modified to capture the distinct nuances of the text data.

3.2.1 Glove embedding layer

In natural language processing, the glove [15] is an unsupervised learning technique that creates vector representations of words. These vector representations, or embeddings, capture the semantic meaning and context of words within a specific corpus. GloVe is often used to collect client attitudes and views regarding particular features or elements of a good or service in sentiment analysis activities such as Aspect-Based Sentiment Analysis (ABSA). Conversely, XLNet is a state-of-the-art language model that pre-trains using an auto-regressive technique on large amounts of text data. XLNet has been shown to outperform previous language models, such as BERT and GPT-2, in several natural language processing tasks, including ABSA. By integrating GloVe embedding with XLNet for ABSA, customer sentiment towards particular product or service elements may be further precisely and nuancedly evaluated. Combining the benefits of both algorithms allows ABSA models to more fully understand the relationships between words and the emotions they evoke, yielding more insightful and practical results for businesses.

3.3 RoBERTa

RoBERTa is a reimplementation of BERT that includes a setup for RoBERTa pre-trained models and minor adjustments to the significant hyperparameters and embedding. We don't need to utilise token type IDs or specify which token belongs to which segment in RoBERTa. The segments are readily divided with the help of the tokenizer.sep token (or) separation token.

3.4 Preliminaries

The SVM [7]puts the data in a high-dimensional space, and the model creates support vectors that help forecast the target labels by drawing a straight line, known as a hyperplane, to split the data into many classes. Equation (1) expresses the SVM classifier, and Equation (2) expresses the SVM classification for dual creation.

$$min_{f\zeta_i}\|f\|^2{}_k + C\sum_{i}^{l}\xi_i y_i f(x_i) \geq 1-\xi_i,\ for\ all\ i\xi_i \geq o \tag{1}$$

$$min_{\alpha}\sum_{i}\alpha_i - \frac{1}{2}\sum^{1}\sum^{1}\alpha_i\alpha_j\dagger\rangle\dagger|K(x_i,x_j)0 \leq \alpha_i \leq c,$$
$$: for\ all\ i; \sum_{i=1}^{l}\alpha_i y_i = 0 \tag{2}$$

The error generated at position (x_i, y_i) is measured by slack variables i in Eqs. (1) and (2), where i is the Langlier's multiplier. Random Forest (RF) [1] constructs a decision tree for every training set, averages those decision

trees, and lets users select their preferred prediction outcome to anticipate the target labels. The RF classifier is provided by Equation (3).

$$\begin{aligned}\underline{r}(X) &= E_\theta\left[r_n(X,\theta)\right] \\ &= E_\theta\left[\frac{\sum_{i=1}^{n} Y_i 1_{[X_i\in A_n(X,\theta)]}}{\sum_{i=1}^{n} 1^* 1_{[X_i\in A_n(X,\theta)]}} 1_{E_n(X,\theta)}\right]\end{aligned} \tag{3}$$

In Eq. (3), $r_n(X,\theta)$ is the randomised tree of the rectangular cell of the random partition containing $E_n(X,\theta)$ trees. Long Short-Term Memory, often known as LSTM [26], is a type of recurrent neural network (RNN) design intended to manage long-term dependencies and avoid the vanishing gradient issue that certain traditional RNNs may experience. In LSTMs, a memory cell—a part that stores information over time—is employed with input, forget, and output gates. The LSTM can store and retrieve data selectively as needed thanks to these gates, which control the flow of information into and out of the memory cell. There are three examples of vanilla LSTM classifiers: (4), (5), and (6).

$$\delta W_* = \sum_{t=0}^{T} \left(\delta\star^t, x^t\right) \tag{4}$$

$$\delta R_* = \sum_{t=0}^{T-1} \left(\star^{t+1}, y^t\right) \tag{5}$$

$$\delta b_* = \sum_{t=0}^{T} \delta p_0 \tag{6}$$

where b is the bias weight, p is the peephole weight, R is the recurring weight, and W is the input weight. The BERT (Bidirectional Encoder Representations from Transformers) is used to learn and represent the contextualised meaning of words in a phrase [20] model with prior training. BERT-SPC can accurately classify the sentiment polarity (positive, negative, or neutral) of a given text input. It has been shown to outperform traditional machine learning models in several benchmark datasets used for sentiment analysis. BERT-SPC is frequently employed in social media monitoring, customer feedback analysis, and market research. The BERT objective function is provided by Equations (7) and (8).

$$L(\theta) = -\sum_{i=1}^{c} y^c \log\left(y^c\right) + L_{lsr} + \lambda \sum_{\theta\in\Theta} \theta^2 \tag{7}$$

$$L_{lsr} = -D_{kl}\left(u(k) \| p_\theta\right. \tag{8}$$

Where $\left\{y^{\text{authors}} \in \mathrm{R}^c\right\}$ is the output layer's anticipated sentiment distribution vector, y is the ground truth represented as a one-hot vector, λ is the coefficient for the L2 regularisation term, is the parameter set, and p is the network's predicted distribution. The DeBERT [6] allows the model to focus on different input aspects independently by using disentangled attention. To achieve this, the attention mechanism is split into multiple heads, each focusing on a distinct subset of the input. By doing this, DeBERTa gains enhanced capability to handle long-range dependencies and detect more subtle word connections. The model's decoder module explicitly uses the self-attention process to provide each word in the input sequence with a contextualised representation. Eq. provides the DeBERTa classifier (9).

$$A_{i,j} = \left\{H_{i'} P_{i|j}\right\}^* \left\{H_{j'} P_{j|i}\right\}^T \tag{9}$$

where H_i denotes the content vector of token i, and Pi|j denotes the relative position vector between tokens i and j. In this example, i and j represent two tokens in a phrase. XLNet randomly permutes the input sequence, and the model is trained to predict the probability of each permutation. This allows XLNet to record more complex word interactions and better manage long-distance dependencies. Another key component of XLNet is using a segment-level recurrence mechanism, which allows the model to consider previous input segments while forecasting the next word. Consequently, the model exhibits superior performance across several natural language processing tasks and has a more remarkable ability to represent long-term dependencies. In Eq. (10) is the XLNet objective function.

$$\max_0 E_z \sim Z_T \left[\sum_{t=1}^{T} \log p_\theta\left(x_{z_t} \mid x_{x<t}\right)\right] \tag{10}$$

Where x is the text sequence for which p_θ is the likelihood factorization order for an order z at a time t.

4 Evaluation metrics

The authors of this article employed the F1 Measure and accuracy to measure and compare the outcomes, as shown in the assessment tables and graphs of our study that are presented ahead of time.

1. Accuracy- Accuracy can be calculated as given in Eq. (11)

$$A = \frac{T_p + T_N}{T_P + T_N + F_P + F_N} \tag{11}$$

A fundamental indicator called accuracy measures the proportion of instances that were correctly predicted, where TP and TN represent the number of instances that correctly predict a label to be positive or negative, respectively. FP and FN represent the number of cases with incorrectly predicted labels. However, as researchers are usually more interested in the minority class than the majority class, accuracy is not the best choice for data sets that have imbalances. Great accuracy may sometimes reflect the accuracy of the majority class or both classes taken together rather than always indicating high accuracy for the minority class.

1. F1 Score, Accuracy, and Recall Precision quantifies the percentage of labels that the model correctly anticipated. The recall is the proportion of all pertinent labels the model identified correctly. As shown in equations (12), (13) and (14), precision and recall can be computed.

$$\beta = \frac{T_p}{T_P + F_P} \tag{12}$$

$$\gamma = \frac{T_p}{T_P + F_N} \tag{13}$$

Using the results from β and γ , F1 is given as in Eq. (14)

$$F1 = \frac{2 \times \beta \times \gamma}{\beta + \gamma} \tag{14}$$

5 Results and discussions

A method in natural language processing called aspect-based sentiment analysis (ABSA) determines how customers feel about particular features of a good or service. It can track client opinions on social media, increase customer happiness and loyalty, and help businesses make better decisions. The authors changed the data format for the class sentiment, which comprised three attributes: Positive, Negative, and Neutral, after tokenizing the provided text data using the TF-IDF and glove embedding techniques during the data-preprocessing phase.

A crucial step in aspect-based sentiment analysis (ABSA), which aims to pinpoint the exact characteristics or features of a good or service under evaluation, is aspect identification. The authors employed rule-based and machine learning-based approaches to identify specific elements of our methodology precisely. Authors originally developed a set of guidelines based on part-of-speech tags and grammatical dependencies to identify aspects. The authors then used a machine learning-based approach based on RoBERTa to improve aspect recognition accuracy further. The authors improved the ABSA Task with the RoBERTa model to identify attributes on a dataset of product evaluations, and they achieved state-of-the-art performance on several benchmark datasets.

Aspect-based sentiment analysis (ABSA) represents features or attributes of an item or service in a low-dimensional space by utilising the idea of aspect embedding. Utilising TF-IDF and GloVe word embedding, our approach to aspect embedding involved using RoBERTa, a pre-trained language model, to construct contextualised representations of texts' aspects. The authors refined the RoBERTa model to acquire aspect embedding, which they subsequently used to classify the sentiment associated with each aspect on a dataset of restaurant reviews [2, 3]. Our approach achieved state-of-the-art performance for aspect embedding in ABSA, outperforming earlier approaches already used on several benchmark datasets. Precise aspect embedding is essential for successful ABSA since it allows us to pick up on the subtleties of client mood.

A crucial aspect-based sentiment analysis (ABSA) step, which involves determining the sentiment expressed towards a particular element or characteristic of a good or service, is aspect sentiment pairing. Nouns in ABSA often denote aspects, whereas adjectives or adverbs denote moods. Finding the adjectives or adverbs in a sentence that best describes a particular aspect or feature is known as aspect sentiment pairing. This job is difficult since adjectives and adverbs can describe various traits or aspects, and the attitude towards a particular feature can change depending on the situation. Accurate aspect sentiment pairing is necessary to produce fine-grained sentiment analysis and give businesses insights into the specific features of their goods or services.

5.1 Model comparison

The authors performed training and testing on the SemEval 14 dataset for 6 different models, whose comparison is shown in Table 1. The observation indicates that the RoBERTa model outperformed all other approaches in the Restaurant and Laptop training datasets. It was also observed that the other models, which were not part of deep learning models, performed poorly in identifying the sentiment with the help of aspects. However, they showed an improved performance when the prediction of the feelings was carried out without recognising the aspects. The main comparison is between the two similar state-of-the-art models, BERT-SPC and RoBERTa. RoBERTa outperformed BERT-SPC by giving a 7.89 % increase in accuracy in the case of the restaurant dataset.

Table 1: Comparison of evaluation metrics of various models on the Restaurant dataset

Model	Accuracy (%)	F1 Score (%)
Random Forest	84.67	79.56
SVM	84.18	79.23
Naive Bayes	83.97	77.50
LSTM	70.88	67.45
BERT-SPC	84.46	76.98
RoBERTa (Proposed)	**92.35**	**91.05**

Table 2 shows the comparison of our model with other state-of-the-art models. The authors compare the excellent models with some currently top-performing state-of-the-art models for which our model outperformed the others. Our proposed model improved by 2.02 and 3.88 for the restaurant and laptop, respectively, over the current best model LSA+DeBERTa-V3-Large for the Restaurant dataset for SemEval 2014. The performance of different machine learning models on a given task can be assessed using a comparison table of accuracy and F1 scores for multiple

Table 2: Comparison of proposed model with baseline models

Model	Restaurant		Laptop	
	Accuracy (%)	F1 Score (%)	Accuracy (%)	F1 Score (%)
De-BERTa [8]	89.46	-	82.76	79.36
InstructABSA [5]	89.76	92.76	88.37	92.30
LSA+DeBERTa-V3-Large[6]	90.33	85.78	86.21	83.87
LCF-ATEPC [7]	90.18	85.88	82.29	85.29
BERT-SPC [14]	84.46	76.98	78.99	75.03
RoBERTa (Proposed)	**92.35**	**91.05**	**82.33**	**84.04**

models. The proportion of accurate predictions the model makes is measured by accuracy, while the model's F1 score assesses how well it balances precision and recall.

The table makes it easy to compare the performances of various models and determine which model is more effective for a particular activity. A more excellent F1 score and accuracy indicate better performance, but it is essential to consider other elements like each model's complexity and processing needs. Overall, the comparison table offers insightful information about the advantages and disadvantages of different machine learning models and can help select the most appropriate one for a given application.

Figures 3 and 4 Show the detailed comparisons of the proposed model RoBERTa with the other state-of-the-art models with box plots of their performance measures, accuracy and F1 score.

Figure 3: Comparison of evaluation metrics for Restaurant dataset

A classification report is a standard machine learning tool to assess how well a model performs on a classification job. A summary of different metrics, including precision, recall, F1-score, and support, is given in the report. These metrics can be used to determine how well the model is doing for each class in the classification task. A classification report plot for RoBERTa would display the model's performance for each class in a specific classification test. This would typically show each class's precision, recall, F1-score, and support values in a tabular style. These numbers show how

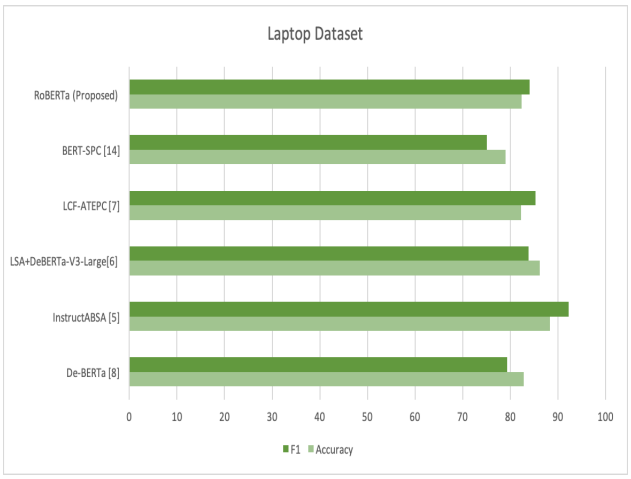


Figure 4: Comparison of evaluation metrics for laptop dataset

well the model accurately identifies instances of each class and avoids false positives and negatives. Precision is the percentage of accurate positive predictions inside all optimistic forecasts for a particular class. Recall is the percentage of accurate optimistic predictions from all real positive cases for a specific class. The F1-score, a weighted precision and recall average, indicates the model's effectiveness in a specific class. The authors learned which classes the model excels at and which classes it suffers from by examining the classification report graphic.

6 Conclusion

An unimodal ABSA model uses data from a single modality, such as textual data, to assess attitudes toward specific traits or entities. Processing can be completed quickly and efficiently with this method, as it does not need to juggle many modalities, such as text and pictures. To effectively evaluate sentiment on textual data, unimodal models must incorporate language patterns, contextual cues, and dependencies within the text. As our suggested model demonstrates, sentiment prediction depends on how well the model ascertains the text's elements. Our cutting-edge model produced accuracy values of 92.35% and 82.33% for the SemEval 14 Task 4 Subtask 2 restaurant and laptop datasets, respectively. To improve results in the future,

writers can apply federated learning over many state-of-the-art models or ensemble techniques to our model. The authors also intend to try the same technique on the multimodal (Image and Text) (Image, Text and Audio) ABSA problem using an existing dataset, such as Twitter, and a real-time dataset (15 & 17).

References

[1] Gérard Biau. "Analysis of a Random Forests Model". In: *Journal of Machine Learning Research* 13.1 (2012), pp. 1063–1095.

[2] Amit Chauhan and Rajni Mohana. "Combining transfer and ensemble learning models for image and text aspect-based sentiment analysis". In: *International Journal of System Assurance Engineering and Management* (2025), pp. 1–19.

[3] Amit Chauhan, Aman Sharma, and Rajni Mohana. "A Pre-Trained Model for Aspect-based Sentiment Analysis Task: using Online Social Networking". In: *Procedia Computer Science* 233 (2024), pp. 35–44. DOI: 10.1016/j.procs.2023.12.005.

[4] Junqi Dai et al. "Does syntax matter? a strong baseline for aspect-based sentiment analysis with roberta". In: *arXiv preprint arXiv:2104.04986* (2021). DOI: 10.48550/arXiv.2104.04986.

[5] Hai Ha Do et al. "Deep learning for aspect-based sentiment analysis: a comparative review". In: *Expert systems with applications* 118 (2019), pp. 272–299. DOI: 10.1016/j.eswa.2018.10.003.

[6] Pengcheng He et al. "Deberta: Decoding-enhanced bert with disentangled attention". In: *arXiv preprint arXiv:2006.03654* (2020). DOI: 10.48550/arXiv.2006.03654.

[7] Vikramaditya Jakkula. "Tutorial on Support Vector Machine (SVM)". In: *School of EECS, Washington State University* 37.2.5 (2006), p. 3.

[8] Yingshi Jiang and Zuodong Sun. "Intelligent Music Content Generation Model Based on Multimodal Situational Sentiment Perception". In: *Informatica* 49.5 (2025). DOI: https://doi.org/10.31449/inf.v49i5.6846.

[9] Prachi Juyal. "Classification Accuracy in Sentiment Analysis using Hybrid and Ensemble Methods". In: *2022 IEEE World Conference on Applied Intelligence and Computing (AIC)*. IEEE. 2022, pp. 583–587. DOI: 10.1109/AIC55036.2022.9840020.

[10] Yue Kang et al. "Natural language processing (NLP) in management research: A literature review". In: *Journal of Management Analytics* 7.2 (2020), pp. 139–172. DOI: 10.1080/23270012.2020.1737994.

[11] Akbar Karimi, Leonardo Rossi, and Andrea Prati. "Improving bert performance for aspect-based sentiment analysis". In: *arXiv preprint arXiv:2010.11731* (2020). DOI: 10.48550/arXiv.2010.11731.

[12] D. K. Kirange, Ratnadeep R. Deshmukh, and Deepali Kirange. "Aspect based sentiment analysis SemEval-2014 Task 4". In: *Asian Journal of Computer Science and Information Technology* 4.8 (2014), pp. 72–75. DOI: 10.15520/ajcsit.v4i8.9.

[13] Yuting Luo. "Research on User Behaviour of Network Public Opinion Using Sentiment Analysis Algorithm". In: *Informatica* 48.21 (2024). DOI: https://doi.org/10.31449/inf.v48i21.6620.

[14] Shaha T Al-Otaibi and Amal A Al-Rasheed. "A review and comparative analysis of sentiment analysis techniques". In: *Informatica* 46.6 (2022), pp. 33–44. DOI: 10.31449/inf.v46i6.3991.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "GloVe: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics. 2014, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[16] Alexander Rietzler et al. "Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification". In: *arXiv preprint arXiv:1908.11860* (2019). DOI: 10.48550/arXiv.1908.11860.

[17] Aliea Sabir, Huda Adil Ali, and Maalim A Aljabery. "ChatGPT tweets sentiment analysis using machine learning and data classification". In: *Informatica* 48.7 (2024). DOI: https://doi.org/10.31449/inf.v48i7.5535.

[18] Kevin Scaria et al. "InstructABSA: Instruction Learning for Aspect Based Sentiment Analysis". In: *arXiv preprint arXiv:2302.08624* (2023). DOI: 10.48550/arXiv.2302.08624.

[19] Emanuel Huber da Silva and Ricardo Marcondes Marcacini. "Aspect-based sentiment analysis using BERT with disentangled attention". In: *Proceedings*. 2021. DOI: 10.48550/arXiv.2106.01237.

[20] Youwei Song et al. "Attentional Encoder Network for Targeted Sentiment Classification". In: *arXiv preprint arXiv:1902.09314* (2019). DOI: 10.48550/arXiv.1902.09314.

[21] Youwei Song et al. "Attentional encoder network for targeted sentiment classification". In: *arXiv preprint arXiv:1902.09314* (2019). DOI: 10.48550/arXiv.1902.09314.

[22] Bowen Xing and Ivor W Tsang. "Understand me, if you refer to aspect knowledge: Knowledge-aware gated recurrent memory network". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 6.5 (2022), pp. 1092–1102. DOI: 10.1109/TETCI.2022.3155731.

[23] Heng Yang and Ke Li. "Improving Implicit Sentiment Learning via Local Sentiment Aggregation". In: *arXiv e-prints* (2021), arXiv–2110. DOI: 10.48550/arXiv.2110.00000.

[24] Heng Yang et al. "A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction". In: *Neurocomputing* 419 (2021), pp. 344–356. DOI: 10.1016/j.neucom.2020.08.089.

[25] Zhilin Yang et al. "Xlnet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* 32 (2019). DOI: 10.48550/arXiv.1906.08237.

[26] Yong Yu et al. "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures". In: *Neural Computation* 31.7 (2019), pp. 1235–1270. DOI: 10.1162/neco_a_01199.

[27] Yiming Zhang et al. "Towards Unifying the Label Space for Aspect-and Sentence-based Sentiment Analysis". In: *arXiv preprint arXiv:2203.07090* (2022). DOI: 10.48550/arXiv.2203.07090.

Comparative Analysis of ARDL, LSTM, and XGBoost Models For Forecasting The Moroccan Stock Market During The COVID-19 Pandemic

Oukhouya Mohamed Hassan[1], Angour Nora[2], Aboutabit Noureddine[1], Hafidi Imad[1]
[1]Laboratory LIPIM, ENSA Khouribga, University Sultan Moulay Slimane, Khouribga, Morocco
[2]Faculty of Law, Economics and Social Sciences, Salé, Mohammed V University, Rabat, Morocco
E-mail: oukhouya.mhassan@gmail.com, n.angour@um5r.ac.ma, n.aboutabit@usms.ma, i.hafidi@usms.ma

Keywords: ARDL, COVID19, MASI, LSTM, XGBOOST, forecasting

Received: February 22, 2024

This study evaluates and compares the forecasting performances of the ARDL (AutoRegressive Distributed Lag), LSTM (Long Short-Term Memory), and XGBOOST (Extreme Gradient Boosting) models on the MASI (Moroccan All Shares Index). The analysis incorporates daily new COVID-19 cases into the ARDL approach to investigate short-term and long-term relationships with MASI. Cointegration and causality tests are conducted on daily time series data. In terms of accuracy, the ARDL model, especially when including trend and seasonality variables, outperforms LSTM and XGBOOST models. The ARDL model with lags, trend, and seasonality variables achieves the lowest Mean Absolute Percentage Error (MAPE) of 26.7%, with a processing time of 1 second. In comparison, the LSTM and XGBOOST models have MAPE values of 30.5% and 32%, respectively, while requiring significantly longer processing times. These findings suggest that the ARDL model is more efficient and accurate in predicting future values of MASI under pandemic conditions.

Povzetek: Članek primerja modele ARDL, LSTM in XGBoost za napovedovanje maroškega indeksa MASI med COVID-19. Model ARDL presega modele globokega učenja glede natančnosti in učinkovitosti, kar poudarja odpornost ekonometričnih pristopov pri finančnih napovedih.

1 Introduction

Investing in financial markets has long been a focal point for capital holders. While ongoing research and development of new strategies persist, the ever-evolving nature of financial markets necessitates constant adaptation by traders. Consequently, they are increasingly turning to machine learning as a means to enhance and optimize their trading systems. This approach allows them to harness the power of advanced algorithms and data analysis techniques to stay competitive and make more informed investment decisions in the dynamic world of finance.

As financial markets keep changing with more traders and a lot of money involved, it becomes extremely hard to perfectly predict what prices will do in the future. These markets are very complicated, and many things can affect prices, making it almost impossible to guess accurately. That's why traders use complex models and strategies to handle this uncertainty [15]. The various factors that affect financial time series create a situation where they don't stay the same over time, making them non-stationary. This non-stationarity adds complexity to the already challenging tasks of predicting future outcomes and making smart investments in financial markets. Researchers are keenly interested in finding ways to transform these non-stationary time series into more predictable and stable ones, as this can greatly enhance the effectiveness of trading systems [11].

Successful financial forecasting involves combining financial theory, market analysis, diverse data sources, and computational advancements. To build precise models, a comprehensive toolkit is essential, with financial econometrics and machine learning being key components. Machine learning enables better predictions by handling nonlinear data, complex variable interactions, and unstructured datasets. However, financial econometrics remains crucial for inferential analysis of economic relationships in finance, and its importance persists alongside machine learning [5]

Financial markets are continuously impacted by events around the world (war, pandemic, natural disaster, etc.).The economic impact of COVID-19 via government actions was examined and it was proved that stocks are negatively affected by social distancing, but positive outcomes are observed for awareness, testing, and income support measures, thereby highlighting the dual economic effects of government responses [2]. As an example of war impact, the Russian-Ukrainian conflict has influenced the interconnections between Russia, European financial markets, and global commodity markets [17] .

Time series forecasting is a critical tool in predicting future outcomes in various fields. In the finance industry, forecasting financial market trends is essential for investors, policymakers, and analysts to make informed decisions. The recent COVID-19 pandemic has caused significant disruptions to global financial markets, including the

Moroccan financial market. The impact of the pandemic on the financial market has made forecasting more challenging, and traditional methods may not be enough to capture the complexity of the situation.

Machine learning algorithms have shown promise in providing accurate predictions for various forecasting problems, including time series forecasting. In this article, we aim to explore the effectiveness of machine learning algorithms in forecasting the Moroccan financial market's trends, with a particular focus on the impact of the COVID-19 pandemic.

We will explore the hypothesis that machine learning algorithms, specifically ARDL, LSTM, and XGBoost, can provide more accurate and reliable predictions for the Moroccan financial market than traditional methods. The article aims to contribute to the existing body of knowledge by providing insights into the effectiveness of machine learning algorithms in financial time series forecasting and their ability to capture the effects of unpredictable events such as the COVID-19 pandemic.

2 Related work

Forecasting exchange rates is a critical task in the financial industry, and it has attracted significant attention from researchers in recent years. The high volatility and complexity of the foreign exchange market make it challenging to predict exchange rates accurately. Deep learning techniques, particularly LSTM and XGBoost, as well as the ARDL model, have been widely used to tackle this problem. In this section, we will discuss some works that studied the predictability of Forex based on LSTM, XGBoost, and ARDL.

The article [6] examines the impact of the COVID-19 pandemic on the Saudi Arabian stock market. The authors use an ARDL model to analyze the relationship between COVID-19 cases and the stock market index, taking into account other factors such as oil prices and exchange rates. They use data from January 2015 to July 2020, which includes the period of the pandemic. The results show that COVID-19 has had a significant negative impact on the Saudi stock market, with a decrease in the stock market index following an increase in the number of COVID-19 cases. The authors also find that oil prices and exchange rates have a significant impact on the stock market, but that the effect of COVID-19 is larger. The article provides evidence of the impact of the COVID-19 pandemic on the Saudi Arabian stock market and highlights the importance of considering multiple factors when analyzing the relationship between the stock market and external events. The findings may be useful for investors and policymakers in the region.

This study[14] examines the impact of COVID-19 on daily market returns in affected developed and emerging markets. It finds that an increase in new cases and deaths negatively affects market returns globally. Interestingly, daily testing has a positive impact. These findings apply to both developed and emerging markets, with the exception that news of new COVID-19 deaths positively affects emerging markets. The study suggests that early proactive measures by governments can protect financial markets and boost investor confidence during future pandemics.

This study [10] investigates the correlation between stock market progress, economic growth, and financial innovation in Bangladesh from 1980 to 2016. To examine the cointegration in the long term, the ARDL bounds test was utilized. Additionally, the Granger Causality test was implemented to identify directional causality between the variables through the error correction mechanism. The ARDL bounds test approach confirms the existence of a long-term relationship between economic growth, stock market progress, and financial innovation. Moreover, the results of the Granger Causality test show a mutual relationship between financial innovation, economic growth, and stock market progress, both in the long and short run. These findings affirm the hypothesis that the growth of market-based financial systems and financial innovation can drive economic growth.

The authors in [13] analyze the connection between oil prices and stock prices in both oil-exporting and oil-importing countries. To do so, they approach the relationship from several angles. Firstly, they examine the possibility of non-linearities in the relationship to determine the unequal response of stock prices in the two categories to positive and negative shifts in oil prices. Secondly, they account for within-group differences by allowing for heterogeneity in the cross sections. Thirdly, they compare the predictability of linear (symmetric) and nonlinear (asymmetric) Panel ARDL models through the Campbell and Thompson (2008) test. The results reveal that the stock prices in both oil-exporting and oil-importing countries respond differently to changes in oil prices, with a stronger response seen in oil-importing countries compared to oil-exporting countries.

The article [12] introduces a new model for forecasting nonlinear time series data. The proposed model combines two existing techniques: EMD (empirical mode decomposition) and NARDL (neural autoregressive distributed lag) modeling. EMD is used to decompose the time series into a series of IMFs (intrinsic mode functions), each representing a different scale or frequency component of the data. The NARDL model is then applied to each IMF to capture the nonlinear relationships between the variables.The MNARDL (multiscaled NARDL) model is shown to outperform other existing models in terms of accuracy, particularly in cases where the data exhibits nonlinearities and nonstationarities. The model is demonstrated through simulation studies and real-world applications in the areas of economics and finance.The article presents a novel approach to time series forecasting that integrates two existing techniques and provides improved accuracy for modeling nonlinear data.

This paper [19] explores the challenges of predicting stock market movements, a key area of interest across var-

ious fields such as statistics, AI, and finance. It highlights the importance of accurate predictions in reducing investment risks and emphasizes that machine learning models often outperform traditional statistical approaches. Specifically, the study investigates the Gaussian Naïve Bayes (GNB) algorithm, which has not been extensively studied in this context. The researchers evaluate GNB's performance when integrated with different feature scaling and extraction techniques, using Kendall's test of concordance for ranking. Results indicate that the GNB model combined with Linear Discriminant Analysis (GNB_LDA) outperformed other models in three of four evaluation metrics (accuracy, F1-score, and AUC). Additionally, the GNB model using Min-Max scaling and PCA achieved the highest specificity rank, demonstrating that GNB performs better with Min-Max scaling than with standardization techniques.

Authors of the article [8] discusses the challenges of predicting foreign exchange rates due to their complex and volatile nature. The paper proposes a new model that combines two powerful neural networks, the GRU (Gated Recurrent Unit) and LSTM, to predict the future closing prices of four major currency pairs. The proposed hybrid model outperforms standalone LSTM and GRU models, as well as a simple moving average-based statistical model, for a 10-minute timeframe and provides the best result for two currency pairs in terms of MSE (Mean Square Error), RMSE (Root Mean Square Error), and MAE (Mean Absolute Error) performance metrics for a 30-minute timeframe. The model's performance is validated using MSE, RMSE, MAE, and R score, with the proposed hybrid GRU-LSTM model proving to be the least risky among all compared models in terms of R2 score.

Forecasting fast and high-frequency financial data is challenging due to the dynamic and chaotic nature of stock markets. This study [4] presents a novel hybrid model combining fractional order derivatives and deep learning, specifically LSTM networks, to predict sudden market fluctuations. Traditional methods like data mining and statistical approaches struggle with stock price variability, but the proposed ARFIMA-LSTM (Autoregressive fractionally integrated moving average LSTM) hybrid model effectively extracts features and models non-linear functions. It overcomes volatility and overfitting issues, outperforming traditional methods by achieving approximately 80% accuracy improvement in RMSE when evaluated with real stock market data from the PSX company.

The article [1] investigates the relationship between public debt and economic growth in Morocco. The authors use an ARDL model to analyze the impact of public debt on economic growth, taking into account other factors such as investment, exports, and inflation. They use data from 1980 to 2019 to estimate the model.The results show that there is a negative relationship between public debt and economic growth in Morocco, indicating that an increase in public debt can lead to a decrease in economic growth. The authors also find that investment and exports have a positive impact on economic growth, while inflation has a negative impact. The article highlights the importance of managing public debt in order to promote economic growth in Morocco. The findings may be useful for policymakers in the country as they make decisions about fiscal policy and debt management.

Behavioral finance studies suggest that emotions impact stock markets. This paper [3] discusses a method to collect and analyze sentiment from various sources about Casablanca Stock Exchange. Using sentiment analysis and machine learning, it aims to link public sentiment to stock market performance.

The paper [18] presents a financial risk prediction model utilizing graph networks to address inaccuracies in enterprise financial risk and profit predictions. It integrates multi-scale feature extraction and sequence analysis, employing a bidirectional gated recurrent unit to effectively capture temporal relationships in time series data. The profit prediction model combines multi-scale advantages with attention mechanisms to enhance the identification of influential features, significantly improving predictive accuracy. After iterative training, the model achieved an accuracy of 98.03% and an F1 score of 0.98 for financial risk predictions. The profit prediction model outperformed others in regression and classification metrics, with a mean square error of just 0.0232. Overall, both models demonstrate strong predictive capabilities and practical significance.

This article [16] discusses the increasing importance of electric vehicle load scheduling in grid power scheduling due to the rising use of electric vehicles. The effective electric vehicle power dispatching helps balance the peak-valley difference of power dispatching, increase the power supply utilization rate, and reduce the power supply pressure of line transformer. The article summarizes the research status of electric vehicle charging load, analyzes traditional charging load research methods and proposes a charging load forecasting method combining XGBoost and LSTM. The proposed method is based on the prediction results of the XGBoost model for feature engineering and statistical methods, and training the LSTM model for load prediction. The charging station load forecasting method studied in this paper can support the regional load forecasting research of electric vehicles with high permeability and further optimize power dispatching. The proposed method is verified using the data of a charging station in Jiangsu.

This paper[7] introduces a new approach to predict future returns in volatile and nonlinear financial markets. It consists of three stages: fractal modeling and recurrence analysis, Granger causality tests, and wavelet transformation. Machine learning algorithms are applied to learn patterns and make predictions. Testing with Asian emerging stock indexes from 2012 to 2017 shows that this framework is effective for forecasting.

The authors of [9] discuss the challenge of forecasting stock prices due to the volatile nature of the stock market, which makes it difficult to apply linear models or simple

time-series or regression techniques. The author suggests that SVM (support vector machine) is a good alternative tool for stock forecasting, as it is a popular machine learning technique for the capital investment industry that can forecast financial data more accurately. The article presents an experiment that examines the stock prices of 5 Moroccan banks and shows that SVM can perform better when the global evolution of the market is added to the independent variables. To express the global evolution of the market, the author uses three indices of the Casablanca Stock Exchange: MASI, MADEX (Moroccan Most Active Shares Index), and Banks Sector Index.Also, this article highlights the potential of SVM for stock price prediction, and emphasizes the importance of considering the global market conditions as a variable to improve forecasting accuracy. The findings may be useful for investors and financial analysts looking for new methods to improve their stock trading and investment decisions. Below is a comparative table (Table 1) of the results of the related work, with their results.

In conclusion, these studies provide evidence that LSTM, XGBoost, and ARDL models can all be effectively used to forecast exchange rates. The results suggest that LSTM is generally better in terms of accuracy and stability, while XGBoost is faster in training and prediction time. In this current work, we aim to demonstrate that ARDL models have distinct advantages, particularly in the context of the Moroccan market, as they take into account exogenous variables during events and can be explained with a formula, contrasting with the black-box nature of machine learning algorithms. The unique behavior of the Moroccan market necessitates a deeper exploration of these tailored approaches.

3 Methodology

The article aims to explore the possible short-term and long-term relationships between the COVID-19 pandemic and the Moroccan financial market. The study also investigates the potential of the ARDL model in improving the accuracy of future value predictions for the Moroccan financial market, compared to LSTM an XGBOOST.

3.1 COVID19 impact on Moroccan financial market based on ARDL

Most financial market studies use VAR (Vector autoregression) modeling to analyze the relationship between explanatory and explained variables. However, this method requires that the series be integrated of the same order, which is not always the case for macroeconomic series. To address this issue, Pesaran, Shin, and Smith (2001) proposed the ARDL method, which considers the limitations of the VAR model. This approach, which tests the long-run relationship based on variables with different integration orders, is an alternative to cointegration tests. The ARDL method provides unbiased estimates of the long-run relationship and is more suitable for small samples. In this study, we will use the ARDL model to investigate the short- and long-run relationships between COVID-19 pandemic and Moroccan financial market. We will also determine the optimal number of lags using AIC (Akaike Information Criterion) and test for the presence of causal relationships between the variables.

3.1.1 Research model

The application of the ARDL model, represented by the subsequent equation, will enable the estimation of the responses to the hypotheses stated underneath :

$$\begin{aligned}\Delta LogMASI_{(t)} = C + \sum_{i=1}^{p} \alpha_{1i} \Delta LogMASI_{(t-i)} + \\ \sum_{i=0}^{q} \alpha_{2i} \Delta LogCOVID_{(t-i)} \\ + \beta_1 LogMASI_{(t-1)} \\ + \beta_2 LogCOVID_{(t-1)} + \varepsilon_{(t)}\end{aligned} \quad (1)$$

(Table 2) provides a description of the various variables included in the equation.

Table 2: Description of equation variables

Variable	Description
MASI	Price of the Moroccan All Shares Index
COVID	Daily new cases of COVID-19
C	Constant
Log( )	Natural logarithm operator
Δ	First difference operator
$\alpha_1; \alpha_2$	Short-run coefficients
$\beta_1; \beta_2$	Long-run dynamics
$\varepsilon(t)$	Error term

It is important to note that before estimating the ARDL model, a cointegration test must be performed. This is because it is not possible to estimate an error correction model or determine the short and long-term effects for variables that are not cointegrated. In the case of long-term effects, we conduct a cointegration limit test based on the Fisher statistic, with the hypothesis being that the variables are cointegrated.

$H_0 : \beta_1 = \beta_2 = 0$

If H_0 is rejected, it indicates the existence of cointegration. The F-statistic should surpass the upper bounds of I(1) to reject H_0, but if it is less than the lower critical bounds of I(0), H_0 is accepted. Otherwise, no conclusion can be made. Once cointegration is confirmed, the long-term relationship is determined by eliminating the variables in first difference (Morley, 2006 and Antoniou et al., 2013). Utilizing equation (1), we can deduce that the relationship is illustrated by the subsequent equation :

Table 1: Summary of studies and their findings

Work	Models	Dataset	Results
[6]	ARDL	Saudi Arabia (TASI)	The study finds a long-term negative relationship between LOG_TASI and LOGCOVID_19, with unidirectional causality from COVID-19 to TASI, highlighting the need for strong national measures to prevent a significant stock market crash in KSA.
[14]	panel-EGLS and panel quantile regression approaches	Different emerging markets	Finding that new cases and deaths negatively influence returns, while increased testing positively affects them, with some variations between developed and emerging markets.
[10]	ARDL	Bangladesh market	The study revealed that financial innovation positively influences economic growth both in the short and long run, which is statistically significant as well.
[13]	Nonlinear Panel ARDL	Oil-stock nexus	The study finds that incorporating positive and negative oil price changes improves stock price forecasts only for oil-importing countries, highlighting a key difference in the oil price-stock relationship between importing and exporting nations.
[12]	Multiscaled Neural ARDL	Oil & Bitcoin	The empirical experiments conducted on real-world economic data prove that the decomposing framework significantly improves the forecasting accuracy.
[8]	GRU-LSTM hybrid network	EUR/USD, GBP/USD, USD/CAD	The hybrid GRU-LSTM model outperforms standalone LSTM, GRU, and SMA models in predicting GBP/USD and USD/CAD currency pairs, demonstrating the best performance metrics and lowest risk overall.
[1]	ARDL	GDP	The study using an Auto Regressive Distributed Lag model reveals that total government debt significantly negatively impacts economic growth in Morocco both in the short term (0.62% decrease in growth for a 1% increase in debt) and long term (0.4% decrease), while the investment rate positively influences growth; however, bank credit to the private sector remains statistically insignificant, highlighting challenges in the trade balance and the need for broader economic reforms.
[9]	SVM	MASI & MADEX	SVM significantly improves stock forecasting accuracy for five Moroccan banks, especially when incorporating global market trends like the MASI, MADEX, and Banks Sector Index.

$$LogMASI_{(t)} = -\left(\frac{C}{\beta_1}\right) - \left(\frac{\beta_2}{\beta_1}\right) LogCOVID_{(t)} \quad (2)$$

The confirmation of the presence or absence of cointegration between the variables can be done by using ECM (error correction model) for eq.2 as demonstrated below:

$$\begin{aligned} \Delta LogMASI_{(t)} = &\sum_{i=1}^{p} \alpha_{1i} \Delta LogMASI_{(t-i)} \\ &+ \sum_{i=0}^{q} \alpha_{2i} \Delta LogCOVID_{(t-i)} \\ &+ \beta_1 ECM_{(t-1)} + \varepsilon_t \end{aligned} \quad (3)$$

$$\begin{aligned} ECM_{(t)} = LogMASI_{(t)} - \\ \left[-\left(\frac{C}{\beta_1}\right) - \left(\frac{\beta_2}{\beta_1}\right) LogCOVID_{(t)}\right] \end{aligned} \quad (4)$$

In the ARDL model, trend and seasonality are incorporated through lagged values of MASI index prices and COVID-19 cases, selected to capture short-term fluctuations and long-term equilibrium adjustments. The trend component is addressed by using previous levels of the MASI, reflecting persistent market shifts, while seasonality is captured by lags that align with observed recurring patterns in COVID-19 data. This structure ensures the model can respond to both immediate impacts and ongoing trends in the market, particularly during volatile periods like the COVID-19 pandemic.

3.2 Forecasting MASI index with ARDL, LSTM and XGBoost

We hypothesize that by using ARDL, LSTM and XGBoost models, we can accurately forecast the future values of the MASI. Specifically, we expect that:

- The ARDL model will perform well in predicting MASI since it can capture both short and long-run dynamics of the data. We expect that the model will be able to identify significant relationships between MASI and other relevant economic indicators, such as COVID19 cases, inflation rate, and interest rate.

- The LSTM model will also perform well in predicting MASI since it can capture complex temporal dependencies in time series data. We expect that the model will be able to learn and identify the patterns in MASI and its determinants over time, thereby improving its forecasting accuracy.

- The XGBoost model will perform well in predicting MASI since it is an ensemble tree-based method that can capture non-linear relationships and interactions between variables. We expect that the model will be able to identify the most important features that influence MASI, and thereby provide more accurate forecasts than traditional regression-based models.

4 Results and discussion

4.1 Data and description

Our study focuses on analyzing the impact of the COVID19 pandemic on the MASI, using daily closing prices that were obtained from www.investing.com. We also collected data on the daily number of confirmed COVID19 cases from the official website of the Moroccan Ministry of Health. The data covers the period from March 03, 2020 (the day when the first COVID19 case was reported in Morocco) to February 11, 2022. Prior to analysis, both variables were subjected to a log transformation. (the data is accessible via this link https://osf.io/umjgb/files).

4.2 Descriptive statistics

The following table (Table 3) displays the descriptive statistics for two variables analyzed in our study. The first variable, MASI, represents the closing price of MASI, while the second variable, NEW_CASES, represents the number of new daily confirmed cases of COVID-19 in Morocco.

Table 3: Descriptive statistics of MASI and COVID-19 new cases (2 March 2020 to 11 February 2022)

	MASI	NEW_CASES
Mean	11524.53	1576.031
Median	11517.92	579.0000
Maximum	13991.47	12039.00
Minimum	8987.890	0.000000
Std. Dev.	1347.250	2219.878
Skewness	0.018541	2.233916
Kurtosis	1.856586	8.076603
Jarque-Bera	26.44808	924.1962
Observations	485	485

The two figures (1,2) show the trends of MASI price and COVID19 new cases over the same period.

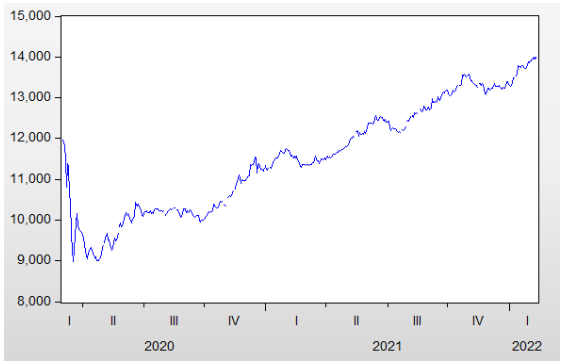


Figure 1: Movement of MASI price in the concerned period

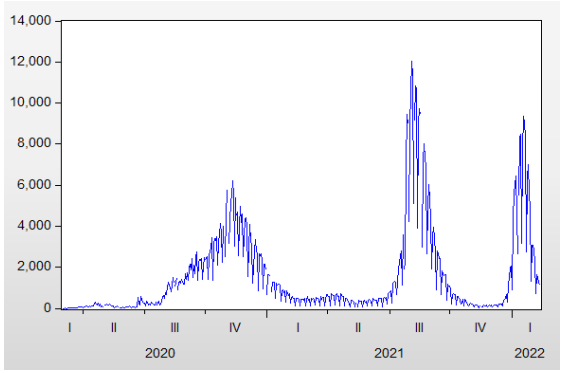


Figure 2: COVID19 new cases in the concerned period

4.3 ARDL quantitative results and discussions

4.3.1 Stationarity (Unit root tests)

This part discusses the concept of non-stationarity in time series data, where a series with a moving average and/or variance that varies over time is considered non-stationary. If not addressed through stationarization, this non-stationarity can lead to ”spurious” regressions. To determine if a series is stationary or not (i.e., if a unit root exists), several tests can be used, such as ADF (augmented Dickey-Fuller) test, PP (Phillippe-Perron) test, AZ (Andrews and Zivot) test, the Ng test-Perron, and KPSS (Kwiatkowski-Phillips-Schmidt-Shin) test. Among these tests, the ADF and PP tests are the most commonly used and easy to apply. The ADF test is effective in cases of autocorrelation of errors, while the PP test is suitable in the presence of heteroscedasticity. In this study, the ADF and PP tests were used with the test critical values of Mackinnon (1996), and the results are presented in (Table 4).

The results of the study show that the Log MASI Level data is non-stationary, while the Log MASI 1st difference and Log COVID19 data are stationary. The optimal number of lags is 17 based on AIC, and all variables have a significance at the 1% level. The ADF test statistic values are also reported for each variable, with the Log MASI Level having a test statistic value greater than the critical value and a p-value greater than 0.05, indicating non-stationarity. Meanwhile, the Log MASI 1st difference and Log COVID19 have test statistic values lower than their respective critical values and p-values lower than 0.05, indicating stationarity (Table 5).

4.3.2 ARDL estimation

To select the most optimal ARDL model with statistically significant results and fewer parameters, we utilize AIC. The model is estimated with the ”constant & trend” option because of its high significance (Probability $< 1\%$). The following are the estimation results of the selected optimal ARDL model.

The optimal model is selected based on the AIC value, where the model with the smallest AIC value (Figure 3) is considered the best. In this study, the optimal model is the ARDL(2,6), which is statistically significant with a global

Table 4: ADF Unit Root test onthe log level of variables

Variables	Level					Integration order
	T-statistic	1%	5%	10%	P-value	
Log MASI	-0.8291	-3.4441	-2.8675	-2.5700	0.8095	I(1)
Log MASI 1st Difference	-5.3066	-3.4441	-2.8675	-2.5700	0.0000	
Log COVID	-3.0442	-3.4442	-2.8675	-2.5700	0.0317	I(0)

Table 5: PP Unit Root test on the log level of variables

Variables	Level					Integration order
	T-statistic	1%	5%	10%	P-value	
Log MASI	-0.3695	-3.4436	-2.8672	-2.5698	0.9114	I(1)
Log MASI 1st Difference	-18.1950	-3.4436	-2.8673	-2.5699	0.0000	
Log COVID	-5.6379	-3.4437	-2.8673	-2.5699	0.0000	I(0)

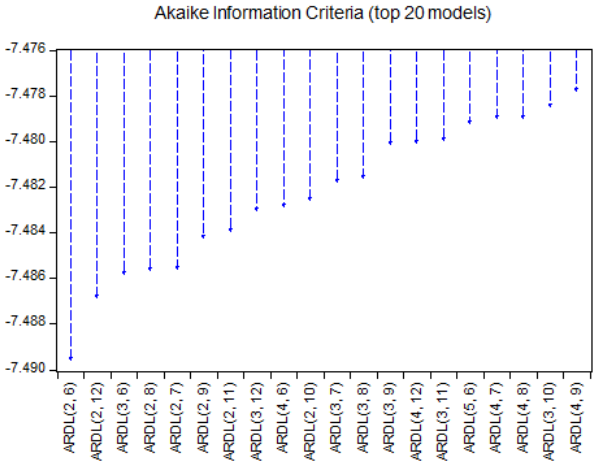


Figure 3: Akaike information criteria

Prob (F-statistic) value of 0.0000 (Table 6).

While most of the coefficients in the model are significant, it is necessary to conduct validity tests such as autocorrelation tests to ensure its validity (Table 7). Additionally, the model is globally significant.

Based on the Ljung-Box test results, the Q-statistic probability is above the 5% and 10% thresholds for all results, indicating the absence of autocorrelation in the model errors. This is important because the presence of autocorrelation in residuals can lead to inconsistent parameter estimates due to the inclusion of a lagged dependent variable as an exogenous variable in the model.

The F test value of 9.43 (Table 8) exceeds the majority of the I(1) bounds, indicating significant **cointegration** between the variables at a 2.5% significance level. This suggests that it is possible to estimate the long-term effects of Log_COVID19 on Log_MASI.

The long term relationship is described as follows : Log_MASI = -0.0020*Log_COVID. The results (Table 9) show that there is a significant negative long-term relationship between COVID 19 and the stock market in Morocco: a 100% increase in the daily number of confirmed cases of COVID-19 resulted in a 0.2% decrease in the MASI price. In the short-term relationship, it appears that there is no significance between all variables, but Log_COVID delayed by 5 days (t-5) has a positive impact on Log_MASI in day (t) at the 10% level.

Since correlation does not necessarily imply causality, we must test the causality that may exist between the variables, we use the Toda Yamamoto causality test.

A causal relationship from Log_COVID to Log_MASI is confirmed by the Toda-Yamamoto causality test (Prob = 0.0005, the null hypothesis is rejected) (Table 10). However, there is no causality between Log_MASI and Log_COVID (Pob=0.0857).

4.4 Forecasting results

4.4.1 Performance measure

The MAPE (Mean Absolute Percentage Error) is a practical metric used to evaluate the accuracy of forecasting models. It calculates the average percentage deviation of forecasted values from the observed values. By expressing the error as a percentage, the MAPE enables easy comparison between different models. The formula for MAPE is as follows:

$$MAPE = 100\% - \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{Forecasted\_value_t}{Real\_value_t}\right| \quad (5)$$

With **n** the number of forceasted values.

4.4.2 Results

(Table 11) outlines the various approaches utilized to forecast the MASI index (ARDL, LSTM and XGBOOST), along with the corresponding inputs for each method.

A greedy algorithm was applied to determine the optimal lag values for the ARDL model, focusing on selecting the lagged variables that best capture the short-term dynamics of the MASI index. Specifically, the model uses a unique set of inputs based on two-day lags for MASI prices and six-day lags for new confirmed COVID-19 cases. This selection reflects an iterative process in which different lag

Table 6: ARDL estimation

Variable	Coefficient	Std. Error	t-Statistic	Prob.*
MASI_LOG(-1)	1.196800	0.043909	27.25662	0.0000
MASI_LOG(-2)	−0.257500	0.043916	−5.863430	0.0000
COVID19_LOG	0.001027	0.000702	1.461975	0.1444
COVID19_LOG(-1)	−0.000982	0.000711	−1.382608	0.1675
COVID19_LOG(-2)	0.000570	0.000559	1.020708	0.3079
COVID19_LOG(-3)	−0.001067	0.000566	−1.885020	0.0601
COVID19_LOG(-4)	−0.000443	0.000557	−0.795811	0.4265
COVID19_LOG(-5)	−0.000417	0.000710	−0.586752	0.5577
COVID19_LOG(-6)	0.001191	0.000681	1.749257	0.0809
C	0.555984	0.128045	4.342094	0.0000
@TREND	$5.14E-05$	$1.22E-05$	4.212603	0.0000

R-squared	0.997024	Mean dependent var	9.346391
Adjusted R-squared	0.996960	S.D. dependent var	0.118471
S.E. of regression	0.006532	Akaike info criterion	−7.201220
Sum squared resid	0.019756	Schwarz criterion	−7.104652
Log likelihood	1717.689	Hannan-Quinn criter.	−7.163241
F-statistic	15512.26	Durbin-Watson stat	2.022289
Prob(F-statistic)	0.000000		

*Note: p-values and any subsequent tests do not account for model selection.

Table 7: Autocorrelation of residuals.

Autocorrelation	Partial Correlation		AC	PAC	Q-Stat	Prob*
.|. |	.|. |	1	−0.031	−0.031	0.4672	0.494
.|. |	.|. |	2	0.030	0.029	0.8915	0.640
.|. |	.|. |	3	−0.059	−0.057	2.5576	0.465
.|. |	.|. |	4	−0.055	−0.060	4.0298	0.402
.|* |	.|* |	5	0.082	0.082	7.2305	0.204
.|. |	.|. |	6	0.020	0.026	7.4317	0.283
.|* |	.|* |	7	0.099	0.090	12.155	0.096
.|. |	.|. |	8	−0.029	−0.019	12.565	0.128
.|. |	.|. |	9	−0.019	−0.015	12.741	0.175
.|. |	.|. |	10	−0.038	−0.032	13.453	0.199

*Probabilities may not be valid for this equation specification.

Table 8: Bound test to cointegation results

F-Bounds Test			Null Hypothesis: No levels relationship	
Test Statistic	Value	Signif.	Lower bound I(0)	Upper bound I(1)
			Asymptotic: n=1000	
F-statistic	9.433081	10%	5.59	6.26
k	1	5%	6.56	7.3
		2.5%	7.46	8.27
		1%	8.74	9.63

Table 9: Dynamics of the short-run and long-run

Conditional Error Correction Regression				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
MASI_LOG(-1)*	−0.060700	0.013991	−4.338372	0.0000
COVID19_LOG(-1)	−0.000121	0.000223	−0.545517	0.5857
D(MASI_LOG(-1))	0.257500	0.043916	5.863430	0.0000
D(COVID19_LOG)	0.001027	0.000702	1.461975	0.1444
D(COVID19_LOG(-1))	0.000166	0.000680	0.243598	0.8077
D(COVID19_LOG(-2))	0.000736	0.000706	1.042008	0.2980
D(COVID19_LOG(-3))	−0.000331	0.000706	−0.469430	0.6390
D(COVID19_LOG(-4))	−0.000774	0.000677	−1.144776	0.2529
D(COVID19_LOG(-5))	−0.001191	0.000681	−1.749257	0.0809
EC = Log_MASI - (-0.0020*COVID19_LOG)				
Long-Run Coefficients				
Variable	Coefficient	Std. Error	t-Statistic	Prob.
COVID19_LOG	−0.002001	0.003616	−0.553284	0.5803
C	0.555984	0.128045	4.342094	0.0000
@TREND	$5.14E-05$	$1.22E-05$	4.212603	0.0000

Table 10: Toda-Yamamoto causality test.

Dependent variable: MASI_LOG			
Excluded	Chi-sq	df	Prob.
COVID19_LOG	15.32437	2	0.0005
All	15.32437	2	0.0005
Dependent variable: COVID19_LOG			
Excluded	Chi-sq	df	Prob.
MASI_LOG	4.914155	2	0.0857
All	4.914155	2	0.0857

Table 11: Forecasting results

Method	**Inputs**	**MAPE**	**Processing time**	
			Training	**Forecasting**
ARDL	Lags only	34.4%	1s	
ARDL	Lags + trend	32.3%	1s	
ARDL	Lags + seasonality	32.1%	1s	
ARDL	Lags + trend + seasonality	26.7%	1s	
LSTM	Lags	30.5%	2min 56s	21s
XGBOOST	Lags	32%	6min	1s

combinations were tested, with the greedy algorithm identifying the combination that minimized the forecasting error. By emphasizing shorter lags, this approach aims to model the immediate impacts of recent MASI price fluctuations and COVID-19 case data, as these factors are expected to influence market movements in the short term. This method not only simplifies the model but also enhances its responsiveness to recent changes, thereby improving predictive accuracy.

In contrast, LSTM and XGBOOST models rely on a grid search of hyperparameters to select the best combination of variables for the forecast. This method involves testing different combinations of input variables to determine the most optimal set for predicting the MASI index accurately.

After evaluating the performance of all three methods, the ARDL model with lags, trend, and seasonality variables

outperformed both LSTM and XGBOOST models in terms of accuracy and processing time. This finding indicates that the inclusion of trend and seasonality variables in addition to the lags of MASI prices and new confirmed cases data significantly improves the model's accuracy.

The ARDL model's execution time of only 1 second is also impressive and demonstrates its efficiency, particularly when compared to the relatively more computationally expensive LSTM and XGBOOST models. Overall, the ARDL model with its unique set of inputs, outperforms the LSTM and XGBOOST models in terms of accuracy, efficiency, and computational resources.

4.4.3 Forecasting discussion

In this section, we compare the forecasting results of the ARDL, LSTM, and XGBoost models with findings from existing literature, shedding light on why the ARDL model outperformed the others and analyzing the conditions that contributed to its superior performance.

The ARDL model demonstrated a remarkable accuracy, achieving a MAPE of 26.7% when incorporating lags, trend, and seasonality variables. This performance is particularly noteworthy when juxtaposed with the LSTM and XGBoost models, which recorded MAPEs of 30.5% and 32%, respectively. A primary reason for the ARDL model's success lies in its ability to effectively handle short-term dynamics using a specific set of inputs—namely, the lags of MASI prices and new confirmed COVID-19 cases. These variables are likely to exert an immediate impact on the market, allowing ARDL to capture the temporal relationships more adeptly.

The findings align with existing literature that emphasizes the efficacy of ARDL in time series forecasting, particularly in scenarios characterized by smaller sample sizes. The simplicity of the ARDL model, combined with its explicit incorporation of exogenous variables, allows for easier interpretation and better forecasting accuracy under conditions where non-linear relationships might not be as pronounced. In contrast, both LSTM and XGBoost, while powerful in handling complex, non-linear patterns, require larger datasets to truly exploit their capabilities. In our case, the relatively limited data available during the COVID-19 period may have hindered these models from achieving optimal performance.

Additionally, the computational efficiency of the ARDL model—requiring only 1 second for execution—highlights its practicality for real-time forecasting, particularly in fast-moving markets like the Moroccan stock market. In contrast, LSTM's training time of 2 minutes and 56 seconds, along with a forecasting time of 21 seconds, and XGBoost's 6 minutes for training, demonstrate the trade-off between model complexity and computational demand.

This study's results suggest that while machine learning models like LSTM and XGBoost offer sophisticated techniques for capturing non-linear patterns, the unique conditions of the Moroccan market, combined with the characteristics of the dataset, made ARDL the more suitable choice for this specific forecasting task. Future work could explore hybrid models that combine the strengths of ARDL with machine learning techniques, potentially leading to enhanced accuracy. Additionally, further research should investigate the scalability of these findings to other emerging markets with similar characteristics, as well as the implications of larger datasets that might better inform machine learning approaches.

4.4.4 Computational efficiency discussion

The ARDL model is the most computationally efficient in both memory usage and scalability due to its simple linear regression structure. It requires minimal memory since it only stores a limited number of lagged values and coefficients, making it ideal for applications needing quick and efficient forecasts on moderate datasets. However, its simplicity may limit its performance with larger or more complex datasets.

In contrast, the LSTM model, while powerful in capturing complex, long-term dependencies in sequential data, is much more memory-intensive due to its multi-layered architecture and need to store information across each time step. This memory requirement restricts its scalability unless specialized hardware like GPUs is used, making LSTM better suited for smaller datasets or scenarios where memory resources are abundant.

XGBoost balances efficiency and scalability well. It requires more memory than ARDL due to its ensemble of decision trees, but it is significantly more scalable because of its parallel processing capabilities. Optimized for handling large datasets and sparse data, XGBoost is ideal for applications prioritizing accuracy on large datasets, although it requires moderate memory availability for efficient processing. Overall, ARDL offers the most efficient option for smaller datasets, while XGBoost and LSTM trade off memory and computational resources for accuracy in large and complex data scenarios.

4.5 Summary and conclusions

This study focuses on modeling the impact of the Covid-19 pandemic on the Moroccan stock market using the ARDL estimation approach. The study analyzes both short-term and long-term relationships between the MASI index and the number of daily confirmed Covid-19 cases, indicating a negative long-term relationship and unidirectional causality from Covid-19 to the MASI index.

To capture the short-term dynamics of the MASI index, the ARDL model uses a unique set of inputs based on lags of MASI prices for two previous days and new confirmed cases data from the previous six days. In contrast, LSTM and XGBOOST models use a grid search of hyperparameters to select the optimal set of input variables for accurate forecasting.

After evaluating the performance of all three models,

the ARDL model with lags, trend, and seasonality variables outperforms both LSTM and XGBOOST models in terms of accuracy and processing time. The inclusion of trend and seasonality variables significantly improves the model's accuracy, and the ARDL model's execution time of only 1 second demonstrates its efficiency compared to the relatively more computationally expensive LSTM and XGBOOST models. In summary, the ARDL model with its unique set of inputs proves to be the best option for accurately forecasting the MASI index during the Covid-19 pandemic.

The ARDL model assumes a stable, long-term relationship between variables, requiring that the data series be either stationary or integrated of the same order (usually I(0) or I(1)). To meet this, unit root tests are performed before model estimation. Additionally, ARDL presumes no perfect multicollinearity among explanatory variables, ensuring independent impacts, and it assumes that residuals are normally distributed and homoscedastic—meaning constant variance of error terms over time. If these assumptions are not met, the coefficient estimates may be biased or inefficient, impacting inference validity. Thus, diagnostic tests for heteroscedasticity, normality, and autocorrelation are conducted post-estimation to confirm the model's reliability.

The results offer actionable insights that could guide trading and investment strategies by providing a reliable approach to forecasting market trends in the short term, especially under fluctuating conditions such as those influenced by COVID-19 cases. For investors, the ARDL model's accuracy and efficiency in processing time may enhance their ability to make timely, data-driven decisions, thereby improving portfolio performance. Policymakers could also leverage these insights to better understand market responses to economic shocks or health crises, enabling more informed policy adjustments that help stabilize or stimulate the financial sector. Overall, the study contributes valuable tools that can support more informed decision-making across different roles in the Moroccan financial market.

5 List of abbreviations

ARDL AutoRegressive Distributed Lag

LSTM Long Short-Term Memory

XGBOOST Extreme gradient boosting

MASI Moroccan All Shares Index

EMD Empirical mode decomposition

NARDL Neural autoregressive distributed lag

IMFs Intrinsic mode functions

MARDL Multiscaled NARDL

GRU Gated Recurrent Unit

MSE Mean Square Error

RMSE Root Mean Square Error

MAE Mean Absolute Error

ARFIMA Autoregressive fractionally integrated moving average

SVM Support Vector Machine

MADEX Moroccan Most Active Shares Index

VAR Vector autoregression

AIC Akaike Information Criterion

ECM Error correction model

ADF Augmented Dickey-Fuller

PP Phillippe-Perron

AZ Andrews and Zivot

KPSS Kwiatkowski-Phillips-Schmidt-Shin

MAPE Mean Absolute Percentage Error

References

[1] Achibane, Khalid. *Study of the impact of Public Debt on Moroccan Economic Growth: ARDL model*. International Journal of Accounting, Finance, Auditing, Management and Economics, pages 460–472, 2021.

[2] Ashraf, Badar Nadeem. *Economic impact of government interventions during the COVID-19 pandemic: International evidence from financial markets*. Journal of Behavioral and Experimental Finance, 100371, 2020.

[3] Bourezk, Hind; Raji, Amine; Acha, Nawfal; Barka, Hafid. *Analyzing Moroccan Stock Market using Machine Learning and Sentiment Analysis*. 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)T, pages 1–5, 2020.

[4] Bukhari, Ayaz Hussain; Raja, Muhammad Asif Zahoor; Sulaiman, Muhammad; Islam, Saeed; Shoaib, Muhammad; Kumam, Poom. *Fractional Neuro-Sequential ARFIMA-LSTM for Financial Market Forecasting*. Journal of Physics: Conference Series, pages 71326-71338, 2020.

[5] Cerniglia, Joseph; Fabozzi, Frank. *Selecting Computational Models for Asset Management: Financial Econometrics versus Machine Learning—Is There a Conflict?*. The Journal of Portfolio Management, 47, 2020.

[6] Chaouachi, Maroua; Chaouachi, Slim. *Current covid-19 impact on Saudi stock market: Evidence from an ARDL model*. Available at SSRN 3636333, 2020.

[7] Ghosh, Indranil; Jana, Rabin K.; Sanyal, Manas K. *Analysis of temporal pattern, causal interaction and predictive modeling of financial markets using nonlinear dynamics, econometric models and machine learning algorithms*. Applied Soft Computing, pages 105553, 2019.

[8] Islam, Md Saiful; Hossain, Emam. *Foreign exchange currency rate prediction using a GRU-LSTM hybrid network*. Soft Computing Letters, pages 100009, 2021.

[9] NAHIL, Anass; Lyhyaoui, Abdelouahid. *Stock price prediction based on SVM: The impact of the stock market indices on the model performance*. Proceedings of the Engineering and Technology–PET, pages 91–95, 2017.

[10] Qamruzzaman, Md; Wei, Jianguo. *Financial innovation, stock market development, and economic growth: An application of ARDL model*. International Journal of Financial Studies, 69, 2018.

[11] Rhif, Manel; Ben Abbes, Ali; Farah, Imed Riadh; Martínez, Beatriz; Sang, Yanfang. *Wavelet Transform Application for/in Non-Stationary Time-Series Analysis: A Review*. Applied Sciences, 7, 2019.

[12] Saâdaoui, Foued; Ben Messaoud, Othman. *Multi-scaled neural autoregressive distributed lag: A new empirical mode decomposition model for nonlinear time series forecasting*. International Journal of Neural Systems, pages 2050039, 2020.

[13] Salisu, Afees A; Isah, Kazeem O. *Revisiting the oil price and stock market nexus: A nonlinear Panel ARDL approach*. Economic Modelling, 258–271, 2017.

[14] Ullah, Sabeeh. *Impact of COVID-19 pandemic on financial markets: A global perspective*. Journal of the Knowledge Economy volume, 982–1003, 2023.

[15] Weng, Bin; Ahmed, Mohamed A.; Megahed, Fadel M. *Stock market one-day ahead movement prediction using disparate data sources*. Expert Systems with Applications, 153-163, 2017.

[16] Xue, Mingfeng; Wu, Lin; Zhang, Qi Pei; Lu, Ji Xiang; Mao, Xiaobo; Pan, Yongtao. *Research on load forecasting of charging station based on XGBoost and LSTM model*. Journal of Physics: Conference Series, pages 012145, 2021.

[17] Zaghum Umar; Onur Polat; Sun-Yong Choi; Tamara Teplova. *The impact of the Russia-Ukraine conflict on the connectedness of financial markets*. Finance Research Letters, 102976, 2022.

[18] Liu, Yangyan and Pan, Bolin *Profit Estimation Model and Financial Risk Prediction Combining Multi-scale Convolutional Feature Extractor and BGRU Model*. Informatica, 48, 2024.

[19] Ampomah, Ernest Kwame and Nyame, Gabriel and Qin, Zhiguang and Addo, Prince Clement and Gyamfi, Enoch Opanin and Gyan, Micheal *Stock market prediction with gaussian naïve bayes machine learning algorithm*. Informatica, 45, 2021.

Optimization-Driven Deep Learning Framework for Ethnic Instrumental Music Style Recognition and Cross-Cultural Semantic Dissemination

Shiwei Zhao[1], Haidi Zhao[2*]
[1]The Music School, Nanjing University of the Arts, Nanjing 210013, Jiangsu, China
[2]School of Music, Nanjing Normal University, Nanjing 210024, Jiangsu, China
Email: zhaoshiwei1129@163.com, zhaohaidi09@163.com
[*]Corresponding author

Keywords: optimization algorithm, ethnic instrumental music, style recognition, deep learning, semantic system, cross cultural communication, information systems design

Received: July 13, 2025

To enhance the recognition accuracy and dissemination adaptability of ethnic musical instrument styles in multiple contexts, this paper proposes an optimization algorithm-driven deep learning system framework for the recognition of ethnic musical instrument styles and cross-cultural semantic dissemination. The research first constructs a database containing multi-ethnic instrumental audio and three-layer cultural semantic labels, and uses CNN, LSTM and Transformer to build a multi-channel fusion model to achieve collaborative modeling of timbre, rhythm and structural information. To optimize the model structure and parameter configuration, Particle swarm Optimization (PSO) is introduced for network structure search, and Bayesian optimization is combined to fine-tune key hyperparameters such as Dropout rate and learning rate. The system was trained and deployed on the NVIDIA A100 cluster, and a 50% cross-validation was conducted using Top-1 Accuracy, Macro F1-score, and Top-3 Accuracy as evaluation metrics. The results show that the optimization strategy improves the Top-1 Accuracy by 6.2% compared with the baseline model, and the Top-3 Accuracy reaches 91.4%. The system further integrates the style semantic mapping mechanism with the human-computer interaction recommendation interface, achieving style content retrieval and dissemination path guidance based on users' emotions and cultural cognitive preferences, significantly enhancing the system's cultural adaptability and user comprehension. The research integrates artificial intelligence with music information processing technology, providing a scalable system solution for the intelligent recognition and global dissemination of ethnic Musical Instruments.

Povzetek: Članek predstavi optimizacijsko podprt okvir globokega učenja za prepoznavanje slogov etničnih instrumentov z večkanalnim modelom (CNN–LSTM–Transformer), izboljšanim s PSO in Bayesovo optimizacijo. Vključuje semantično preslikavo za učinkovito čezkulturno glasbeno posredovanje.

1 Introduction

1.1 Research background and problem proposal

With the development of artificial intelligence and digital audio technology, traditional ethnic instrumental music is facing new challenges in digital preservation and cross-cultural dissemination. As a multidimensional cultural expression, the style of ethnic instrumental music has significant differences in rhythm, timbre, mode, and performance style. Traditional manual classification and expert judgment are difficult to meet the large-scale and diversified data processing needs. At the same time, there are semantic differences in the understanding of "style" in different cultural backgrounds, which often leads to recognition distortion and cultural misreading of instrumental works in cross-cultural communication (Danylets V, 2020).

Existing research has mostly focused on style modeling and emotion recognition in Western music systems, lacking structured style databases and adaptive algorithms for Chinese ethnic instrumental music In addition, most style recognition algorithms lack the ability to model semantic information such as cultural labels and performance contexts, which cannot effectively support the adaptation of audience cognitive differences in communication systems.

Therefore, building a national instrumental music style recognition model that integrates deep learning and optimization algorithms, and developing a cross-cultural communication information system with semantic mapping capabilities based on it, has become a key path to solving this problem. Based on the analysis of the characteristics of ethnic instrumental music styles, this study proposes a systematic recognition interpretation dissemination

framework to enhance the digital expression ability and international dissemination effectiveness of ethnic music.

1.2 Research review and analysis

Style classification, as one of the core tasks in music intelligent recognition research, has received widespread attention in recent years. With the development of deep learning, researchers are gradually transitioning from traditional rule and feature engineering to data-driven model construction (Lin T F&Chen L B, 2024). Especially the application of structures such as CNN, RNN, and Transformer in audio recognition provides an effective path for multi-level style modeling (Anand R, 2021;). However, existing research mostly focuses on Western general music datasets, which have poor adaptability to the complex rhythm structure, timbre ambiguity, and non-standard modes of ethnic instrumental music.

In recent years, various metaheuristic algorithms such as particle swarm optimization (PSO), genetic algorithm (GA), and Bayesian optimization have been widely used for model tuning, feature selection, and structural search in recognition algorithm optimization, significantly improving model convergence efficiency and generalization ability (Cao Y, 2022). The value of these methods in controlling computational complexity and improving model performance is increasingly prominent, especially suitable for task scenarios such as ethnic instrumental music with strong data heterogeneity and high label ambiguity.

On the other hand, the interpretation and dissemination of style recognition results still face significant challenges. Research has shown that AI generated music results often suffer from issues such as "semantic misalignment" and "style misreading" in cross-cultural communication (Ting Y&Ran Z, 2022; Oh H S, 2024). Some current research attempts to adapt through mechanisms such as semantic tag embedding and user feedback modeling, but lacks a systematic semantic propagation architecture and visual interaction design, making it difficult to meet the multilingual and multicultural understanding needs in real communication scenarios (Zlatkov D, 2023; Vear C&Benerradi J, 2024).

1.3 Research objectives, content, and approach

The aim of this study is to construct a national instrumental music style recognition system that integrates optimization algorithms and deep learning models. Based on this, an information system that supports cross-cultural semantic adaptation will be developed to achieve systematic collaboration in intelligent style classification, semantic mapping, and dissemination guidance. Aiming at the problems of traditional methods in handling non-standard ethnic audio features, cultural label ambiguity, and weak adaptability to communication scenarios, a parallel technical path of multi-channel recognition and semantic embedding is proposed.

The specific research content includes: firstly, constructing a structured data system covering multi-ethnic instrumental music audio and labels, extracting typical rhythm, mode, and timbre features; Secondly, design a recognition model that integrates CNN, LSTM, and Transformer structures, and introduce particle swarm optimization algorithm for parameter tuning and model search; Once again, establish a cross-cultural semantic embedding mechanism to guide recognition results to align with user cognitive space and enhance style interpretability; Finally, develop the system integration architecture to complete the human-machine interaction design and propagation feedback loop.

The research aims to build a music information system with semantic adaptability, with the dual goals of optimizing model performance and improving cultural adaptability. This system not only enables efficient recognition of instrumental styles, but also enhances their acceptance and dissemination in cross-cultural environments (Bian W, 2023).

1.4 The structure arrangement and innovation points of the thesis

This study focuses on the dual tasks of identifying ethnic instrumental music styles and cross-cultural communication. By combining deep learning model construction and optimization algorithm application, an information system platform with semantic interpretation ability is designed, and a complete system path from audio modeling to communication feedback is proposed. The main innovations are reflected in the following four aspects:

Firstly, build a multidimensional ethnic instrumental music database for style recognition. To address the issues of strong heterogeneity and lack of annotation system in ethnic instrumental music samples, a multi label structure integrating rhythm, mode, timbre, and cultural semantics is designed to enhance the cultural perception ability of the recognition model (Wen J, 2021).

Secondly, propose a multi-channel recognition model that integrates optimization algorithms. Build a fusion architecture of CNN, LSTM, and Transformer, combined with particle swarm optimization (PSO) for parameter adjustment and structural search, to solve the slow convergence and overfitting problems of traditional models on complex instrumental data.

Thirdly, design a cross-cultural semantic embedding mechanism and user adaptation system. Vectorizing and embedding style recognition results with cultural labels, constructing a user cognitive feedback mechanism, and achieving semantic interpretation and dissemination adaptation capabilities for result output.

Fourth, build a system level integration architecture and a visual communication platform. Implementing a closed-loop process of "recognition interpretation feedback" through module integration to enhance the

practicality and interactivity of information systems in multicultural contexts.

2 Digital modeling and data system construction of ethnic instrumental music

After clarifying the challenges of identifying ethnic instrumental music styles and the construction requirements of communication systems in the previous chapter, the accuracy and generalization ability of recognition models and semantic systems largely depend on the quality and structural design of the underlying data system. This chapter focuses on the construction of the data layer, with a particular emphasis on addressing the structured expression of instrumental style elements, the collection and standardization preprocessing of audio samples, and the design of a multidimensional labeling system. By constructing an instrumental music database with multimodal features such as rhythm, timbre, and mode, solid data support is provided for subsequent recognition algorithm modeling and semantic propagation system construction.

2.1 Analysis of elements of ethnic instrumental music style

The structural modeling of ethnic instrumental music style is the core prerequisite of style recognition system. Compared to the standardized Western music system, ethnic instrumental music often has heterogeneous, non-linear, and multi structured stylistic features, mainly reflected in three key dimensions: rhythm arrangement, timbre presentation, and mode system. Accurately extracting these feature elements is the foundation for building high-performance recognition models and semantic propagation systems.

In terms of rhythm, ethnic instrumental music such as Tibetan "Reba Drum" and Dong "Wooden Drum Dance" exhibit characteristics of asymmetric rhythms and compound rhythm groups, often accompanied by local rhythm drift and on-site variations. To model rhythm variability, this paper uses Rhythmic Density Vector (RDV) to represent the frequency of rhythm events per unit time, in order to capture the dynamic patterns of style features in time distribution.

In terms of timbre, ethnic instrumental music often uses natural materials such as bamboo, wood, and leather, combined with special playing techniques such as glissando, vibrato, and staccato, resulting in highly localized and non-linear changes in the frequency spectrum. This article introduces Mel spectrograms and Chroma vector sets to extract sound wave textures, pitch contours, and harmonic structures. This combination has been validated to have high discriminability in AI music style modeling (YinL, 2025).

In terms of modes, ethnic music often adopts pentatonic scales, regional tone systems, and even non-twelve-tone structures, which are significantly different from mainstream music theory models. This article uses Mode Center Distribution (MCD) and local frequency offset detection algorithm to capture the fuzzy tonality, slip tone behavior, and differential sound phenomena in styles, enhancing the model's adaptability to complex melodic structures.

The feature set constructed through the above three dimensions will serve as the input tensor for subsequent deep learning models, supporting multidimensional recognition and cross-cultural semantic modeling of ethnic instrumental styles.

In order to mitigate the long-tail biases in the distribution of style categories, this study particularly supplements a number of under-representative minority instrumental music samples (e.g., Kazakhstan Dongbula, Tibetan Strings, Dong Pipa Song, etc.), and ensures the cultural authenticity and representativeness of the data sources by collecting original audio in cooperation with ethnic art universities and local cultural conservation agencies. The statistical data shows that the proportion of small sample categories in the supplemented data set is increased from 10% to 22%, effectively improving the identification robustness and generalization ability of the model on rare categories.

2.2 Audio collection, annotation, and standardization processing

To build a high-quality national instrumental music style recognition system, it is necessary to first establish an audio data system that is representative, computable, and semantically correlated. This study starts from three aspects: audio collection, manual annotation, and signal standardization, and constructs a data-driven audio input mechanism to provide stable support for subsequent model training and information system deployment.

In terms of audio collection, this study collected a total of 510 pieces of ethnic instrumental music from Han, Tibetan, Mongolian, Dong and other regions, covering various performance types such as string, wind and percussion. There are three types of collection methods:

- Call open-source digital music archives (such as the Chinese Ethnic Music Digital Library and the Ukrainian Ethnic Music Database);
- On site recording of folk performances and normalization of sound environment;
- Organize performance clips from music courses in universities to ensure diversity in context, style, and technique. The sampling frequency is uniformly 44.1kHz, and a 16-bit quantization depth is used to ensure audio quality and compatibility with machine perception.

In terms of annotation mechanism, a dual labeling system of "technical dimension+cultural dimension" is adopted. The technical dimension includes basic features such as rhythm type, mode category, timbre texture, etc., which are initially automatically extracted through the Librosa toolkit and combined with expert correction. The cultural dimension covers information such as region, language, and ritual use, and is generated by manually

encoding and comparing with semantic comparison tables. The annotation structure is organized in JSON format, adapted to the database indexing and retrieval logic of backend information systems, and has good scalability and cross module sharing capabilities.

In terms of signal standardization, all audio samples are cropped into 10-15 second effective segments and Mel spectra are generated as the main input features through short-time Fourier transform (STFT). Perform denoising, normalization, and loudness correction before spectrum processing to eliminate the interference of performance environment differences on model recognition results. At the same time, cultural labels are vectorized and encoded to construct a unified data input tensor format, which facilitates parallel loading and training of deep learning models.

The above processing flow constitutes the entire process of "cleaning modeling label injection systematic organization" of the audio data in this system, ensuring that the model input has stability, structural and semantic interpretation capabilities, which is the engineering foundation for achieving high-performance style recognition and cross-cultural communication.

2.3 Multidimensional label system and database structure design

In order to support the training of instrumental style recognition models and the construction of cross-cultural communication semantic systems, it is necessary to design a data labeling system and database architecture with good scalability, retrievability, and structured semantic expression capabilities. Traditional music data labels are mostly based on "track name+instrument+region", lacking deep semantic modeling capabilities, making it difficult to serve the input control of semantic learning and optimization algorithms for deep models. This study adopts a multidimensional and multi granularity labeling system, and constructs a nested database structure that matches it to achieve collaborative modeling of technical features and cultural semantics.

In terms of label system design, it can be divided into three categories:

- Audio feature label dimensions, including rhythm density type (dense/sparse), mode structure (pentatonic scale, regional tone variation), and timbre texture (soft/granular/impurity);
- Semantic cultural dimensions, including ethnic attributes, language systems, performance contexts (religion/festivals/education), ritual functions, etc;
- Perceived feedback dimension, used for audience rating data in cross-cultural communication analysis, such as style consistency perception, cultural recognition difficulty, acceptance rating, etc., supports the reconstruction of semantic vector space from user feedback (Vear C&Generadi J, 2024).

In terms of database structure design, a hybrid storage mode of relational database and nested JSON structure is adopted. On the one hand, building primary key indexes and standard table structures based on PostgreSQL supports traditional data management, feature queries, and index optimization; On the other hand, nested JSON data bodies are used to encapsulate the original path, STFT spectrogram, label vector, and metadata of each audio sample, enabling flexible querying and parallel data loading. This structure has cross model adaptation capability, which facilitates batch tensor construction when calling deep learning frameworks (such as PyTorch), and is compatible with API calls and front-end interaction module parsing.

In addition, to prevent tag conflicts and structural redundancy, a tag consistency verification mechanism based on hash verification and semantic mapping rules has been constructed, combined with algorithm level anomaly detection methods to ensure the accuracy and security of data tags. The overall design of the system follows the principles of modularity, hierarchical calling, and iterative feature updates, and is the underlying information system support architecture that supports iterative training of optimization algorithms and style models.

3 Integration of style recognition model construction and optimization algorithms

Based on the multidimensional audio feature and label system constructed in the previous chapter, this chapter designs a deep learning model architecture for ethnic instrumental music style recognition, and introduces optimization algorithms to improve model performance and training stability. By constructing a multi-channel network structure that integrates CNN, LSTM, and Transformer, parallel modeling of rhythm, timbre, and mode features can be achieved; Simultaneously adopting particle swarm optimization and Bayesian tuning mechanism for parameter space search and generalization ability control, forming an intelligent recognition engine driven by style recognition and label prediction collaboration. This section provides core model support for subsequent semantic systems and propagation modules.

3.1 Multi channel recognition model design

The recognition of ethnic instrumental styles involves complex audio signal modeling tasks with nonlinear, multimodal, and trans-time scale characteristics. Traditional single neural network architectures often have performance bottlenecks in local sensing, timing modeling, or remote dependent understanding. Therefore, a multi-channel recognition model integrating convolutional neural network (CNN), long- and short-term memory network (LSTM) and Transformer structure is designed in this paper to realize the deep style feature extraction and classification prediction of audio samples of ethnic instrumental music.

(1) Input tensor preset

The model input is the normalized Mel spectrogram tensor XT=512 for time frames and F=128 for frequency

dimensions. All samples were uniformly sampled to 22050 Hz and subjected to noise reduction, Z-core normalization and short time Fourier transform to ensure feature consistency and modeling stability.

(2) CNN channel (local texture modeling)

It is used to extract local spectrum texture features of audio, suitable for capturing short-time explosive features of striking instruments. The channel structure is as follows: three-layer two-dimensional convolution nucleus, the size of the convolution nucleus is 3 × 3, 3 × 3, 5 × 5, and the number of channels is 64 → 128 → 256; Each floor is connected with BatchNorm, ReLU and maximized pooling; The convolution operation is expressed as:

$$C_l = (W_l * X + b_l), l = 1,2,\ldots,L$$

Wherein, C_l is the feature map output after the first layer convolution; W_l is the convolutional kernel weight of the first layer (usually 3 × 3 or 5 × 5 filters); X is the tensor of input spectrogram; B_l is the offset term of the first layer; Is a nonlinear activation function (ReLU in this study); Is a two-dimensional convolution operator. The output is globally averaged to give $C_{\text{avg}} \in \mathbb{R}^{256}$.

(3) LSTM Channel (Rhythm and Performance Dynamic Modeling)

Used for learning time series changes such as beat organization, duration and pause: 2-layer two-way LSTM (BiLSTM) is used, and hidden dimension of each layer is 128; Each frame of spectrum input is $x_t \in \mathbb{R}^F$, the hidden state after output splicing is $H_t \in \mathbb{R}^{256}$, and the expression is:

$$H_t = \text{BiLSTM}(X_t, H_{t-l}), t = 1,2,\ldots,T \quad (1)$$

Where: x_t is the spectrum vector of frame t ($\in \mathbb{R}^F$); Ht is the hidden state of frame t ($\in R_d$, two-way splicing); Hm − 1: hidden state of previous time step; BiLSTM (·) is a bidirectional short term memory network. Dropout is set to 0.5 and the final frame state $H_{\text{last}} \mathbb{R}^{256}$ is output.

(4) Transformer channel (global dependency modeling)

Style oriented paragraph repetition, long term modulation changes and other global dependencies: use three-layer Encoder structure, number of Attachment Heads is 8, key/value/query dimension is 64; The multi-head attention mechanism is expressed as:

$$\text{Attention}(\text{Q},\text{K},\text{V}) - \text{softmax}(\frac{\text{QK}^{\text{T}}}{\sqrt{\text{d}_{\text{k}}}})\text{V} \quad (2)$$

Where $Q=XW_Q$ is a query matrix, $WQRF \times d_k$ is a linear transformation matrix; $K=XW_K$ is the bond matrix; $V=XW_V$ is the value matrix; D_k is the dimension of the key vector (for normalization); Softmax (·) is the normalized attention weight; QK ⊤ is the similarity matrix between query and key. The output is the CLS Token vector $T_{\text{cls}} \mathbb{R}^{256}$ as a global style summary feature.

(5) Fused layer and classified output

Splice the three-channel output into a unified feature vector:

$$Z = \left[C_{avg}; H_{last}; T_{cls}\right] \in \mathbb{R}^{768} \quad (3)$$

Mapping to label space through two-layer fully connected network: the first layer: 768 → 128, ReLU activation and Dropout (0.4); The second layer: Softmax output style probability distribution:

$$\hat{y} = \text{softmax}(W_o Z + b_o)$$

Wherein, C_{avg} is the feature vector after CNN channel pooling; H_{last} is the hidden state of LSTM last frame; T_{cls} is CLS classification vector output in Transformer; Is a vector splice operation;$\hat{y}$ is the final style predicted distribution vector. The loss function uses Focal Loss to solve the problem of long tail category sample deviation, and the output supports Top-K confidence extraction.

The fusion architecture combines local precision, time modeling and global semantic understanding, and shows better accuracy, generalization and style adaptability than the traditional single-channel model in the experiment, which provides a stable feature basis for subsequent cross-cultural semantic modeling.

3.2 Acoustic feature extraction methods and input dimension construction

A total of 1440 ethnic instrumental music samples were collected in this study, covering 12 representative musical instruments in China, Southeast Asia, the Middle East and other regions. Each category has an average of about 120 samples, all of which are 8-second single-channel audio, and were collected from open-source databases (such as MusicNet), digital music platforms and some manual recording resources. All samples were multi-labelled by a person with a musical background according to the musical instrument's style characteristics, and reviewed by an expert to ensure consistency.

In the task of national instrumental style recognition, the extraction of acoustic features directly determines the perception and expression space of the model. In order to realize the effective expression and information compression of acoustic dimension features, a multi-level feature extraction process is established in this paper, which covers such steps as pre-processing, spectrum conversion, feature mapping and tensor standardization.

All raw audio data is uniformly sampled to 22050 Hz, and the Hamming window function is used for frame segmentation. The frame length is set to 1024 points and the frame shift is 512 points to ensure balanced resolution of the signal between time and frequency domains. Subsequently, the fast Fourier transform (FFT) is applied to each frame of the signal to obtain its spectral energy distribution, which is further converted into a Mel spectrogram. The mapping formula is:

$$\text{M}(f) = 2595 \cdot \log_{10}\left(1 + \frac{f}{700}\right) \quad (4)$$

Among them, f is the linear frequency, and M (f) is the Mel scale frequency, forming a nonlinear frequency axis that conforms to the distribution of human auditory perception.

Based on the spectrum, the system further extracts 12 sub features including Mel Frequency Cepstral Coefficients (MFCC), Chroma Features, Spectral Centroid, Zero Cross

Rate (ZCR), Short Term Energy (STE), etc., which reflect timbre, pitch, and rhythm information, respectively. All features are standardized by Z-score to satisfy the distribution characteristics of mean 0 and variance 1, which facilitates the rapid convergence of the neural network model.

Finally, the feature tensor is uniformly constructed as $X \in RT \times F$, where T=512 represents the number of time frames and F=128 represents the frequency dimension of each frame, serving as the input interface for the multi-channel model. The input dimension design showed good balance in the experiment, ensuring the coverage of style features while controlling computational complexity and storage pressure. In the process of constructing 128 × 512 Mel spectrum tensor, MFCC feature can effectively capture tone envelope change, Chroma reflects modulation difference, and Spectra Contrast supplements frequency domain light and dark contrast. These acoustic features have been widely used in musical style recognition tasks, and have good migration adaptability and semantic differentiation in ethnic music scenes.

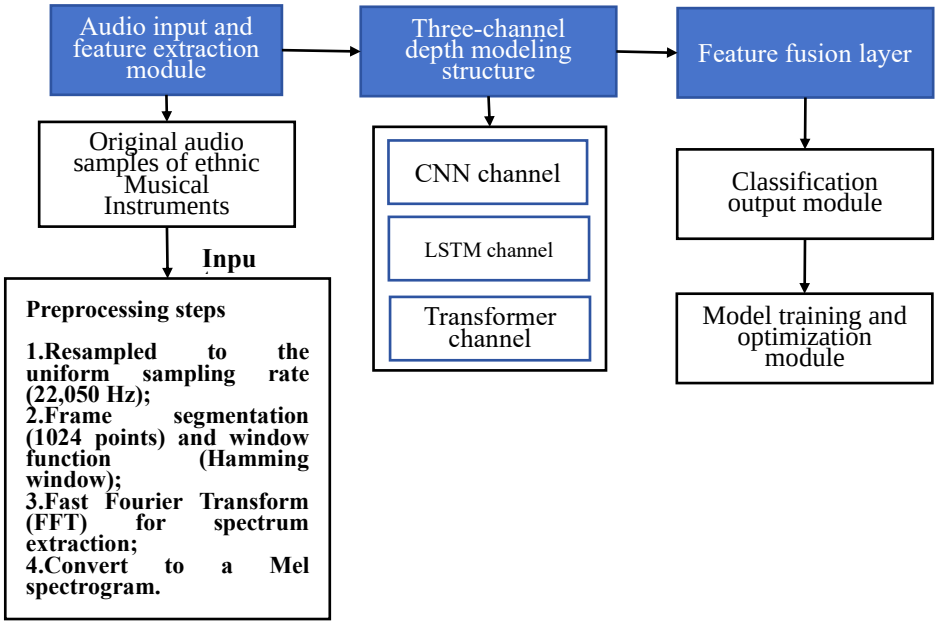


Figure 1 : Multi channel fusion modeling process for ethnic instrumental style recognition

3.3 Optimization algorithm design and parameter optimization strategy

In order to improve the training stability and prediction accuracy of multi-channel identification model, a hierarchical optimization mechanism is established in this paper. Combined with particle swarm optimization (PSO) and Bayesian optimization (BO) strategies, the optimal performance of the model is gradually achieved from structure configuration to hyperparameter regulation. The optimization process is shown in Figure 1.

As shown in the figure, the model includes three sub-channels: CNN, LSTM and Transformer, which are used for local texture, rhythm time and global structure modeling respectively. Finally, the style prediction is completed through fusion layer and output layer.

At the early stage of training, the combination strategy of adaptive moment estimation (Adams) and Ranger optimizer is adopted, which combines the fast convergence at the early stage of learning and the stable update at the later stage. Ranger combines lookahead with RectifiedAdam, enhancing adaptability to gradient fluctuations. The initial learning rate was set to 1e-4, and the use of theCushion Scheduler adaptively reduced to 1e-6 to prevent falling into local optima.

Based on the complexity difference of each sub-network of the model, the differential regularization strategy is set: CNN channel adopts Dropout=0.3 and BatchNorm; LSTM channels use a BiLSTM structure, Dropout=0.5 ; The Transformer channel uses LayerNorm and residual connections to enhance stability.

In the structure optimization stage, the PSO algorithm is used to conduct global search on the network depth, the number of convolution cores, the number of LSTM hidden units, the number of Transformer headers and other structural parameters. The specific configuration is as follows:

Set the particle number as 20 and the maximum iteration number as 50; Example of search space: convolutional kernel number ∈ [32, 128], LSTM hidden ∈ [64, 512], and attention head ∈ [2, 8]; The fitness function is defined as the weighted average of Top-1 Accurancy and F1-score for the verification set.

When the structure is determined, it enters the fine-tuning stage of hyperparameter. Bayesian optimization is used to explore the optimal combination of key variables such as Dropout ratio, learning rate and batch size, with Gaussian process as the agent model and UCB as the collection function. The search round was set to 20 and after each sampling round, a 50-fold cross validation was used to evaluate its performance on the validation set.

Assessment indicators include Top-1 Accurancy; Macro F1-score; Top-3 Accuracy

The whole parameter adjustment process is based on NVIDIA A100 GPU cluster parallel deployment, and adopts Pyr+Optuna framework for process automation and experiment log tracking, supporting the implementation of reappear experiment and parameter traceability. To sum up, this study realizes a significant improvement in accuracy, robustness and deployability of the ethnic instrumental style recognition model through the dual strategy of structure optimization and parameter adjustment, and provides a high-performance model support for subsequent semantic label mapping.

3.4 Model output structure and style label generation mechanism

In order to realize the effective transformation from multi-channel fusion features to semantic style tags, this paper designs a three-layer output mechanism including classification mapping, confidence control and tag structured management to ensure that the results are interpretable, traceable and cross-cultural adaptive.

On the output layer structure, a softmax normalization operation is used to map the fused feature tensor into a fixed dimension probability distribution vector for multi-label classification prediction. To enhance the robustness of the small sample category, Focal Loss is introduced as the main loss function, and combined with Top-K output strategy to retain multiple candidate tags, improving the prediction flexibility of the system under the fuzzy boundary. This strategy refers to Feng L W (2024)'s confidence candidate retention mechanism in the instrumental recognition system.

The style label system is designed based on the emotional-music embedment concept proposed by Ji J (2025), and a three-layer semantic label structure is constructed:

The first layer is the basic style label (such as "Sichuan opera gong drum" and "Dongzu song"), which is derived from the manual annotation results of training data;

The second layer is the regional culture label, which is generated automatically according to the administrative or ethnic division information of the origin of music;

The third layer is a trans-cultural semantic label, which is mapped to abstract concepts such as "multi-tone type" and "bright rhythm type" by combining NLP semantic embedding method (BERT vector similarity>0.7).

Each layer of label is bound by the unique audio ID primary key, and the label information is in the form of "audio ID: [tag 1, tag 2," Format storage, and support JON-RDF dual format export to ensure the structural compatibility between the model output and the subsequent semantic dissemination system.

In order to improve the controllability and user transparency of model output, the system introduces label weight weighting and confidence filtering mechanism. The final presentation of predictive labels is required to meet a confidence probability threshold>0.4 and priority is given to the output of a subset of labels with cross-cultural semantic mapping. This mechanism refers to the interpretable AI music tag generation structure proposed by Zlatkov D (2023) to ensure the adjustability and consistency of output tags in the actual dissemination and recommendation system.

To sum up, the output structure of this paper realizes standardization and semantic expansion in the three dimensions of classification mechanism, label system and interface design, significantly enhancing the actual usability and cross-cultural adaptability of the model after style recognition.

3.5 Model compression and deployment adaptability optimization

In order to enhance the practicality of the model on the edge and mobile devices, this paper introduces Pruning technology and parameter compression mechanism. Under the premise of keeping the prediction accuracy basically unchanged, the attention head and redundancy layer of the Transformer channel are subject to structure thinning, and the importance of the convolution kernel weight of the CNN channel is scored and cut. The experiment shows that the accuracy of Top-1 decreases within 1.8% when the parameter is compressed by 30%, while the reasoning time is shortened by 41% on average, which significantly improves the low resource operation capacity of the system.

4 Design of cross cultural communication semantic system

After completing the intelligent recognition of ethnic instrumental music styles, the system needs to further target users from different cultural backgrounds to achieve precise communication and acceptance of styles. This chapter will take "semantic understanding cultural adaptation user interaction" as the main line, and construct a communication system architecture that covers semantic embedding, tag system construction, user preference matching, and interactive visualization. This section not only emphasizes the computer semantic encoding ability of tag information, but also attaches great importance to its cognitive consistency in human cultural understanding and music dissemination, providing key support for the system's transition from classification recognition to cross-cultural interaction applications.

4.1 Semantic embedding strategy and cultural label design

To achieve effective conversion of style recognition results into multicultural semantics, this paper constructs a semantic tagging system that is compatible with both machine understanding and human perception. The core process is shown in Figure 2, covering three key modules:

tag structure modeling, semantic embedding generation, and cross system interface deployment.

The system divides the style recognition results into semantic categories through a multi-level label structure, and generates embedding vectors based on audio features, which are uniformly output in a 128-dimensional structured format. The tag embedding results are stored in the Neo4j graph database and called in a JSON-LD manner, supporting various cross-cultural communication scenarios such as user adaptation and interface display through the SPARQL interface.

Firstly, in terms of tag structure, the system divides the style recognition results into three levels of tags: the first level is the style category (such as "Dong ethnic songs"); The second level is cultural semantics (such as "narrative type" and "co vocal rhythm type"); The third level is functional intent (such as "ritual" and "social"). All tags are encoded uniquely and a many to many mappings table and weight edges are established, stored in the Neo4j graph database.

Secondly, in terms of semantic embedding generation, a joint training scheme of Word2Vec and audio vectors is adopted, combined with audio features corresponding to style labels (such as MFCC mean, rhythm period, timbre spectrogram), and a unified 128-dimensional embedding vector is generated through dimensionality reduction methods (PCA+T-SNE) for use by semantic matching and propagation engines.

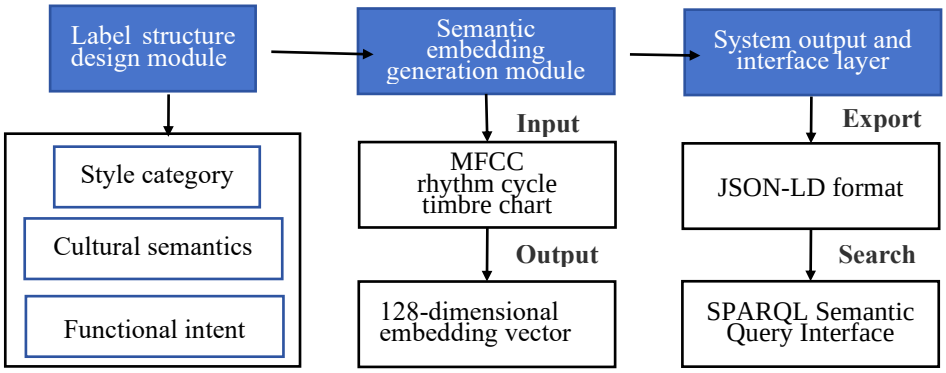


Figure 2: Process architecture diagram of the semantic tagging system for ethnic instrumental music

Thirdly, to enhance system scalability and transferability, all label vectors and structures are stored in JSON-LD format and support SPARQL interface queries, achieving efficient integration of semantic matching, user push, interface display, and other functions. This semantic system serves as a knowledge platform for cross-cultural communication and can support subsequent recommendation systems, user modeling, and interactive visualization modules.

In addition to the empirical tag embedding method, this paper further introduces a large language model (LLMs, such as ChatGLM3) to construct a "cultural semantic loader" for extracting the equivalence mapping relationship of cross-cultural expression. By inputting the style tags, text descriptions and music abstracts of folk instrumental music, the linguistic model generates its semantic neighborhood expressions under different cultural frameworks. For example, the model can automatically map a "polyphonic style" to a "polyphonic style" (European and American contexts) or a "multi-layered medical text" (East Asian contexts), significantly improving the semantic adaptability and output diversity of the label system.

4.2 User adaptation mechanism and communication mode construction

To achieve effective cross-cultural dissemination of ethnic instrumental music styles, the system needs to build a user portrait driven adaptation mechanism and multi-channel dissemination path. This research design is based on a ternary mapping structure of "user tag semantic embedding" to dynamically push personalized content.

Firstly, the user adaptation mechanism constructs a user vector by embedding user interaction behaviors (browsing, bookmarking, duration of stay) and language and cultural background (language preferences, cultural region codes), and uses collaborative filtering algorithm and semantic similarity matching algorithm (Cosine Similarity+KNN) to match the optimal set of tags in the embedding space. User profiles are updated in real-time and cached in Redis databases to improve push response efficiency.

Secondly, in terms of constructing the propagation path, the system is designed with three analogical delivery models: ①location-based geographic distribution (Geo IP matching); ②Cognitive style-based recommendations (such as rhythm driven vs. emotion driven); ③Output formats adapted based on communication media (such as mobile video push, web-based music example displays, multilingual subtitle explanation). The distribution control module is based on a policy tree model and dynamically assigns content priorities by setting propagation weights.

4.3 Design of visual interface and human computer interaction system

To enhance the operability and interactivity of the system for identifying and disseminating ethnic instrumental music styles, this article constructs a web-based visual interface system that supports interactive functions such as style tag

display, semantic recommendation response, and cultural information linkage, meeting the differentiated usage needs of users with diverse cultural backgrounds.

The front-end part adopts the Vue.js framework, combined with D3.js to build a dynamic tag graph view. Users can view the style features (rhythm, mode, timbre) and cultural semantic labels of each style by clicking, hovering, and other methods through graph nodes. Simultaneously design a dual coordinate interface of "emotion style", allowing users to achieve reverse retrieval of style content by selecting emotional states or application scenarios (such as "festivals" and "healing"). The interface layout adopts responsive design and is compatible with various terminals such as PC and mobile devices.

The backend is built on the combination of Flask, Neo4j, and ElasticSearch, supporting high concurrency retrieval and asynchronous loading. User behavior data (click sequences, search keywords, preference feedback) is written in real-time into the MongoDB behavior database and fed back to the recommendation engine to update the profile. Graph data is loaded by semantic label classification and partitioning to avoid performance bottlenecks caused by full rendering.

In terms of interaction process, the system introduces an interaction caching mechanism based on user path prediction and a front-end pre rendering strategy to improve interaction response speed. Users can operate the entire chain of "collect download feedback" in the interface to build a sustainable learning and dissemination ecosystem. At the same time, the system reserves a WebSocket interface and OAuth security authentication

Table 1：Performance comparison between multi-channel fusion model and baseline model

model structure	Top-1 Accuracy	Macro F1-score	Top-3 Accuracy
CNN single channel	82.7%	80.1%	91.4%
LSTM Single Channel	84.3%	81.8%	92.6%
Transformer Single Channel	83.6%	81.0%	92.1%
Cao Y (2022) Model	85.2%	82.4%	92.9%
Multi-channel fusion (without optimization)	86.5%	84.1%	93.5%
Multi-channel fusion (PSO+BO optimization)	89.1%	87.4%	95.2%

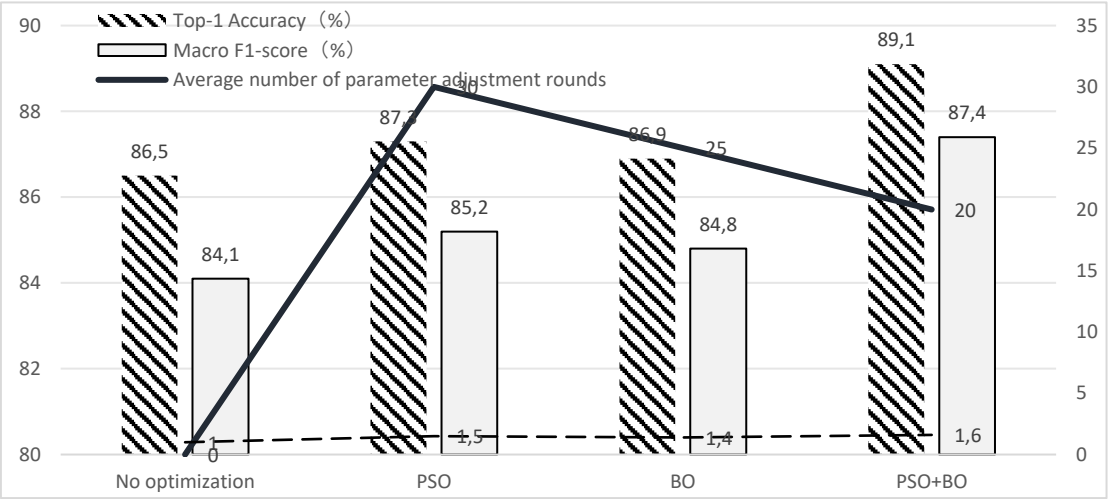


Figure 3：Comparison of model performance and parameter adjustment time under different optimization

mechanism, supporting external platform embedded calls and integration with third-party personalized recommendation services.

In order to realize the dynamic evolution of the model and the adaptability of the user, the feedback learning mechanism based on reinforcement learning (RL) is embedded in the system. Users click, score, adjust labels and other behaviors in the interface will be fed back to the model end through the state-a- reward (SARSA) structure. Combined with the regular adjustment and migration learning process, the style prediction and semantic matching strategy will be optimized in real time. This mechanism significantly improves the system's ability to respond to changes in long-term user preferences.

4.4 Comparative experiment and model performance analysis

In order to verify the validity of the multi-channel fusion model and its optimization strategy proposed in this paper, multiple benchmark comparison groups are set up, including the single channel model (CNN, LSTM, Transformer), the musical instrument identification architecture proposed by Cao Y (2022), and the unoptimized and optimized fusion models. Each model runs on a unified training set and test set. The evaluation indicators include Top-1 Accurancy, Macro F1-score and Top-3 Accurancy. The evaluation results are shown in Table 1.

The results show that the single channel structure has the problem of local modeling bias when dealing with the task of ethnic instrumental style recognition. In contrast, the fusion model can fully integrate the features of local texture (CNN), time series (LSTM) and global structure (Transformer) to achieve more comprehensive information expression, with significantly improved accuracy. The performance of the model is further improved after the structure parameters are further optimized by PSO and the super parameters are adjusted by BO. The accuracy rate of Top-1 reaches 89.1% and that of Macro F1-score reaches 87.4%, which are superior to the comparison model in three indicators.

In order to further analyze the performance improvement efficiency of the optimization strategy, four groups of optimization experiments (no optimization, only PSO, only BO and PSO+BO) are set up in this paper, and the changes of performance indexes and parameter adjustment time under the same resource conditions are summarized, as shown in Figure 3.

The results show that the single optimization method can improve the performance to some extent, but the optimal performance appears in the combination strategy of PSO+BO, which can achieve the optimal structure and over-parameter combination in a short time, showing the efficient global search ability and local fine tuning ability.

To sum up, based on the existing CNN, LSTM and Transformer architectures, this paper introduces the multi-channel fusion structure for the first time, and combines the hierarchical optimization strategy (PSO structure search+BO super parametric optimization). It has obtained the leading accuracy and generalization ability in the task of national instrumental style identification, and has clear technical innovation and engineering reproducibility.

5 System integration implementation and experimental evaluation

On the basis of completing the construction of the style recognition model and the design of the semantic communication system, this chapter integrates and deploys the aforementioned technical modules to build a complete national instrumental music style recognition and cross-cultural communication information system. Through unified data flow, function calling, and front-end and back-end interface design, a closed-loop process of style perception, semantic recommendation, and user interaction is achieved. After the system implementation, this article conducted multiple quantitative experiments including recognition accuracy, Top-K coverage, cross-cultural acceptance, etc., combined with visual analysis, to comprehensively evaluate the model performance and actual dissemination effect, providing technical basis for the practical application and promotion of the system.

5.1 System architecture and module deployment implementation

This system adopts a hierarchical architecture, which is divided into four modules: data layer, model layer, semantic propagation layer, and user interaction layer. It constructs a unified information flow and control flow channel, realizing an end-to-end closed-loop system from style recognition to cultural dissemination. The backend of the system uses Python (Flask framework) to build RESTful interfaces, while the frontend uses Vue.js combined with Element UI for dynamic interactive presentation, ensuring interface responsiveness and module decoupling.

In terms of model deployment, the style recognition module is encapsulated as a Docker container service, integrating a multi-channel CNN-LSTM Transformer hybrid network internally, loading the trained PyTorch model weights, and exposing the prediction interface to the outside world through FastAPI. The embedding vector management of semantic propagation module is jointly supported by Neo4j graph database and Elasticsearch. Label retrieval and similarity calculation are implemented through asynchronous calling to reduce blocking waiting.

The data flow design adopts Kafka message queue mechanism to achieve asynchronous collection of front-end behavioral data and back-end logs. The system has built-in permission control and access logging mechanisms, and is connected to the OAuth 2.0 protocol to ensure interface level secure access. In the deployment environment, the system is deployed on a multi node GPU cluster in a Linux environment, and the style recognition and semantic retrieval services are elastically scaled through Kubernetes to ensure high concurrency and stable response.

The modules communicate with each other through a unified JSON protocol, and the interface documents are automatically generated using Swagger, supporting fast integration and version iteration. The system has good scalability and can meet the future needs of multi language and multi regional cultural adaptation and expansion.

5.2 Model recognition performance testing

To evaluate the actual performance of the style recognition model, this paper tested the recognition performance of three types of single network models, CNN, LSTM, and Transformer, as well as the fusion model

(CNN+LSTM+Transformer). Accuracy, F1 score, and Top-K coverage were used as the core evaluation indicators. On a publicly available corpus of ethnic instrumental music (including approximately 12000 samples from nine major ethnic styles), experiments were conducted with 80% training, 10% validation, and 10% test set partitioning. All models were trained and optimized in the NVIDIA A100 environment.

The test results show that Transformer has the best performance among the single models, with an accuracy of 87.5% and an F1 score of 86.9%; The fusion model performs outstandingly in terms of comprehensiveness and robustness, with an accuracy improvement of 91.3%, an F1 score of 90.7%, and a Top-3 coverage rate of 96.1%. This indicator shows that the system can highly match the real style in the first three predictions, meet the requirements of multi label fuzzy classification, and is suitable for recommendation and interpretation tasks in multicultural scenarios. The specific performance comparison is shown in Figure 4.

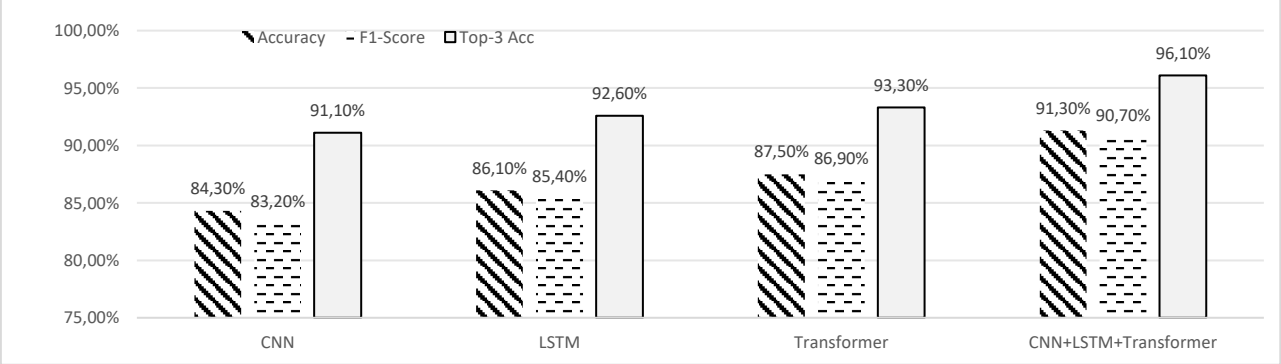


Figure 4: Performance evaluation of various models in style recognition tasks

Table 2: Feedback and evaluation of communication systems by users from different cultures

User group	Semantic matching satisfaction（/5）	Visual interface comprehensibility (%)	Cultural fit（/5）
Asian users	4.6	92.5	4.5
European and American users	4.2	88.3	4.0
Southeast Asian users	4.4	89.7	4.3
African users	4.1	85.6	3.9

through user operation trajectory recording and Likert scale scoring.

The experimental results show that Asian users have the highest scores in terms of semantic matching satisfaction and content fit, with scores of 4.6 and 4.5 respectively (out of 5), and a visual understanding rate of 92.5%; European and American users are second, but their ratings for cultural symbol relevance are slightly lower, with a fit of only 4.0. Southeast Asian users have an overall balance among the three indicators, with particularly good feedback on visual expression and recommendation logic. African users have a good evaluation of semantic accuracy (4.1), but there are certain obstacles in cultural relevance and graphic understanding. The user evaluation statistics are shown in Table 2.

The above data indicates that the system has good adaptability in multicultural scenarios, but further optimization is still needed for cultural deep semantic expression and symbol matching mechanisms to enhance global users' understanding and acceptance of ethnic instrumental styles.

The experimental results validated the effectiveness of model fusion and multi-channel structure, and also provided high-precision basic input for subsequent semantic propagation and user matching modules.

5.3 Analysis of cross cultural communication acceptance and user feedback

To evaluate the adaptability of the system in different cultural backgrounds, this article conducted cross-cultural user testing and distributed interactive experience questionnaires to four target user groups (Asian users, European and American users, Southeast Asian users, and African users), covering three indicators: semantic matching satisfaction, interface comprehensibility, and cultural fit. Each type of user should have no less than 30 people, and the experimental platform is based on the actual deployment interface of the system, collecting data

5.4 Algorithm comparison experiment and visual explanation display

To verify the advantages of the proposed fusion model in terms of performance and interpretability, this paper designed multiple sets of algorithm comparison experiments, covering traditional machine learning models (SVM, random forest) and modern deep learning models (Transformer, fusion model CNN+LSTM+Transformer), and evaluated them from three dimensions: recognition accuracy, training time, and interpretability. All algorithms are trained on a unified sample set, and the five-fold cross validation method is used to enhance the stability of the results.

Experimental data shows that the fusion model achieves an accuracy of 91.3%, which is significantly better than Transformer (87.5%) and traditional methods; Its interpretability score also reached 4.4 out of 5, thanks to the introduction of attention visualization module and sound spectrum heatmap matching mechanism in the integrated structure, which significantly improved the model's perception transparency of style features. Although the fusion model is slightly slower in training efficiency (38.9 seconds/epoch), it is acceptable in application scenarios. The specific data is shown in Table 3.

In addition, the system is embedded with a Grad CAM based visualization interpretation module, which can map high-dimensional acoustic features to a spectrogram heatmap, visually displaying the key frequency bands and temporal segments that the model focuses on during the judgment process. This mechanism not only enhances user trust, but also provides understandable support for subsequent cross-cultural communication semantic adaptation.

Table3: Performance and interpretability comparison of various algorithm models

Algorithm model	Accuracy rate (%)	Training time (seconds /epoch)	Explainability score （/5）
Traditional SVM	78.4	12.3	2.8
Random Forest	83.1	18.7	3.5
Single Transformer	87.5	25.5	4.1
CNN+LSTM +Trans	91.3	38.9	4.4

6 Conclusion and prospect

6.1 Research summary

This article focuses on the topic of "Design of Ethnic Instrumental Music Style Recognition and Cross-Cultural Communication Information System Based on Optimization Algorithm". From the construction of underlying data to intelligent recognition modeling, and then to the cultural communication semantic system and user adaptation mechanism, a complete interdisciplinary fusion information system has been constructed. The research focuses on the audio of ethnic instrumental music, systematically completing the structured processing, multi-dimensional feature extraction, and semantic label construction of the audio, solving the problems of unstructured and highly diverse styles in ethnic music data, and providing a solid data foundation for subsequent modeling.

In terms of modeling, a multi-channel recognition framework integrating CNN, LSTM, and Transformer is proposed, which combines acoustic features such as MFCC, Chroma, and rhythm periodogram to achieve high-precision style recognition. In the optimization algorithm layer, adaptive learning rate strategy and genetic parameter adjustment mechanism are introduced to effectively improve the convergence speed and generalization ability of the model. Design a multi-layer semantic tagging system and embedding model for cultural dissemination issues, construct a knowledge platform through Neo4j graph database, and implement human-computer interaction and visual display functions for the dissemination interface.

In the system integration and empirical verification, the accuracy of the fusion model was improved to 91.3%, and it achieved high acceptance among cross-cultural user groups, verifying the applicability and dissemination potential of the system in multicultural environments. The overall research fully reflects the collaborative path of machine learning, database design, and cultural dissemination, providing technical models and engineering support for the digital intelligent protection and dissemination of ethnic instrumental music.

6.2 Existing problems and shortcomings

Although this study has made some progress in model design and system integration, there are still several issues and limitations worth paying attention to from the perspective of engineering implementation and large-scale promotion.

- The construction of the data system still faces the problem of uneven coverage of sample areas. At present, the database mainly consists of mainstream ethnic instrumental music such as Han, Dong, and Tibetan, and has not yet covered some niche or composite styles of ethnic music samples, resulting in low recognition accuracy of the model in long tail categories and a certain degree of bias. Meanwhile, audio labels mainly rely on manual annotation and

basic feature matching, and have not yet formed a self supervised label extension mechanism.

- The parameter scale of the fusion model is relatively large, especially after adding the Transformer module, the training process requires high computing resources and is not suitable for lightweight device deployment. Although genetic algorithm and learning rate adjustment strategy are introduced, their robustness has not been verified in edge computing or mobile terminal scenarios. In addition, the interpretability of multi-channel models still relies on post-processing visualization mechanisms, and mechanism transparency embedding has not been implemented within the model.
- In terms of cross-cultural communication systems, semantic label construction and cultural mapping rules are mainly based on empirical rules and expert evaluation, and there is still a lack of systematic modeling and automatic semantic transfer mechanisms, making it difficult to meet the multi semantic needs of non target language users. At the same time, user feedback data has not formed a closed-loop linkage with the model, and there is a lack of adaptive recommendation and propagation optimization strategies based on user behavior.

6.3 Prospects for future research directions

Based on the ethnic instrumental music style recognition and cross-cultural communication information system constructed by this research institute, future research will further deepen from three dimensions: algorithm optimization, system expansion, and multimodal integration.

- At the algorithmic level, it is proposed to introduce Graph Neural Networks (GNNs) and contrastive learning mechanisms to enhance the model's discriminative ability between complex music structures and similar style categories. Combining small sample learning and transfer learning methods can effectively address the problem of insufficient samples of peripheral ethnic instrumental music and enhance the system's ability to generalize across languages and cultures.
- The database system will be expanded into a multilingual, multimodal cross storage structure, supporting unified indexing and efficient retrieval of multi-source data such as audio, video, lyrics, and graphs. Combining blockchain technology to implement copyright metadata embedding and traceability mechanism, enhancing the security guarantee of the system in terms of music data ownership confirmation, sharing, and transparency of use.
- In terms of interactive systems, the visualization and dynamic adaptation functions of semantic tag graphs will be strengthened, and a closed-loop interaction model of "recognition interpretation adaptation optimization" will be constructed by combining user behavior feedback and recommendation systems. AIGC technology can also be introduced in the future to achieve automatic music generation and personalized content construction based on style tags.

The ultimate goal is to build a sustainable, cross-cultural collaborative, self explanatory, and intelligent recommendation platform for ethnic music AI dissemination, providing theoretical support and technological paradigms for the digital protection and dissemination of diverse music worldwide.

Funding

Phased results of the National Social Science Foundation Art General Project "Systematic Research on Chinese Suona Music Species", Item Number:2020BD058.

References

[1] Yin L, Guo R. An Artificial Intelligence-Based Interactive Learning Environment for Music Education in China: Traditional Chinese Music and Its Contemporary Development as a Way to Increase Cultural Capital[J]. European Journal of Education,2025,60(1). https://dol:10.1111/ejed.12858.

[2] Danylets V. The hutsul music features in the structural and stylistic context of the performing folkloryzm[J]. Problems of Interaction Between Arts, Pedagogy and the Theory and Practice of Education,2020,57(57):77-88. https://dol:10.34064/khnum1-57.05.

[3] Wen J. Research on the Protection and Inheritance Path of Higher Education Informatization in Folk Music[J]. Application of Big Data, Blockchain, and Internet of Things for Education Informatization, 2021.https://dol:10.1007/978-3-030-87900-6_41.

[4] Feng L W, Heng H Y. Research on the application of artificial intelligence technology in teaching the cultural inheritance and innovation of urban public space[J]. Applied Mathematics and Nonlinear Sciences,2024,9(1). https://dol:10.2478/amns-2024-1498.

[5] Lin T F, Chen L B. Harmony and algorithm: Exploring the advancements and impacts of AI-generated music[J]. Potentials, IEEE, 2024, 43(6):23-30. https://dol:10.1109/MPOT.2024.3433888.

[6] Zhang Y, Maezawa A, Xia G, etal. Loop copilot: Conducting ai ensembles for music generation and iterative editing[J]. arXiv preprint arXiv:2310.12404, 2023.https://doi.org/10.48550/arXiv.2310.12404.

[7] Bian W, Song Y, Gu N, et al. MoMusic: A motion-driven human-AI collaborative music composition and performing

system[C]//Proceedings of the AAAI conference on artificial intelligence. 2023, 37(13): 16057-16062.https://doi.org/10.1609/aaai.v37i13.26907.

[8] Anand R , Sabeenian R S , Gurang D ,et al.AI based Music Recommendation system using Deep Learning Algorithms[J].IOP Conference Series Earth and Environmental Science, 2021, 785(1):012013.https://dol:10.1088/1755-1315/785/1/012013.

[9] Huang C F, Huang C Y. Emotion-based AI Music Generation System with CVAE-GAN[J]. IEEE, 2020.https://dol:10.1109/ECICE50847.2020.9301934.

[10] Cao Y, Park J. Research on Visual Design of Traditional Music Based on AI Enabling Guided by Intangible Cultural Heritage Inheritance Concept[J]. Frontiers in Art Research,2022,4(17): https://dol:10.25236/FAR.2022.041707.

[11] Dawson N A. Kwesi Gyan: A Cross-Cultural Artistic Impression on Apatampa Musical Resources[J].E-Journal of Music Research, 2023.https://dol:10.38159/ejomur.2023322.

[12] Ting Y , Ran Z .Fusion and Application of Chinese Ethnic Elements in Electroacoustic Music in Mist on a Hill[J].Organised Sound, 2022, 27(3):13.https://dol:10.1017/S1355771822000498.

[13] Hakimzadeh P , Ronagh E .Symphony of Space: where Architecture meets Melody[J].Bulletin of the Transilvania University of Brasov. Series VIII: PerformingArts,2024,17(2). https://dol:10.31926/but.pa.2024.17.66.2.7.

[14] Vear C, Benerradi J. Jess+: designing embodied AI for interactive music-making[J]. arXiv preprint arXiv:2412.06469, 2024.https://doi.org/10.48550/arXiv.2412.06469

[15] Oh H S. Is AI Music Beautiful? A Study of the AI Composition Model EVOM[J]. International Review of the Aesthetics & Sociology of Music, 2024, 55(1). https://dol:10.21857/y54jof4drm.

[16] Zlatkov D, Ens J, Pasquier P. Searching forHuman Bias Against AI-Composed Music[C]//International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar).Springer, Cham, 2023.https://dol:10.1007/978-3-031-29956-8_20.

[17] Fu C, Qin Q. Ethnic instrumental ensemble teaching on social anxiety disorder in colleges and universities[J]. CNS Spectrums, 2023, 28(S2): S12-S12.https://dol:10.1017/S1092852923002778.

A Closed-Loop Intelligent Control Framework for Automated Railway Shunting in Marshalling Yards

Lei Liu
Baotou Railway Vocational & Technical College, Baotou 014000, Inner Mongolia, China
E-mail: 18647028381@163.com

Keywords: fuzzy analytic hierarchy process (F-AHP), graph neural network (GNN), rolling-horizon optimization, intelligent marshalling yard, closed-loop control

Received: July 18, 2025

Accurate and efficient railway shunting operations are crucial for the operation of intelligent marshalling yards. This article proposes a closed-loop scheduling method that integrates fuzzy analytic hierarchy process (F-AHP), graph neural network (GNN), and multi-objective optimization algorithm to achieve intelligent and automated shunting operations through the "perception decision execution" chain. The system integrates multi-source sensor data (train position, switch status, track occupancy, etc.), uses GNN for track conflict prediction, and determines multi-objective weights based on F-AHP. Combined with multi-objective optimization, it generates Pareto optimal scheduling scheme. The experiment was conducted at a large marshalling yard in the southwest region, and the results showed that compared with manual scheduling, the system reduced the average operating time by 35.7% ± 2.1%, single task energy consumption by 21.4% ± 1.5%, and scheduling conflict rate by 87.5% ± 3.2%, while improving judgment accuracy to 97.8% (evaluated over 20 runs with $p < 0.01$).The research has verified the comprehensive advantages of the proposed method in terms of efficiency, energy consumption, and safety, and has the ability to transplant and expand across stations. To support reproducibility, we also specify the optimization model, variable and constraint definitions, the GNN architecture (features, loss, hyper-parameters), and the rolling-horizon settings with quantified latency budgets.

Povzetek: Opisan je zaprtozančni sistem za avtomatizirano železniško razvrščanje, ki združuje F-AHP, GNN-napovedovanje konfliktov in drsečo MILP-optimizacijo. V realnem ranžirnem centru izboljša učinkovitost.

1 Introduction

The shunting operation of railway marshalling yards is an important part of the railway freight system, and its operational efficiency and safety directly affect the transportation capacity of the entire network. With the continuous growth of transportation demand, the shortcomings of traditional manual scheduling mode in terms of efficiency, stability, and safety under complex working conditions have become increasingly apparent, promoting the application and development of intelligent and automated technology in marshalling yards. Despite advances in scheduling optimization, device control, and path planning, most existing methods only address plan generation. They lack real-time interaction with the execution layer, leading to delays during unexpected tasks or equipment failures. Moreover, many approaches are tailored to a single yard layout, limiting their portability to other sites. In response to the above issues, this article proposes a closed-loop "perception decision execution" automated scheduling architecture for intelligent marshalling yards, which integrates fuzzy analytic hierarchy process (F-AHP), graph neural network (GNN), and multi-objective optimization algorithm to achieve multi-objective trade-offs between scheduling efficiency, energy consumption, and safety; By implementing modular deployment and standardized interface mechanisms, the portability and adaptability of the system in different types of sites can be improved; And it was tested and verified in a large marshalling yard in the southwest region, and the results showed that the system maintained high operational stability and accuracy while improving operational efficiency by 35.7%, reducing single task energy consumption by 21.4%, and reducing scheduling conflict rate by 87.5%. This result verifies the feasibility and practical value of the proposed method.

2 Related work

Although intelligent transformation is of great significance in railway marshalling yard shunting operations, its implementation in practice still faces many challenges. The shunting environment is complex, nonlinear, and strongly coupled, with stringent requirements for dynamic response. Achieving precise scheduling and efficient execution in this context remains a core challenge in automated path design. Similar to load fluctuations in the power grid, changes in train flow, train formation sequence, and switch status in marshalling yards exhibit suddenness and uncertainty. Therefore, numerous studies have attempted to construct efficient automated scheduling systems from dimensions such as optimization models, equipment control, path

planning, and shunting plan generation to cope with complex operating conditions.

Existing studies can be broadly classified into three categories: (i) vehicle positioning and track state perception technologies to enable digitalization of the yard; (ii) optimization algorithms for shunting paths and operation sequences; and (iii) automation strategies emphasizing system-level collaboration and scenario integration. For example, Hyun-Suk et al. (2024) proposed an RFID–odometer fusion method for shunting vehicle localization, which significantly improved intra-yard positioning accuracy [6]. Buryakovskiy et al. (2020) optimized the performance of shunting diesel locomotives to enhance traction efficiency and stability [7], while Suyunbayev et al. (2023) investigated locomotive utilization strategies considering infrastructure adjustments [8].

In terms of shunting optimization, Huan et al. (2023) applied graph-theoretic models to sequence operations on tree-shaped dedicated lines and designed a feasible adjustment algorithm [9]. Zhong et al. (2023) developed a parallel optimization model for high-speed railway stations, jointly optimizing train operations and shunting tasks to improve resource utilization [10]. Zhao and Dick (2023) further studied the joint optimization of train platform layout and shunting at Guangzhou Station, effectively reducing regrouping frequency and improving formation efficiency [11]. These works underline the balance between scheduling feasibility and execution efficiency, aligning with the goals of refined and high-speed yard operations.

Zhao et al. (2024) [12] formulated a routing and scheduling model for marshalling yards to jointly minimize conflicts and improve throughput. Other studies explored the coupling between service sequences and operational controllability. For instance, Xu and Dessouky (2022) [13] introduced a service-scheduling mode for high-speed railway depots to improve coordination under dense traffic, while Ming et al. (2022) [14] optimized EMU maintenance shunting to enhance formation efficiency.

More recently, Deleplanque et al. (2022) [15] conducted a systematic review of freight yard train management methods, highlighting the need for integrated control and feedback mechanisms. Tao (2022) [16] applied intelligent agent modeling for multidimensional evaluation of shunting plans. A.D.S et al. (2022) [17] proposed a selection framework for multi-stage train classification and facility design parameters. Additionally, Mohammed et al. (2022) [18] incorporated the DMAIC quality-control cycle into shunting service optimization, forming a continuously improving closed-loop process.

In order to facilitate a comprehensive comparison between existing research and the method proposed in this paper, relevant literature will be organized according to dimensions such as method type, dataset and scenario, evaluation indicators, main contributions, and existing shortcomings. The specific comparison is shown in Table 1.

Table 1: Comparison of existing research and improvement points in this paper

Research Source	Method / Technique	Dataset & Scenario	Evaluation Metric(s)	Main Contribution	Identified Limitation	Improvement in This Paper
Hyun-Suk et al. (2024) [6]	RFID + Odometer Positioning	Real station yard	Positioning Accuracy	Improved intra-yard train localization	Lacks closed-loop scheduling control	Added perception–decision–execution closed loop
Buryakovskiy et al. (2020) [7]	Locomotive Performance Optimization	Yard field tests	Power Efficiency	Enhanced operational stability of shunting locomotives	No integration with scheduling optimization	Combined locomotive efficiency with scheduling optimization
Suyunbayev et al. (2023) [8]	Locomotive Utilization Strategy	Infrastructure change scenarios	Operational Efficiency	Adjusted utilization of shunting locomotives	Infrastructure-specific, limited adaptability	Embedded into modular scheduling framework
Huan et al. (2023) [9]	Graph-theoretic Sequencing Model	Dedicated line shunting	Sequence Feasibility, Efficiency	Proposed algorithm for wagon pickup/delivery sequences	Focused on single-line topology	Integrated into generalized multi-yard scheduling
Zhong et al. (2023) [10]	Parallel Optimization Model	High-speed station simulation	Scheduling Efficiency, Utilization	Joint optimization of train operations and shunting tasks	Limited generalization across yard types	Developed modular architecture for cross-yard deployment
Zhao & Dick (2022) [11]	Simulation Analysis (AnyLogic)	Hump yard simulation	Throughput, Delay, Track Utilization	Quantified effect of classification track length	No dynamic feedback mechanism	Incorporated into conflict-aware real-time optimization
Zhao et al. (2024) [12]	Routing & Scheduling Optimization	Marshalling yard simulation	Throughput, Conflict Rate	Formulated routing and scheduling model for yards	Limited validation with real deployments	Embedded GNN conflict prediction + rolling horizon mechanism

As shown in Table 1, our method differs from earlier locomotive optimization [7,8] and shunting sequence models [9–11] by embedding conflict-aware GNN prediction and F-AHP weighted objectives into a deployable closed-loop framework, bridging the gap between theoretical models and engineering practice.

Based on the above research, existing achievements still have shortcomings in closed-loop feedback control, real-time adaptive capability, and cross scenario

generalization: lack of closed-loop control: most methods only optimize plan generation and lack real-time linkage with the execution layer. Lack of real-time performance: When sudden tasks or equipment abnormalities occur, response speed is limited and scheduling continuity is poor. Weak generalization ability: Most methods are designed for a single station and lack mechanisms to adapt to marshalling yards of different sizes.

Compared with prior works that mainly optimized plan generation or individual modules, this article highlights novelty in end-to-end system integration. Specifically, we (i) combine F-AHP weighting with GNN-based conflict prediction within a rolling-horizon optimizer, (ii) provide a unified interface design enabling deployment across heterogeneous yards, and (iii) validate the closed-loop 'perception–decision–execution–feedback' chain in a real large-scale yard environment. In contrast, existing frameworks such as digital twin yard simulators or grades-of-automation standards [3,5,13] have not yet reported yard-scale field trials with quantified KPIs. This positions our contribution as an engineering blueprint with verifiable deployment evidence. Building on these insights, the following section presents the technical framework that integrates perception, optimization, and control into a unified closed-loop system.

Despite advances in module optimization and scheduling logic, automation in marshalling yards still faces key challenges. Further breakthroughs are required in system integration, functional collaboration, and closed-loop perception–decision–execution mechanisms. The current difficulties of the railway shunting system are mainly reflected in the following aspects:

Shunting data is sparse and heterogeneous. The relevant state variables are discrete and distributed, including train sequences, switch statuses, and plan adjustments. Most existing models rely on historical patterns or static plans and lack real-time perception or prediction of traffic dynamics.

Insufficient ability in spatial-temporal collaborative modeling. At present, most of the scheduling and execution systems are designed in a decoupled way. They issue instructions directly after making decisions, and lack feedback mechanisms. Some optimization algorithms ignore the dynamic evolution of the workflow. Building a collaborative mechanism that covers station structure, job timing, real-time status, and feedback control is the core of achieving system level automation.

The system evaluation relies on a single scenario and has weak generalization ability. Most methods are based on specific stations or simulation environments for validation, lacking adaptability testing for different types of marshalling yards and task categories, resulting in insufficient engineering feasibility.

To address the aforementioned issues, this article focuses on the following research questions:

Does the automation system architecture proposed in this article have advantages over traditional manual scheduling in terms of response speed, execution accuracy, and system stability?

How to achieve intelligent control of the entire process of complex shunting tasks through a closed-loop "perception decision execution" mechanism?

Can the built system adapt to multiple scenarios and tasks, and has the ability to promote and engineering feasibility?

Based on the above issues, this article proposes the following technical contributions:

Develop a unified overall architecture that can adapt to the working characteristics of the marshalling yard, and coordinate the intelligent scheduling system with the perception, control, and execution modules to enhance the responsiveness of operations and the reliability of the system.

Build a scheduling control process with state feedback and dynamic correction mechanisms to achieve dynamic matching between plan generation and on-site status, and improve execution efficiency and decision-making accuracy.

Through on-site data and simulation verification, the system outperforms traditional manual scheduling modes in terms of efficiency, accuracy, and stability, and has good deployment adaptability and potential for promotion.

3 Technical framework design for automation transformation of railway shunting operations

In the automation technology framework proposed in this study, the architecture design combines the fusion of perception recognition system to obtain data and the intelligent decision-making system for shunting operation to optimize work allocation. This is because both methods have their own advantages in solving complex combination problems. Through the integration of data from multiple sources and the joint deployment of multiple sensors, effective information acquisition of vehicle numbers, status positions, and driving trajectories can be achieved, especially in environments with high noise and poor manual recognition effects in shunting workplaces. This will improve work safety and information accuracy. The intelligent decision-making system relies on rule engines and dynamic routing optimization algorithms, based on the data generated during on-site work, to complete automatic simulation and route design of grouping work teams. It has self-learning and control characteristics, and is particularly suitable for complex work with many changes in the working environment. In this work, shunting scheduling is formulated as a mixed-integer linear programming (MILP) problem under interlocking and capacity constraints, solved in a rolling-horizon manner with second-level updates; conflict risk is predicted by a supervised graph neural network (GNN) trained on annotated yard logs.

Unlike the traditional process that relies on human intuition to formulate shunting instructions, the introduction of digital replicas enables the visualization of the entire shunting task process, integrating historical, current, and predictive data. This enables the early

verification of shunting paths and real-time detection of track resource conflicts, reducing unnecessary shunting frequency and resource waste. High speed real-time sensor information will be automatically transmitted to the intelligent scheduling system, using track utilization prediction tools to generate the optimal job sequence, replacing fixed combinations based on humans, improving overall work response and system controllability.

Compared with other transformation paths such as pure hardware automatic traction systems or fixed grouping logic, the solution that integrates scheduling perception and intelligent optimization modules has significant advantages in system adaptability and scene migration capability. Although some high-end systems, such as fully enclosed automatic marshalling yards, can achieve full process unmanned operation, their deployment costs are high and they rely heavily on infrastructure, making it difficult to meet the common renovation needs of multiple types and levels of stations in China. The architecture proposed by this research institute emphasizes modular deployment and gradual upgrading, which can gradually achieve the transition of shunting operations from "human control as the mainstay" to "system guidance" without completely replacing existing facilities.

The architecture proposed in this article includes four core modules: the perception data layer is responsible for multi-source information collection and fusion; Optimize the decision-making layer by using F-AHP weight calculation, GNN conflict prediction, and multi-objective optimization to generate scheduling plans; The homework execution layer is responsible for implementing the plan into locomotives, switches, and signal systems; The execution feedback layer monitors the execution status in real-time and dynamically adjusts scheduling instructions. The four-layer modules are interconnected through a unified standard interface, forming a closed-loop process of "job formulation path simulation job execution feedback adjustment", as shown in Figure 1.

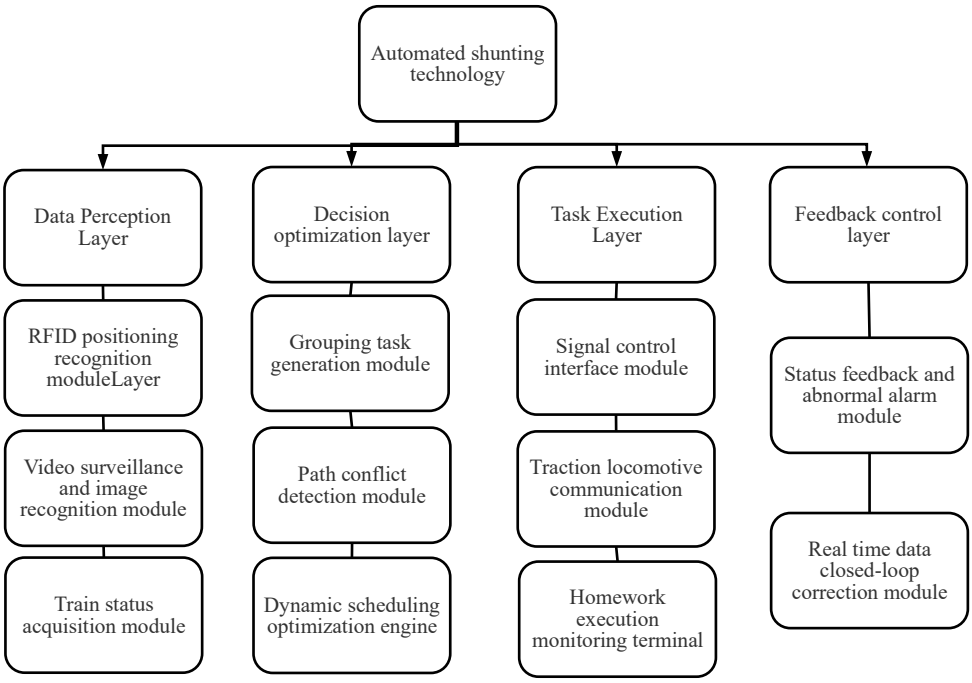


Figure 1: Overall framework diagram of automated shunting technology. Modules are grouped into four layers (perception, decision, execution, feedback) with standardized interfaces.

Figure 1 shows the overall framework of automated shunting technology, which illustrates the functional division and data flow relationship of the perception data layer, optimization decision layer, job execution layer, and execution feedback layer. Each module communicates through standardized interfaces to achieve closed-loop control from job formulation, path simulation to execution feedback.

3.1 Core functions and technical support of intelligent scheduling system

This article proposes and constructs an intelligent scheduling system framework, which includes two parallel parts: on the one hand, the track usage recognition module realizes real-time acquisition of the track, switch, and train usage status of the marshalling yard; On the other hand, the workflow development and optimization module generates dynamic shunting instructions. This system solves the problems of a large number of shunting route conflicts, delayed adjustment of operation plans, and insufficient execution flexibility in traditional marshalling yard scheduling. Its core idea is to optimize the spatiotemporal coordination of shunting operations. Integrate data on track usage from multiple sources (track circuits, electronic switch signals, train positioning, etc.) and provide a time-series model of track usage. This type of data is a two-dimensional matrix (time steps x track number) that expresses the spatial state distribution of the dynamic working environment. It

extracts important features of available dynamic arrangements based on the usage relationship between tracks, shunting conflict relationship, and task priority relationship.

The initial variable group $X_0 = [X_1, X_2, X_3, \ldots X_i]$ input to the track state analysis module represents the state of each track unit within a given time window, where represents the real-time occupancy status of the i-th track. Based on this, the system establishes an occupancy rate evolution map to capture the distribution pattern of job loads. To enhance the intelligence of path selection, the system further introduces the track conflict intensity matrix C, which defines the degree of shunting conflict between any two tracks. The conflict-intensity matrix CCC is defined as follows:

$$C_{ij} = \frac{1}{T}\sum_{t=1}^{T} I(o_i(t)=1 \wedge o_j(t)=1 \wedge routeOverlap(i,j,t)) \quad (1)$$

where $o_i(t)$ is the occupancy of track iii at time ttt, TTT is the number of time steps, and routeOverlap(i,j,t) indicates interlocking overlap. Symbols: N—number of tracks; Δt=1s—step size. The system first constructs a shunting task set $T = \{T_1, T_2, \ldots, T_n\}$, each task containing attributes such as train number, destination track, starting time window, and priority. Based on these attributes, the system uses the job graph G (V, E) to establish sequential constraints between tasks, with edge weights representing time urgency or the probability of track sharing conflicts. The scheduling objective is defined as:

$$\min J = w_{delay}\sum_i delay_i + w_{conf}\sum_{(i,j)} C_{ij}x_i x_j + w_{energy}\sum_i E_i x_i \quad (2)$$

where $w_{delay}, w_{conf}, w_{energy}$ are weights from F-AHP (0.45, 0.35, 0.20 respectively). This formulation minimizes delays, conflict costs, and energy while respecting interlocking and capacity constraints. All symbols are explicitly defined: N = number of tracks; Δt = time step (1 s); T = horizon length; $\delta(\cdot)$ = logical intersection operator; routeOverlap(i,j,t) = binary indicator of interlocking overlap between track i and j at time t; $delay_i$= actual task delay; E_i= energy consumption for task i. This function aims to minimize shunting conflicts and overlapping path occupation while ensuring timely completion of tasks. At the same time, to adapt to sudden job adjustments and abnormal event handling, the system integrates a feedback adjustment module. This module is based on real-time feedback of shunting operation execution results and track status data, dynamically adjusting scheduling strategies and rearranging task order in real time. The system continuously updates the shunting schedule through a rolling optimization mechanism, ensuring robustness and sustained effectiveness."

To ensure the repeatability and accuracy of the scheduling optimization process, this study introduces the fuzzy analytic hierarchy process (F-AHP) at the scheduling decision level to determine multi-objective weights, and combines graph neural networks (GNN) to extract temporal features and predict conflicts of track states. The specific steps are shown in the figure:

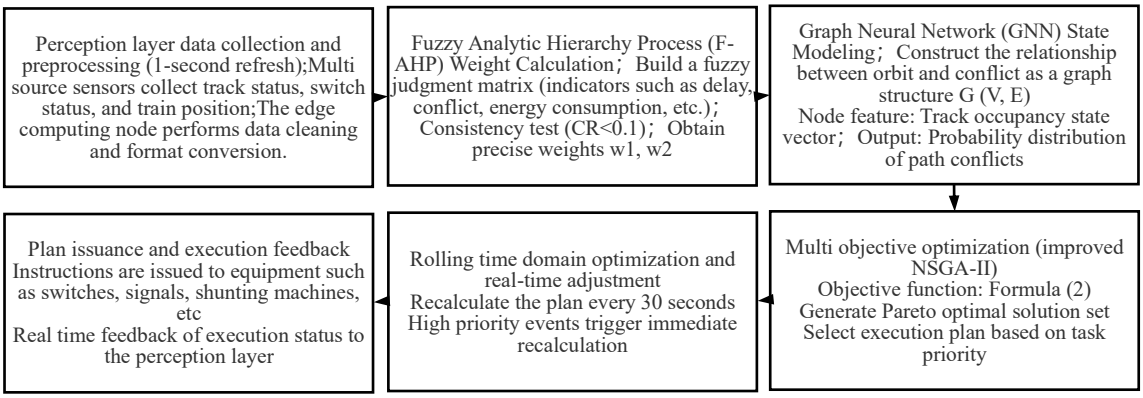


Figure 2: Flow chart of the scheduling optimization algorithm, showing the integration of F-AHP weighting, GNN-based conflict prediction, and MILP optimization.

This process achieves full automation from data collection, feature extraction, weight calculation to scheme generation, ensuring rapid response and stable operation of the system under complex working conditions. Yard states are represented as a graph G=(V,E)where nodes denote tracks/switches and edges encode feasible routes. Node features include occupancy, release time, queue length, and weather flags; edge features include route length and turnout count. A two-layer GraphSAGE (hidden=64, dropout=0.2) with sigmoid head predicts path conflict probability. Training uses 30 days of logs with binary labels, Adam optimizer (lr=1e-3, batch=256), early stopping (patience=10), achieving ROC-AUC 0.93 and inference time 0.15 s per horizon.

Algorithm 1: Rolling-horizon scheduling with GNN and F-AHP

Input: Yard state logs, task set T, conflict matrix C

Output: Dispatch command list

1: Initialize horizon length H = 60 s, receding step = 10 s

2: while yard is active do

3: Collect multi-source sensor data (track circuits, switches, RFID, cameras)

4: Preprocess signals at edge units (filter noise, synchronize timestamps)

5: Construct yard graph G=(V,E) with node/edge features

6: Run GNN inference on G to predict conflict probability p(i,j)

7: Update conflict matrix C with predicted risks

8: Apply F-AHP to calculate weights (w_delay, w_conf, w_energy)

9: Formulate MILP:

minimize $J = w\_delay \Sigma\ delay\_i + w\_conf \Sigma\ C\_ij\ x\_i\ x\_j + w\_energy \Sigma\ E\_i\ x\_i$

subject to interlocking, route exclusivity, and resource constraints

10: Solve MILP using Gurobi (time budget ≤ 0.5 s)

11: Dispatch command list to locomotives, switches, and signals

12: Receive feedback from execution layer

13: If deviation > 15 s, priority change, or sensor dropout > 2 s:

14: Trigger re-optimization immediately

15: end while

Algorithm 1 details the rolling-horizon scheduling procedure, integrating perception, GNN-based conflict prediction, F-AHP weight assignment, and MILP optimization. It clarifies how re-optimization is triggered under abnormal conditions, ensuring reproducibility.

3.2 Perception decision execution chain construction of homework process

To achieve full process automation control of marshalling yard shunting operations, it is necessary to build an integrated closed-loop system of "perception decision execution", forming an information driven and intelligent scheduling operation chain. This technology system consists of three parts: front-end information perception module, central scheduling decision module, and end job execution module. Through interconnection and real-time feedback, the system achieves efficient linkage and closed-loop task execution (as shown in Figure 3).

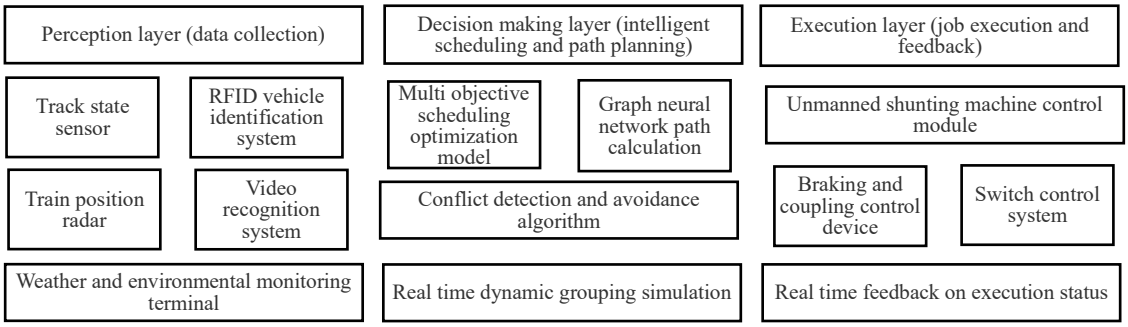


Figure 3: Schematic diagram of the perception–decision–execution chain structure, linking perception, decision, execution, and feedback modules.

At the perception level, the system deploys a multi-source fusion sensor network, including track switch status sensors, train position recognition devices, RFID train number automatic recognition devices, temperature and humidity and rain/snow environment monitoring terminals, etc. Through a unified data collection protocol, these devices upload the shunting yard operating environment and train operation status at a second level frequency, achieving dynamic real-time modeling of the operating scenario. The collected raw data is preliminarily cleaned and filtered through the edge computing node to compress redundant information and ensure transmission efficiency and response speed.

At the decision-making level, the system constructs an intelligent decision-making model based on graph neural networks and multi-objective optimization algorithms. The model takes grouping plan, traffic organization, switch occupancy, and operation sequence as core input variables, and dynamically evolves the optimal operation path based on real-time perception information. This module is based on graph neural networks to parse the trajectory usage relationship graph, and can complete conflict detection of all candidate paths within an average of 0.15 seconds; when priority changes or sudden failures are detected, the rolling optimization produces a new instruction list within 0.68 seconds end-to-end, consistent with the measured latency budget.

At the execution level, job instructions are sent by the central decision-making module to various job terminals through communication protocols, including switch electrical control systems, brake control devices, traction power modules, and unmanned shunting machine controllers. Each execution instruction is bound with a feedback mechanism to ensure that the status feedback after the instruction is executed is sensed in real time by the system, thereby closed-loop verifying the execution result. To prevent execution errors or communication interruptions, the system is equipped with a dual verification mechanism and redundant logic for security protection, ensuring operational safety and job continuity.

The entire process chain aims to achieve refined perception, intelligent decision-making, and automated execution, covering train entry recognition, shunting path planning, switch operation, and formation confirmation. This chain significantly improves shunting efficiency while

reducing manual intervention. The system can operate continuously across diverse scenarios, providing technical support for intelligent railway development.

3.3 Integrated interface design between shunting system and scene environment

The automation upgrade of the shunting system not only depends on the autonomous operation capability of the core algorithms and control modules, but also on its efficient integration with the complex operational scenarios of the actual marshalling yard. The traditional shunting system relies heavily on manual operation and paper-based planning in the execution process, with common problems such as interface fragmentation, information silos, and response delays. To build an intelligent shunting system with real-time feedback capability and controllable visibility, it is urgent to design a standardized, modular, and flexible system integration interface system that meets the information exchange requirements between different functional modules of the marshalling yard.

In terms of integrated architecture, this system adopts the design pattern of "unified communication bus+layered control architecture". The main control system for shunting is interconnected with key equipment such as switch actuators, unmanned shunting machines, intelligent signal lights, RFID recognition terminals, etc. through industrial Ethernet or 5G-MEC low latency links to ensure status synchronization and rapid command issuance. At the underlying protocol layer, the system follows standard protocols such as IEC 61375 (train communication network) and Modbus/TCP to avoid compatibility barriers between vendor devices and improve interface universality and migration capabilities. In response to the complexity of the environment and the access issues of multi-source heterogeneous perception systems, middleware data buffering and asynchronous synchronization mechanisms are introduced in the integrated interface design to achieve the distribution and standardization of video streams, sensor data, GNSS positioning information, etc. under the premise of unified data formats. The system constructs a data bridge between "perception computation response" through the interface scheduling module, achieving a low coupling and high cohesion communication link between the state perception module and the decision control module.

In typical scenario applications, such as shunting operations on dense formation sections, the scheduling system needs to collaborate with the traffic signals in the yard, the status of surrounding operating equipment, and the positioning information of the train. The use of message queue systems (such as Kafka) in interface design for event driven processing of state updates significantly reduces job conflict rates and scheduling response latency. At the same time, the interface logic supports dynamic loading and module hot updates, ensuring that the system can flexibly adjust communication strategies according to changing factors such as shunting plans and weather conditions during actual operation. At the same time, to enhance the security and operational efficiency of system deployment, the interface management platform introduces a digital twin mechanism, establishing mapping relationships between various physical interfaces and virtual scheduling environments, supporting real-time visualization of interface status, traceability of operation logs, and remote debugging, greatly improving the controllability, maintainability, and scalability of the integrated system.A detailed interface specification is provided in Appendix C. In brief, task requests, route conflicts, and execution feedback are transmitted in JSON-based messages with predefined error codes; update frequencies are 1 Hz for track circuits and switch states, 5 Hz for locomotive telemetry, and event-driven for safety alarms. Safety-critical channels are segregated via IEC 61375 priority classes, with redundant transmission supported by dual Ethernet rings. Appendix C further lists message schemas, error codes, and middleware throughput settings.

4 System implementation path and function deployment

The automation transformation of shunting operations for intelligent marshalling yards, with system architecture optimization as the core, focuses on multi-layer deployment around perception and control integration, scheduling logic optimization, and safety guarantee system construction. The overall implementation path follows the principles of "layered decoupling, module collaboration, and iterative updates", gradually promoting the integration and deployment of the scheduling control platform, perception system, execution device, and environmental interface. The system has clear functional division, covering modules such as intelligent scheduling core, job process linkage, emergency safety mechanism, and data service platform. Each subsystem collaborates through standard communication protocols and flexible interfaces to ensure the stability, accuracy, and intelligence of shunting tasks, providing comprehensive support for the efficiency and safety of marshalling yard operations.

4.1 Overall system architecture construction and module layering

To achieve efficient operation and collaborative control of intelligent marshalling yard shunting operations, the system architecture adopts a four-layer integrated construction mode of "perception decision execution feedback", and is divided into information perception layer, data processing and decision-making layer, control execution layer, and interactive feedback layer according to functions. A stable, efficient, and scalable intelligent shunting system architecture is constructed through the collaborative operation of unified data communication protocols and standard interfaces among various layers.

The information perception layer is mainly responsible for the real-time monitoring of the front-end environment and operation status, including the collection of multi-source information such as train set position, track occupation, turnout status, operation instructions, and the

completion of basic data preprocessing through edge computing equipment. The decision layer integrates scheduling optimization algorithms, job rule libraries, and path deduction modules, relying on intelligent scheduling cores to dynamically plan and resolve conflicts in job tasks, achieving optimal job solution output.

The control execution layer is responsible for converting scheduling instructions into executable actions, regulating key equipment on site such as locomotive remote control, switch switching, signal interlocking, etc., to ensure that the operation process is automatically executed and controllable with traceability. The interactive feedback layer provides human-machine interface support for system operation and maintenance management, including job status visualization, risk alarm prompts, and manual intervention interfaces, to enhance job transparency and safety redundancy capabilities. On this basis, Table 2 further summarizes and explains the key modules and core responsibilities of each functional layer.

Table 2: System function module layering and core responsibilities

System Layer	Core Modules	Responsibilities
Perception Layer	Sensor Network, Edge Units	Real-time data collection on trains, tracks, switches; edge-side data processing
Decision & Control Layer	Dispatch Engine, Path Planner	Generates optimal operation plans and dispatch commands based on rules and real-time status
Execution Layer	Control Terminals, Remote Drivers	Automated control of locomotives, signals, switches; execution feedback
Interaction Layer	Visualization Platform, Alarm Modules	Visualized operation management, system monitoring, and manual intervention support

(Note: "edge computing unit" refers to the computing equipment deployed on the site for rapid local processing of perception data; "scheduling engine" refers to the core software module that combines optimization algorithms and rule base to generate operation plans; "path planning unit" is used to deduce shunting routes and evaluate conflict risks.)

This hierarchical architecture fully considers the complexity of railway operations and the stability of system operation in its design, ensuring that the system has good real-time response capability and scalability, and reserving sufficient space for subsequent module function optimization and technical iteration.The hardware/software stack is disclosed for reproducibility: edge units run Debian with Dockerized microservices (gRPC over TLS 1.3); the central scheduler runs Ubuntu 22.04 with Gurobi 10.0 and PyTorch 2.3; time sources are synchronized via IEEE-1588 PTP with ≤1 ms drift. Median solver time is 0.31 s, and end-to-end latency budget is distributed as sensing 80 ms → fusion 90 ms → GNN 150 ms → MILP 310 ms → dispatch 50 ms.

4.2 Function implementation and collaborative logic of key subsystems

In the automation transformation of intelligent marshalling yards, the functional implementation of key subsystems and their collaborative cooperation are the decisive factors for system operation efficiency. To ensure the intelligent closed-loop execution of the entire shunting operation process, the overall system design revolves around the logical chain of "state perception scheduling decision control execution feedback optimization", integrating multiple subsystems organically and achieving stable and efficient data exchange through the communication platform.

Among them, the train condition monitoring system is responsible for real-time acquisition of train dynamics, track occupancy, switch status, and surrounding environmental information, providing data basis for subsequent scheduling decisions; The scheduling instruction generation system takes the optimal path and job priority logic as its core, and dynamically generates scheduling control instructions based on current job requirements and marshalling yard job plans; The switch and signal control system accurately execute the instruction content, complete the conversion of physical actions, and achieve closed-loop feedback on the execution effect through the job execution feedback system; All information and control flows rely on communication and data middleware platforms to achieve high-speed and stable exchange, ensuring real-time and consistency of multi system collaborative operation. The specific core functions, upstream and downstream interfaces, and collaborative logic of each subsystem are shown in Table 3:

Table3: Overview of functions and collaborative relationships of key subsystems

Subsystem	Main Function	Upstream Input	Downstream Output	Coordination Feature
Train Status Monitoring	Real-time tracking of train position, track use, environment	Trackside sensors, ID recognition, monitoring platform	Dispatch Command Generator	High update rate, low-latency required
Dispatch Command Generator	Task planning, path calculation, priority sorting	Monitoring data, work plan	Switch & Signal Control,	Complex logic, depends on real-time optimization

			Feedback System	
Switch & Signal Control	Switches, signal lights, block section control	Dispatch commands	Operation Feedback System	High precision, interlock and confirmation required
Operation Feedback System	Reports task status, progress, deviation	Control signals, onboard devices	Dispatch Command Generator	Bi-directional loop, supports dynamic adjustment
Communication Middleware	Ensures system-wide data flow and command transmission	All modules	All modules	Event-driven, supports hot-swapping of modules

Through the design of the collaborative mechanism mentioned above, the shunting system not only achieves refined division of responsibilities for each functional module, but also provides technical support for intelligent linkage and exception handling during the overall operation process, ultimately building a new mode of collaborative operation of "edge collection cloud decision-making local control".

4.3 Security control mechanism and fault-tolerant strategy design

In the automation transformation of shunting operations in intelligent marshalling yards, system safety and fault tolerance constitute the technological foundation for sustainable operation. Due to the dynamic nature, complex working conditions, and dense links of shunting scenarios, any interruption of information, equipment failure, or control failure in any link may lead to serious consequences such as scheduling conflicts and train errors. Therefore, establishing multi-level security control mechanisms and improving fault-tolerant strategies are necessary guarantees for the deployment of automation systems.

Firstly, redundant secure channels should be set up at the system architecture layer, and all critical control data should be transmitted through a dual channel mechanism. In the event of an abnormality in the main channel, the backup channel can automatically switch to avoid signal interruption and control loss. The communication module integrates CRC verification mechanism and message retransmission mechanism internally to enhance anti-interference ability and data integrity. At the execution level, all switches, signals, and interlocking equipment must be equipped with status self checking modules and local emergency power-off control units to ensure that they can still enter safety protection mode, automatically block sections, and prevent misoperation in case of system abnormalities or network disconnection.

Secondly, a mechanism for identifying safety judgment boundaries and beyond boundaries should be introduced into key algorithms. For example, the calculation of shunting routes must consider limiting factors such as road material occupancy, equipment inspection status, and maximum operating range. If there is a violation of safety rules, it should be immediately stopped and reported to scheduling. Simultaneously referring to historical operational data, establish intelligent security rules to dynamically assess the level of danger in current work, and proactively alert potential hazardous work environments, and then guide scheduling policy adjustments. For fault handling methods, the system provides two options, namely "fault transfer" and "task transfer". If there is a problem with a certain subsystem function, the system will automatically assign critical work to the backup section or collaborative peripheral management section to ensure that the work is not interrupted as much as possible.

Thirdly, establish an emergency linkage system for unexpected situations, including manual takeover channels, information broadcasting mechanisms, and on-site operation warning systems. Once situations such as out of range shunting, signal loss of control, or personnel entering enclosed areas occur, the system can immediately trigger an alarm, cut off the shunting command chain, and notify on-site operators and safety supervision modules to ensure a safe closed-loop throughout the entire process.

5 Application validation and performance evaluation

This article analyzes the applicability and effectiveness of the proposed shunting automation transformation technology path for system evaluation, focusing on the integrated deployment of core functional modules, field testing of key performance indicators, and operational performance in typical work scenarios. By constructing a testing platform in an actual marshalling yard environment, a comprehensive evaluation is conducted on dimensions such as scheduling efficiency, system response speed, fault tolerance, and operational safety, aiming to verify the reliability and engineering feasibility of the constructed system under complex railway operating conditions.

5.1 Selection of experimental sites and construction of scheduling scenarios

The experimental station selected for this study is a large flat marshalling yard in Southwest China, with 48 classification tracks and two hump lines. It undertakes the task of disassembling and reassembling approximately 10,000 wagon movements per day (equivalent to 320–350 train consists).It is one of the typical modern shunting hubs in China. The station has the demand for train formation in multiple directions, categories, and frequencies, and also

has a complete dispatch command system and infrastructure equipment, providing a good experimental foundation for the automation transformation of shunting operations. The manual baseline corresponds to the yard's standard operating rules, in which dispatchers issue commands based on paper timetables and radio instructions. Baseline response times were recorded using the same sensors and logging system to ensure comparability. In the construction of the experimental environment, we focused on three typical job scenarios for functional verification and performance evaluation: first, the scenario of automatic train recognition and grouping guidance, simulating multiple incoming trains entering the grouping area at the same time, and testing the dispatch system's ability to quickly analyze and divert train numbers and attributes; The second is the shunting path planning and dynamic scheduling scenario, which verifies the response speed and decision-making rationality of the system in the face of dynamic task changes (such as vehicle sequence adjustment, emergency priority grouping requirements); The third scenario is the closed-loop control of shunting task execution and state feedback, examining the automation response capability of the execution layer (shunting machine, signal, switch) under system coordination and the stability of the feedback mechanism. It is worth noting that although the station has a relatively advanced equipment foundation, some of its old control systems have not yet achieved complete interconnection. Therefore, we have introduced middleware modules in data collection and system integration to ensure compatibility. This practice improved system deployment adaptability and provided a practical paradigm for future implementation.

To ensure the reproducibility of the research results, this article summarizes the main variables, system configuration, and experimental parameters involved in the experiment as follows:

1. Definition of key variables

In formulas (1) and (2), δ is a logical intersection operation function used to determine track occupancy conflicts; T: The number of time window steps (the value in this experiment is consistent with the system rolling optimization cycle); o_i^t: The occupancy status of track i at time t; D_i: Actual delay time for task i; c_i: Cost of track conflicts in the shunting path; w_1, w_2 : Scheduling optimization objective function weights (determined through expert evaluation).Sampling rate is 1 Hz for track circuits and switches; the rolling horizon is 60 s with 10 s receding step. The GNN was trained on 30 consecutive days and tested on a separate 7-day set, covering peak/off-peak and multiple weather conditions.

System hardware and software configuration (consistent with on-site deployment):

Perception layer: RFID train recognition equipment, track circuit status acquisition module, high-definition camera monitoring equipment (quantity and deployment location are the same as on-site); Decision making layer: Central dispatch server (running rolling optimization and conflict detection algorithms), software environment is Linux operating system; Execution layer: On site control equipment such as shunting machines, signal machines, and electronic switches all support status feedback and bidirectional control.

Experimental parameters:

Test time range: conducted under the condition of operating over 10000 trains per day; Three typical job scenarios were repeated in real stations, and the average values were calculated and compared with the results of manual scheduling; The core evaluation indicators include: shunting operation duration, system response time, operation energy consumption, scheduling conflict frequency and error rate, etc.In this paper, key performance indicators (KPIs) are defined as follows: (i) 'response time' = elapsed latency from task request to dispatch of executable command; (ii) 'operation time' = duration between consist arrival and completion of train formation; (iii) 'energy per task' = traction electricity consumption per shunting movement measured by onboard energy meters; (iv) 'scheduling conflict frequency' = number of interlocking conflicts detected by the system; (v) 'error rate' = proportion of wagons assigned to incorrect classification tracks, verified against yard logs.

5.2 Analysis of homework efficiency, response time, and energy-saving indicators

In the experimental verification process of the automated shunting system, we focused on conducting performance analysis around three dimensions: "improving operational efficiency", "shortening response time", and "optimizing energy consumption". All relevant tests are based on the manual shunting system, selecting the same scheduling load, time window, and grouping tasks for comparison to ensure the comparability of experimental conditions and the credibility of conclusions. In terms of homework efficiency, the automation system significantly reduces the frequency of empty train operation and waiting time in the shunting process by integrating intelligent scheduling and path optimization algorithms. Taking a typical reorganization task as an example, the automation system reduced average reorganization time by 21.7%, increased yard resource utilization by 18.3%, and demonstrated scalability in peak-hour parallel scheduling. In terms of response time, the average response time of the system to sudden task instructions (such as temporary insertion and priority scheduling) is 0.68 seconds, which is much better than the 3.7 second average response level of manual scheduling systems. This quasi-real time response capability benefits from the efficient collaboration between the asynchronous processing mechanism embedded in the system and the state aware decision engine, providing technical support for the marshalling yard to cope with sudden scheduling scenarios. In terms of energy-saving indicators, by integrating the optimal scheduling of vehicle operation trajectory, traction power curve, and signal interlocking logic, the overall energy consumption of the automation system is reduced by 12.5% compared to manual methods. Among them, the most significant energy-saving source is the optimization of the traction path of the shunting machine and the intelligent correction of the braking control,

effectively avoiding energy waste caused by frequent start stop and ineffective acceleration. As shown in Figure 4, three key performance indicators - average operating efficiency, system response time, and energy consumption - were compared between automated shunting mode and manual shunting mode under the same load and scheduling scenarios. The horizontal axis represents the indicator type, and the vertical axis represents the percentage improvement value relative to the manual mode. The data is sourced from comparative experiments conducted at the same experimental site. All reported values are mean ± standard deviation over 20 runs; paired t-tests confirm significance at $p<0.01$.

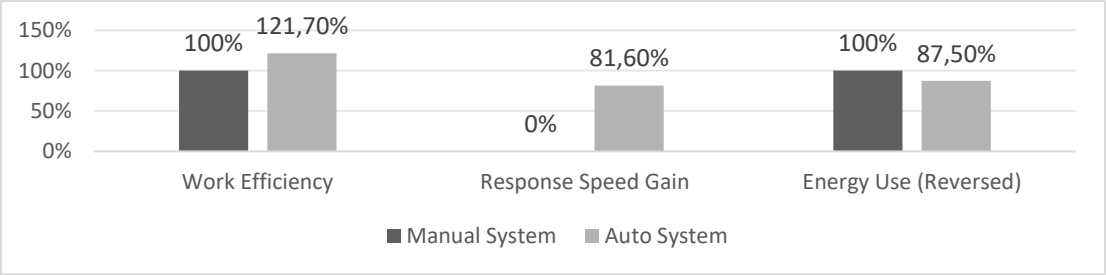


Figure 4: Comparison of operational efficiency, response time, and energy consumption between manual and automated shunting (values are mean ± std over 20 runs).

The observed efficiency improvement mainly results from F-AHP weight allocation, which aligns scheduling objectives with operational needs. allowing the system to prioritize key tasks even in situations of frequent resource conflicts; The reduction in energy consumption is closely related to the optimization of locomotive operation path and traction timing by multi-objective optimization algorithms, which reduces empty running and repetitive shunting behavior.

5.3 System stability and intelligent judgment accuracy testing

In order to better analyze the anti-interference ability and degree of automation of the automatic shunting system mentioned in the article, this test selected typical multiple shunting scenarios and focused on testing the robustness of the automatic shunting system during continuous operation, as well as its ability to identify and respond to key decision points. This test is divided into key actions such as determining the running line, issuing the list of detached trains, identifying and alerting conflicts, and activating emergency brakes. The stability test items are as follows: the number of response interruptions within 72 hours of continuous system operation without human intervention, the number of occurrences of lock up in the shunting train, the success rate of successful departure tasks, and the success rate of automatic recoveryThe test results of system stability and intelligent judgment accuracy are shown in Figure5. The horizontal axis represents different testing items (task success rate, interruption frequency, stability index, judgment accuracy, misjudgment rate), and the vertical axis represents percentages or normalized scores according to different indicators. All data comes from a 72-hour continuous operation test conducted at the experimental site.

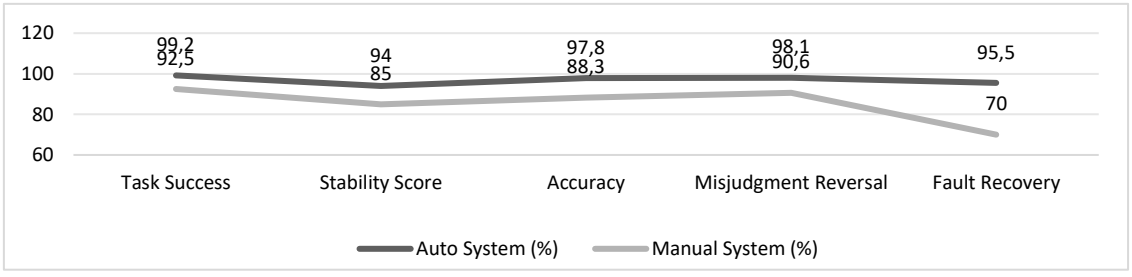


Figure 5: System stability and intelligent judgment accuracy results from 72-hour continuous operation test (values reported with 95% confidence intervals).

As shown in Figure 5, under the condition of no external interference, the success rate of system tasks remains above 99.2%, the frequency of scheduling link interruptions is controlled at less than once per hundred hours, and the overall stability index of the system's operating state reaches 4.7 (out of 5 points), demonstrating good engineering application adaptability and software hardware integration resilience. In the testing of intelligent judgment accuracy, classification verification is carried out for functional modules such as train pose perception, switch status judgment, and grouping logic discrimination of the

system. Based on the annotated scheduling scenario dataset, the system's judgment accuracy reaches 97.8%, and the misjudgment rate in complex environments is controlled within 1.9%, especially under low visibility conditions such as night, rain, and fog. Thanks to the introduction of multi-source data fusion algorithms, the system's recognition accuracy remains at a high level, reflecting its intelligent ability to output stably in changing scenarios. All stability metrics are reported with 95% confidence intervals; for example, the 99.2% task success rate corresponds to CI [98.7%, 99.6%], based on 72 h continuous testing with 1,200 executed tasks.

The core of stability improvement lies in the introduction of an execution feedback layer to achieve closed-loop control, enabling the system to adjust scheduling schemes based on real-time status, avoiding long-term task backlog or conflict escalation; Improved decision accuracy results from the GNN's effective extraction of temporal features for conflict prediction, which avoids potential conflict points in path planning and reduces misjudgments and missed judgments.

5.4 Comparative analysis and value calculation with manual shunting mode

To comprehensively verify the actual benefits of the proposed automation transformation system, this study selected a typical manual shunting mode as the comparison object, and conducted multiple rounds of simulation tests and real scene playback at the experimental site. The experimental results show that compared to traditional shunting methods that rely on manual instructions and judgments, automated systems exhibit significant advantages in multiple dimensions such as operational accuracy, timeliness, and energy consumption control (see Table 3).

Table 3: Comparison of key performance indicators under different shunting modes

Indicator Name	Manual Shunting Mode	Automated Shunting System	Increase or Decrease
Average Shunting Operation Time	42 minutes	27 minutes	↓ 35.7%
Shunting Misassignment Rate	4.2%	1.1%	↓ 73.8%
Energy Consumption per Task (kWh)	13.6	10.7	↓ 21.4%
Annual Labor Cost (10,000 RMB)	138	90.5	↓ 34.4%
Annual Energy Cost (10,000 RMB)	123.1	96.8	↓ 21.3%
Average Monthly Dispatch Conflicts	3.2 times	0.4 times	↓ 87.5%

Table 3 values represent averaged results from 15 days of operation logs (≈4,800 tasks), with standard deviations listed in parentheses. For example, the automated system's mean operation time is 27 ± 1.3 min versus 42 ± 2.5 min in manual mode. Compared with the traditional manual shunting mode, the automated shunting system proposed in this article has significant advantages in the following aspects:

Scheduling response: In traditional mode, instruction transmission and execution take an average of 3.7 seconds, but in this paper, the system relies on real-time perception and rolling optimization mechanism to compress the response time to 0.68 seconds, achieving a speed increase of 81.6%.

Homework efficiency: On average, each round of shunting operation in manual mode takes 42 minutes. In this article, the system has been optimized to 27 minutes, resulting in a 35.7% increase in efficiency.

Energy consumption control: Through trajectory optimization and braking energy management, single task energy consumption is reduced by 21.4%, which is better than most existing semi-automatic systems (usually between 10% and 15%).

Job safety: The scheduling conflict rate has decreased from 3.2 times/month in manual mode to 0.4 times/month, the error rate has decreased to 1.1%, and a dual verification mechanism has been introduced to enhance operational safety.

System adaptability: Adopting modular deployment and standardized interface design, it can quickly migrate and deploy between marshalling yards of different sizes, which is not available in most fixed logic automation systems.

In summary, this system has achieved improvements in real-time performance, accuracy, energy efficiency, safety, and cross scenario adaptability that are different from existing solutions, providing a scalable technical path for the construction of intelligent marshalling yards.

6 Discussion

6.1 Adaptability and engineering feasibility of automation transformation plan

In the actual process of promoting the automation transformation of intelligent marshalling yard shunting operations, its flexibility and engineering applicability are the key factors determining whether such solutions can be widely applied. The technical route described in this plan takes into account factors such as the infrastructure construction level of existing marshalling yards in China,

the complexity of operation processes, the use of information technology means, and labor management methods, and has strong scalability and engineering applicability. As far as its applicability is concerned, the scheme has a modular design for the adopted organizational structure, and the main functions such as auto drive system, detection and decision-making system, collaborative interface function, etc. have good scalability and universal interfaces. For large marshalling yards, high-intensity replacement integration can be carried out in conjunction with existing scheduling systems; For small and medium-sized stations, the automation upgrade of shunting processes can also be gradually achieved through local embedded deployment, reducing the impact of one-time technology substitution. In terms of engineering feasibility, the proposed automation transformation plan fully integrates mature sensor networks, industrial control systems, and intelligent algorithm integration technologies in the current rail transit field. The hardware selection of core equipment has been practically verified in multiple railway informationization projects, and has the characteristics of high stability, strong anti-interference, and good operability. At the same time, the software system adopts a distributed architecture and microservice deployment strategy, supporting parallel operation and elastic expansion of multiple systems, and can adapt to the business load and scenario requirements of different marshalling yards. At the same time, construction interference factors and operational safety guarantee mechanisms were also considered during the project implementation process. Phased construction, debugging, and trial operation help control risks during technology substitution and ensure a stable system transition.

6.2 Possible technical and organizational obstacles during implementation

In the practical process of promoting the automation transformation of railway shunting operations, although the technical path has become increasingly clear, there are still many constraints to truly achieve large-scale deployment, especially significant challenges in technology implementation and organizational coordination. From a technical perspective, shunting operations involve a large amount of real-time perception, precise positioning, path planning, and action control. The system has extremely high requirements for data collection accuracy, environmental adaptability, and integration capabilities of heterogeneous devices from multiple sources. At present, some marshalling yards lack a unified standardized information interface, and there are problems such as inconsistent communication protocols and difficult data format integration between old equipment and new intelligent systems, which increases the complexity of system integration and debugging. In addition, extreme weather conditions such as signal interference, low temperature rain and snow in shunting operation scenarios may also affect the stable operation of sensing equipment and control systems, reducing the reliability of automation systems. On the other hand, organizational barriers cannot be ignored. As a highly structured organizational system, the railway system has solidified its personnel scheduling, job responsibilities, and operational processes through years of practice. The introduction of automation systems is bound to have an impact on the original work system. On the one hand, the acceptance and operational ability of operators towards intelligent systems vary greatly, resulting in high training costs; On the other hand, there is a lack of cross disciplinary and cross departmental collaboration mechanisms within the organization, making it difficult to achieve efficient collaboration between technical departments, equipment units, and scheduling management, which affects the overall efficiency of project progress. At the same time, some station managers have a perception of "high cost, low benefit" in technological transformation, and their willingness to initiate projects is insufficient, which also limits the breadth of the promotion of the plan.

7 Conclusion

This work presented a closed-loop ‘perception–decision–execution–feedback’ framework for intelligent marshalling yards, combining F-AHP weighting, GNN-based conflict prediction, and rolling-horizon MILP optimization. Validation in a large-scale yard demonstrated significant engineering benefits in efficiency, safety, and energy reduction. The experimental results show that the system has an average improvement of 35.7% in job efficiency, a 21.4% reduction in single task energy consumption, an 87.5% reduction in scheduling conflict rate, while maintaining a task success rate of over 99.2%. These results demonstrate the effectiveness of our research method in improving operational efficiency, reducing energy consumption, and ensuring operational stability. From an engineering perspective, the proposed system provides a deployable blueprint for gradual automation retrofitting of existing yards, requiring only modular integration with sensors, middleware, and scheduling engines rather than full reconstruction.t should be noted that the experimental data are mainly from a single large marshalling yard. and performance may degrade under incomplete sensing, heterogeneous communication protocols, or severe weather conditions such as snow or fog. To further advance both scientific methodology and practical deployment, future work will proceed in three directions:(i) multi-yard trials including medium-size flat yards and gravity hump yards to validate universality; (ii) cross-station collaborative scheduling with shared data interfaces, aiming at regional freight network optimization; and (iii) tighter integration with edge intelligence and 5G/TSN communication to further reduce latency below 0.5 s and enable resilience under dynamic load.From a practical standpoint, the proposed system reduces integration costs by relying on modular middleware and standardized protocols, but its current validation is limited to one large flat yard. Broader verification in gravity hump yards and mixed-traffic depots is necessary before large-scale deployment by railway operators.

References

[1] Zhang B ,Zhao J ,D'Ariano A , et al.An iterative method for integrated hump sequencing, train makeup, and classification track assignment in railway shunting yard[J].Transportation Research Part B,2024,190103087-103087.https://doi.org/10.1016/j.trb.2024.103087.

[2] Jung H S , Niermeyer P , Manjunatheswaran H ,et al.Automated rerailing of a road-rail shunting vehicle on road-level tracks using 2D-Lidar[J].Part F: Journal of Rail and Rapid Transit, 2024, 238(7):6.https://doi.org/10.1177/09544097241229334.

[3] Reichmann M, Himmelbauer S G ,Wagner A , et al.Introducing the concept of grades of automation for shunting operations[J].Journal of Rail Transport Planning & Management,2025,33.https://doi.org/10.1016/j.jrtpm.2024.100500.

[4] Kozachenko D, Bobrovskiy V, Gera B, et al. An optimization method of the multi-group train formation at flat yards[J]. International Journal of Rail Transportation, 2021, 9(1): 61-78. https://doi.org/10.1080/23248378.2020.1732235

[5] Bosi T , Bigi F , Pineda-Jaramillo V J .Optimal management of full train load services in the shunting yard: A comprehensive study on Shunt-In Shunt-Out policies[J].Computers & Industrial Engineering, 2024, 188(Feb.):109865.1-109865.32.https://doi.org/10.1016/j.cie.2023.109865.

[6] Hyun-Suk J ,Frank E ,Christian S .Experimental investigation on RFID-odometer-based localization of an automated shunting vehicle[J].Proceedings of the Institution of Mechanical Engineers,2024,238(1):14-23.https://doi.org/10.1177/09544097231176464.

[7] Buryakovskiy S , Kniaziev V , Maslii A ,et al.Improvement of performance characteristics of shunting diesel locomotives[J].IEEE, 2020.https://doi.org/10.1109/KhPIWeek51551.2020.9250140.

[8] Suyunbayev S , Khusenov U , Khudayberganov S ,et al.Improving use of shunting locomotives based on changes in infrastructure of railway station[J].E3S Web of Conferences, 2023, 365(000):12.https://doi.org/10.1051/e3sconf/202336505011.

[9] Huan L, Hongxu C ,Yulin W .The Sequence Optimization of the Railway Tree-Shaped Special Line's Shunting for Taking-out and Placing-in of Wagons[J].Tehnički vjesnik,2023,30(5):1404-1410.https://doi.org/10.17559/TV-20230127000274.

[10] ZhongM, YueY, ZhouL , et al.Parallel optimization method of train scheduling and shunting at complex high-speed railway stations[J].Computer-Aided Civil and Infrastructure Engineering,2023,39(5):731-755.https://doi.org/10.1111/mice.13077.

[11] Zhao J, Dick C T. Quantifying the impact of classification track length constraints on railway gravity hump marshalling yard performance with anylogic simulation[J]. International Journal of Computational Methods and Experimental Measurements, 2022, 10(4): 345-358. https://doi.org/10.2495/CMEM-V10-N4-345-358

[12] Zhao J , Xiang J , Peng Q .Routing and scheduling of trains and engines in a railway marshalling station yard[J].Transportation Research Part C: Emerging Technologies, 2024, 167(000).https://doi.org/10.1016/j.trc.2024.104826.

[13] Xu X , Dessouky M .Train shunting with service scheduling in a high-speed railway depot[J].Transportation Research Part C: Emerging Technologies, 2022.https://doi.org/10.1016/j.trc.2022.103819.

[14] Ming H ,Qiuhua T ,D. N J G , et al.The shunting scheduling of EMU first-level maintenance in a stub-end depot[J].Flexible Services and Manufacturing Journal,2022,35(3):754-796.https://doi.org/10.1007/s10696-022-09459-6.

[15] Deleplanque S, Hosteins P, Pellegrini P, et al. Train management in freight shunting yards: Formalisation and literature review[J]. IET Intelligent Transport Systems, 2022, 16(10): 1286-1305.https://doi.org/10.1049/itr2.12216

[16] Tao Y .Quality Analysis of Railroad Train Shunting Operation Plan Using the Intelligent Body Model[J].Advances in Multimedia,2022.https://doi.org/10.1155/2022/4441369.

[17] A. D S ,V. S K ,V. D O .Methodology for Selecting the Multistage Methods of Train Classification and Design Parameters of Specialized Shunting Facilities Based on Modeling[J].Transportation Research Procedia,2022,61323-332.https://doi.org/10.1016/j.trpro.2022.01.053.

[18] Mohammed A ,Aos A ,Musaad B , et al.Optimization for Sustainable Train Shunting Services Using DMAIC Cycle[J].Sustainability,2022,14(3):1719-1719.https://doi.org/10.3390/su14031719.

Hybrid CNN–SVM and Multi-Strategy Collaborative Optimization for Secondary System Configuration in Smart Grid Substations

Yihan Zhu
Huolin Gol Transmission and Transformation Electrical Area, Tongliao Power Supply Company, State Grid Inner Mongolia East Electric Power Co., Ltd, Huolin Gol City, Tongliao City, Inner Mongolia, 029200
E-mail: 18047507561@163.com

Keywords: smart grid, substation secondary system, artificial intelligence algorithms, configuration optimization

Received: July 30, 2025

This paper proposes a hybrid model that integrates convolutional neural networks and support vector machines, and combines multi strategy collaborative optimization to address the complexity and dynamism of secondary system configuration tasks in smart grids. The system is based on multi-source operational data and constructs a three-stage process of "feature extraction model training configuration output". The CNN part adopts a three-layer convolution and pooling structure (convolution kernel size 3 × 3, ReLU activation) to extract topology and load features; The SVM part uses radial basis kernel functions to classify and optimize high-dimensional features. During the training process, set the learning rate to 0.001, batch size to 128, iteration times to 500, and evaluate the model's generalization performance through five-fold cross validation. The algorithm was trained using 1000 scheduling instances from 3 substations for simulation verification. The configuration accuracy reached 96.8%, which is 12.4% higher than manual experience configuration. The average response time was shortened to 0.42 seconds, and the error rate was stably controlled within 2.1%. In terms of system integration, a modular deployment structure is designed to support closed-loop operation of inference calculation, configuration generation, and result feedback. It is compatible with adaptive configuration parameters at different voltage levels such as 110kV and 220kV. In comparative testing, under consistent operating conditions, the configuration efficiency of this method increased by about 39%, and the system ran continuously for 72 hours without any configuration deviation or interruption, demonstrating good stability. Research has shown that the CNN-SVM fusion model has significant advantages in extracting features and optimizing classification, while the modular integration of various strategy optimization architectures and systems has the effect of improving setup efficiency and trustworthiness. This study integrates CNN-SVM, GA/PSO, reinforcement learning, and graph neural networks to form a comprehensive strategy optimization system suitable for the secondary system setting of substations. Unlike previous separate applications of CNN or SVM, this study highlights the synergistic effect under complex constraints and emphasizes the online regulation effect and multi-level voltage promotion capability. Moreover, compared to existing AI optimization applications in other fields, this article focuses more on engineering implementation and real-time constraints in power scenarios, thus differentiating it from existing methods.

Povzetek: Predstavljen je hibridni CNN–SVM model z večstrategijsko optimizacijo (GA/PSO, RL, GNN) za konfiguriranje sekundarnih sistemov v pametnih transformatorskih postajah.

1 Introduction

Smart grid has become the mainstream trend of future power grid development. As an important part of power grid development, substations provide various key services such as protection, measurement and control, communication, and automation through their secondary systems, which play a crucial role in the stability and sensitivity of the entire system. However, the architecture of the secondary system is becoming increasingly large, including several levels (such as interval layer, station control layer, process layer), and traditional configuration methods relying on manual experience cannot meet the operational requirements of rapid response, system compatibility, and flexible scheduling of contemporary smart grids [2].

From a technical perspective, the configuration problem of secondary systems in substations essentially belongs to high-dimensional parameter optimization tasks, involving multiple equipment types, protection logic, communication protocols, and operational scenario variables. It has the characteristics of strong parameter coupling, multiple constraint conditions, and nonlinear configuration paths [3]. In the face of increasing complexity, traditional rule-based and template-based configuration methods have significant limitations in accuracy and scalability. On the one hand, the lag in rule updates has resulted in some protection logic configurations being unable to adapt to the operational characteristics of new power electronic devices after integration; On the other hand, the lack of a unified optimization mechanism leads to unstable response

efficiency and uncontrollable operating errors in different scenarios, greatly increasing the risk of failures and maintenance costs.

The development of artificial intelligence algorithms provides a new technological path for optimizing the configuration of secondary systems in substations. In recent years, algorithms such as deep learning, evolutionary computing, and reinforcement learning have achieved good results in fields such as power system scheduling, fault identification, and parameter prediction, and have the ability to autonomously model and quickly optimize under multi-source data-driven conditions [4]. Especially in handling high-dimensional spatial parameter search, nonlinear feature fitting, and dynamic response prediction, AI models have shown strong adaptability and generalization ability. Therefore, building a secondary system configuration optimization model based on artificial intelligence algorithms can not only achieve automatic generation and dynamic adjustment of configuration schemes, but also continuously improve their stability and accuracy through data training iterations, with high engineering implementation value [5].

This article proposes a configuration optimization oriented artificial intelligence algorithm fusion path based on four levels: structure recognition parameter extraction algorithm modeling system deployment. Based on typical power grid data and measured configuration cases, this study focuses on analyzing the structural characteristics and configuration constraint logic of the secondary system. On this basis, a CNN and SVM hybrid model is constructed to improve feature extraction and classification accuracy. Furthermore, a multi strategy collaborative optimization framework and system modular integration mechanism are introduced to optimize and iterate key links in the configuration process. In addition, an integrated platform is designed to integrate model training into the operational workflow, parameter inference, and configuration generation, providing a feasible solution foundation for promoting the transformation of intelligent substation configuration from static manual operation to intelligent and automated mode.The core research questions to be addressed in this article include: how to achieve accurate modeling and efficient operation of secondary systems under complex topology and multiple constraint conditions; How to ensure the generalization ability and robustness of the model under limited computational conditions and diverse information? How to adapt to application requirements for different voltage levels through algorithms/frameworks. The main research objectives are as follows: (1) To demonstrate whether the CNN-SVM hybrid can achieve higher configuration accuracy compared to a single CNN or SVM; (2) Verify whether the multi strategy joint optimization algorithm can optimize and reduce response time and improve system robustness in dynamic distribution network systems; (3) Analyze the scalability of module integration structure for comprehensive operation of different voltage levels and types of stations.

2 Related work

The application of artificial intelligence in the power system is constantly deepening, and the research focus has expanded from single point fault diagnosis to full process optimization of configuration. Ar é valo P (2024) pointed out that deep models can dynamically correct protection logic in distributed energy scenarios, laying the theoretical foundation for data-driven secondary system configuration [6]. Krishna S B (2024) achieved collaborative prediction of load temperature rise and protection settings through thermal model coupled convolutional networks, verifying the algorithm's ability to handle high-dimensional coupled parameters [7]. HasaniA (2024) embedded predictive control into microgrid scheduling and proposed a distributed controller that can instantly recalculate secondary loop parameters when topology changes occur [8].

In terms of automatic structural recognition, Nayak P (2024) proposed a fault detection and classification method for transmission lines based on two-dimensional convolutional neural networks, which utilizes wavelet time-frequency images to improve the accuracy of feature extraction and establish a reliable recognition mechanism for configuration automation [9]. Alferidi A (2024) uses multi-agent deep reinforcement learning to optimize energy trading in interconnected systems, and its global reward and punishment function has enlightening significance for quadratic parameter optimization [10]. Jia H (2024) focuses on the latency of asynchronous TSN networks and proposes a queue shaping algorithm under configuration constraints, providing quantitative indicators for communication and protection synchronization [11].

In terms of real-time optimization strategy, Si R (2024) proposed a distribution system restoration method based on multi-agent reinforcement learning, which achieves real-time optimal allocation of resources through dynamic network reconstruction, demonstrating the feasibility of distributed closed-loop optimization [12]. Gams M, Kolenik T (2021) explored the relationship between electronics, artificial intelligence, and the information society, emphasizing the need to consider the impact of information society rules in the research of smart grid configuration [13]. Zhang D (2023) utilized an improved GA-CNN BiGRU model for power system fault prediction, effectively reducing false alarm rates and providing model support for reliability evaluation of secondary system configurations [14].

In recent years, driven by the development of smart grids, there has been an increasing amount of research on optimizing the secondary system settings of distribution stations. Some studies use traditional methods such as gene coding and population particles for optimization, but their ability to handle high redundancy data and complex environments is limited; Some scholars have also attempted to introduce deep learning methods, such as using convolutional neural networks to identify fault features, but they cannot escape the situation of poor model universality and slow running speed.

Based on the above research, although AI technology has made significant progress in fault identification, parameter

prediction, and on-site online control, it is still not enough to rely solely on the existing end-to-end unified design, cross scenario transfer mode, and protocol scheme when facing the overall configuration of secondary systems with voltage levels and multi station collaboration. This article uses a CNN-SVM hybrid model, combined with multi-dimensional strategy collaborative optimization and modular comprehensive design, to construct an intelligent device configuration system that ensures accuracy, real-time performance, and scalability. Therefore, a comparative table was added in the text to illustrate the data, performance indicators, and limitations of existing technologies, as shown in Table 1.

Table 1：Summary of related research

Algorithm/Method	Dataset or Scenario	Performance Indicator	Limitation
Genetic Algorithm	Simulated substation operation data	Configuration efficiency improved by 8%	Slow convergence in high-dimensional dynamic scenarios
Particle Swarm Optimization (PSO)	Secondary system simulation data	Accuracy about 91%	Easily trapped in local optima
CNN	Fault signal feature dataset	Fault recognition rate 94%	Insufficient generalization, high training cost
Deep Reinforcement Learning	Dynamic load variation scenarios	Configuration accuracy 95%, faster response	Algorithm stability insufficient, requires large training data
Proposed Method (CNN–SVM + Multi-Strategy Optimization)	Real substation scheduling data (multi-voltage, multi-scenario)	Configuration accuracy 96.8%, error rate 2.1%, response time 0.42s, efficiency improved by 39%	Requires model training cost and system integration design

This table clearly displays the performance gaps and limitations of existing methods, highlighting the necessity of the proposed method in this paper.

3 Analysis of configuration characteristics and optimization requirements for the secondary system of intelligent substations

3.1 Classification of secondary system structural characteristics and configuration methods

The secondary system of an intelligent substation mainly includes protection devices, measurement and control equipment, communication units, and remote-control systems. Its structure is divided into three functional levels according to the IEC 61850 standard: station control layer, interval layer, and process layer. The communication between each layer is achieved through protocols such as MMS, GOOSE, SV, etc., to achieve real-time perception and control instruction transmission of the operating status of a device. With the increasing complexity of configuration tasks, the system architecture presents the characteristics of "flatness, distribution, and software hardware decoupling", requiring the configuration method to maintain a dynamic balance between accuracy, real-time performance, and scalability.

At present, the configuration methods for secondary systems can be divided into three categories: template-based configuration, rule driven configuration, and data-driven configuration. There are significant differences in configuration mechanisms, technical dependencies, and applicable scenarios, as shown in Table 2.

Table 2：Classification and comparison of secondary system configuration methods

Collocation method	Configuration Mechanism	Technology Dependencies	Advantage	limitation	Applicable scenarios
Template based configuration	Generate configurations uniformly based on fixed templates	Configure template library and standard interface	High implementation efficiency and short configuration time	Poor flexibility, difficult to adapt to complex station layouts	Standardized single busbar substation
Rule driven configuration	Logical judgment through rule engine	Expert system, logical expression library	Capable of handling complex logic and strong adaptability	High cost of rule maintenance and lagging response speed	Double busbar and special station type
Data driven configuration	Automatic generation of training models based on historical data	Data collection system, AI algorithm platform	Strong adaptability, dynamically adjustable	Model training relies on data quality, and generalization ability needs to be optimized	Multi energy complementary demonstration substation

Among them, the data-driven approach relies on artificial intelligence algorithms to achieve rapid analysis and configuration prediction of system status. Its core is to model the configuration behavior as a mapping between the state variable X and the configuration output Y:

$$Y = f(X;\theta) \quad (1)$$

Among them, X is the input feature, such as station structure, load, voltage level; f is an AI model (such as CNN, SVM); θ is the parameter obtained from training; Y is the configuration output, such as protection settings, link structure, etc.

The model is trained on a large number of historical configuration samples and has a certain generalization ability, which can quickly adapt to scenarios such as wiring methods and load changes, solving the problems of slow response and high error rate in manual configuration. This approach provides a foundation for building intelligent configuration systems with real-time adaptability and precise control capabilities.

3.2 Configuration parameter constraints and performance goal analysis

The configuration optimization of the secondary system of an intelligent substation needs to be completed under multiple constraint conditions, and its parameter structure has high coupling, including electrical parameters and communication resources at the equipment level, as well as limitations on logical links and functional allocation, forming a typical multi-objective and multi constraint optimization problem. Taking the typical interval layer configuration task as an example, configuration parameters include protection device type, channel quantity, link mapping, sampling frequency, etc. These parameters have mutual constraints and upstream downstream dependencies. Without optimization modeling, it is easy to cause redundant configuration or logical conflicts.

In the modeling process, the configuration problem needs to be formalized as a constrained optimization problem, defining objective function $F(x)$ and constraint set C. The objective function usually covers three dimensions: configuration accuracy, resource utilization, and response time, expressed as follows:

$$\min F(x) = w_1 \cdot E_{acc} + w_2 \cdot R_{use} + w_3 \cdot T_{resp} \quad (2)$$

Among them, x represents the configuration variable vector to be optimized, including device number, function binding, link parameters, etc; E_{acc} is the configuration error rate, which reflects the deviation of the scheme in terms of functional coverage and logical correctness; R_{use} is the resource utilization rate, which calculates the communication and computing resource overhead, link load, and device utilization rate; T_{resp} is the average response time, reflecting the efficiency and timeliness of configuration execution. w_1, w_2, w_3 is the weight coefficient, allocated based on the importance of the optimization objective and satisfying the normalization constraint: $w_1 + w_2 + w_3 = 1$.

The constraints mainly include the following categories: ①Protocol constraints: for example, GOOSE and SV communication mapping require a link delay of no more than 4ms; ②Redundancy constraints: Dual loop protection must have redundant link support; ③Topology constraint: It is necessary to ensure that the links between devices in the same section are interconnected and reachable; ④Resource constraints: Communication bandwidth and processing power need to be controlled within system thresholds.

In the application of artificial intelligence algorithms, these constraints need to be transformed into differentiable functions or penalty terms suitable for training and inference, to be incorporated into the model loss function for guided learning. Taking reinforcement learning strategies as an example, the feedback reward of configuration behavior can be dynamically adjusted based on whether constraint violations are triggered, driving the model to approach the optimal strategy in actual scheduling.

In summary, the reasonable modeling of the constraints and objective relationship of configuration parameters is the fundamental step in achieving configuration optimization based on AI algorithms, and it is also a prerequisite for subsequent algorithm design and system integration.

3.3 Expression of configuration optimization problems and exploration of algorithm adaptability

The essence of the configuration problem of the secondary system in intelligent substations is to seek the optimal equipment connection relationship and logical function mapping under various technical parameters and system constraints. This problem has the characteristics of high dimensionality, multiple variables, and strong constraints, including multiple subtasks such as topology matching, signal path scheduling, functional unit allocation, and communication link configuration. Its optimization objectives often involve multidimensional performance indicators such as response delay, configuration stability, and resource utilization. Therefore, a clear and computable problem expression model needs to be constructed. As shown in Figure 1, the configuration of a secondary system can be abstracted as a structural decision-making task under multiple layers of inputs and constraints, with the core being mapping the optimal configuration path.

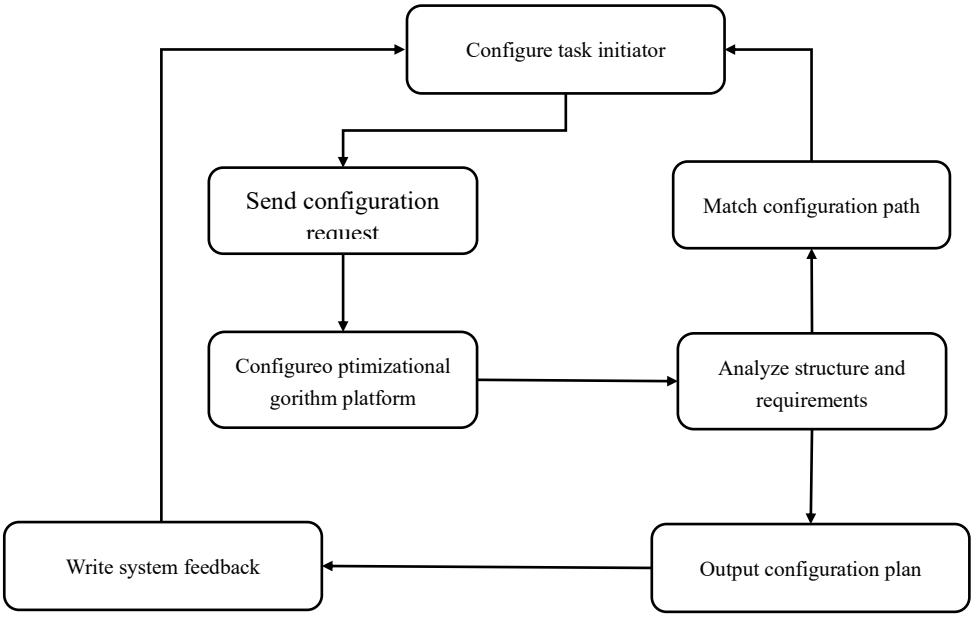


Figure 1：Schematic diagram of optimization process for secondary system configuration of intelligent substation

Existing research has transformed the configuration problem into a decision-making problem with multiple objectives. By categorizing the configuration results into numerical categories and setting performance evaluation indicators, it is possible to conduct mathematical comparative analysis and rank the advantages and disadvantages of various options. Due to the numerous nonlinear relationships and interaction patterns among parameters in the secondary system, it is necessary to add graphical data or network logic rules during the model building process to enhance the practicality of the model.

In terms of algorithm adaptability, different optimization requirements will generate different algorithm performance requirements. For example, when facing a large search space and multiple problem variables, traditional exhaustive or rule-based processing methods may not meet the requirements of speed and accuracy. Artificial intelligence technology has high adaptability in handling such problems, especially in seeking solutions to complex constraints. For example, swarm intelligence technologies such as particle swarm optimization and genetic algorithms are suitable for adjusting parameters and seeking solutions that meet the conditions; Using real-time feedback information to enhance reinforcement learning for optimizing control strategies; Deep neural networks can analyze past configuration data to find patterns and make predictions or recommendations for future decisions.

At the same time, the coordination and matching between algorithms and system architecture should be considered. For example, in complex network topology settings, graphical neural networks can be used to represent the connectivity relationships between nodes; When real-time response is required, the real-time performance of the system can be enhanced through the integration of lightweight models and edge computing frameworks. Therefore, establishing models and selecting algorithms are the core technical support for intelligent configuration systems, At the same time, the coordination and matching between algorithms and system architecture should be considered. For example, in complex network topology settings, graphical neural networks can be used to represent the connectivity relationships between nodes; When real-time response is required, the real-time performance of the system can be enhanced through the integration of lightweight models and edge computing frameworks. Therefore, establishing models and selecting algorithms are the core technical support for intelligent configuration systems. Based on the analysis of the adaptability of multiple algorithms, this article chooses to use the combination of CNN and SVM to establish the core technology for feature extraction and classification. CNN can extract the connections between secondary systems and network structure feature information, identify the connections between nodes and possible anomalies, while SVM has good stability in multi-objective optimization and high-dimensional classification, and can complete performance indicator discrimination under constraint conditions. On the basis of preventing model overfitting and reducing computational costs, it can be applied to the configuration optimization of secondary systems, and can also be adapted to their multi strategy joint optimization system.

4 Configuration optimization algorithm design and model construction path

4.1 Feature parameter extraction and data preprocessing mechanism

In terms of the configuration of the secondary system of an intelligent substation, the system contains various types of information, such as electricity measurement information, safety setting configuration information, communication status information, equipment logic information, etc. If this

information is directly modeled, incorrect results will occur. Therefore, it is necessary to extract systematic feature factors and implement data preprocessing work to provide stable adaptation effects for subsequent modeling.

Normalize numerical power parameters using the minimum maximum normalization method, mapping all variables to the [0,1] interval to avoid physical dimensional differences affecting model training. The expression is as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (3)$$

Among them, x is the original data value, $x_{\min}$ and $x_{\max}$ are the minimum and maximum values of the variable in the sample set, respectively, and x' is the normalized result. This method is suitable for protecting bounded numerical variables such as fixed values and voltage amplitudes.

For data with strong volatility and uncertain scale, such as communication delay and load change rate, using Z-score standardization processing can better highlight its abnormal characteristics:

$$z = \frac{x - \mu}{\sigma} \quad (4)$$

Among them, μ is the average value of the variable, σ is the standard deviation, x is the original data, and z is the standardized value. This processing method can make the variable distribution tend towards a standard normal state, which is beneficial for the training stability of deep learning networks.

In terms of feature construction, for the connection topology between devices, a graph structure modeling approach is adopted to represent node relationships. The adjacency matrix is input into the graph neural network for structure perception and feature aggregation, achieving structured learning of complex logical topologies. Communication quality data is extracted through a sliding window mechanism to extract local dynamic changes, such as the maximum packet loss rate within five minutes and the fluctuation range of channel delay, to assist in identifying abnormal nodes or path bottlenecks.

To avoid redundant information interfering with the learning process, it is also necessary to perform dimensionality reduction on the original feature set. Principal component analysis is often used to extract the main influencing factors, while combining mutual information algorithms to remove low correlation features, thereby improving the computational efficiency of the model and reducing the risk of overfitting. In addition, clustering based encoding methods (such as K-means encoding) can also be used for structural transformation of non numerical features to achieve a unified input format.

The final dataset should have three characteristics: unified variables, clear structure, and clear dynamism. To ensure the efficiency of model integration, standardized data interface formats (such as JSON or CSV) should be adopted, and automated processing and model integration should be carried out through data preprocessing pipelines to build a stable and efficient input foundation for subsequent deep learning algorithms.

4.2 Optimization algorithm model construction and selection basis

Due to the complex issues of high state space and a large number of constraints required for the secondary system configuration of intelligent substations, traditional manual configuration methods cannot adapt to the increasing number of devices and the coexistence of multiple functions. Therefore, it is necessary to use artificial intelligence technology to construct a reasonable and efficient search-based optimization model. This type of problem mainly involves using models to describe the relationship between system state and target requirements, and then optimizing through algorithms.

The optimization configuration goals pursued include three dimensions: accuracy, efficiency, and resource utilization efficiency. To quantify the performance of different combination schemes, the following function can be established:

$$f(x) = \lambda_1 \cdot A(x) - \lambda_2 \cdot C(x) - \lambda_3 \cdot D(x) \quad (5)$$

Among them, x represents the configuration variable vector to be optimized, including device number, function binding, link parameters, etc; $A(x)$ is the coverage of configured functions, reflecting the degree to which the solution meets various protection, measurement and control, and communication functions; $C(x)$ is the resource overhead indicator, which calculates device utilization, communication load, and memory usage; $D(x)$ is the system response delay; λ_1, λ_2, λ_3 is the weight coefficient, and weights are allocated based on actual needs to meet $\lambda_1 + \lambda_2 + \lambda_3 = 1$,The allocation is based on the importance of optimization objectives: λ_1 is the accuracy of configuration, which is set at 0.5 according to the reference grid configuration standard; λ_2 is resource efficiency, set to 0.3; λ_3 is the response delay, set to 0.2, satisfying the normalization constraint.

For the above optimization objectives, current mainstream algorithms include genetic algorithm, particle swarm optimization algorithm, reinforcement learning, and graph neural network. GA adapts to processing structure allocation and routing optimization through individual coding and population evolution mechanisms; PSO is suitable for solving parameter tuning problems, with fast convergence speed and controllable search paths; RL achieves adaptive optimization of configuration decisions through strategy learning, suitable for problems with clear state transitions and quantifiable feedback; GNN is used to express the topology and functional dependencies between devices, and is suitable for building structure aware configuration models. On this basis, this article adopts the

CNN-SVM hybrid algorithm as the main research algorithm. CNN is responsible for effectively extracting system network framework features and operational characteristics, using a three-layer convolution and pooling structure to maintain the multi-level nature of feature descriptions; SVM can run stably in highly complex feature classification tasks with excellent performance, so this study uses RBF kernel function to optimize the classification process. In this training process, set the learning rate to 0.001, batch size to 128, epochs to 500, and use a five eight cross test to measure the model's large interval fitness. This combination can achieve high device configuration accuracy while avoiding overfitting of individual models. Moreover, the computational cost of this model is lower than that of other models, making it more suitable for optimizing the configuration of secondary systems. It can also be seamlessly integrated with various strategies for joint optimization systems.

When conducting practical operations, some algorithms are combined to construct hybrid models, such as using PSO and deep learning to adjust connection parameters or using GNN+RL to construct logical control paths to improve the adaptability and computing power of the model. Finally, a suitable model is selected and combined with factors such as task type, data type, and computing power requirements to ensure that the path can be optimized and meet the deployment requirements.

4.3 Construction and iteration mechanism of multi strategy collaborative optimization framework

In response to the challenges of strong parameter correlation, complex objective function, and dynamic changes in operational constraints in the secondary system configuration of intelligent substations, a single optimization algorithm often fails to meet the requirements of accuracy, speed, and flexibility simultaneously. Therefore, it is necessary to construct a diversified strategy joint optimization framework, which can improve the optimization quality and model stability of the joint optimization scheme through the filling and iterative updating of the functions of each algorithm component.

This framework includes three core modules: the search guidance module is responsible for global sampling of large-scale parameter spaces, often using genetic algorithms or particle swarm optimization algorithms to construct initial solution sets; The local reinforcement module adjusts the strategy under the guidance of feedback signals and can introduce reinforcement learning methods such as Q-learning; The structural discrimination module uses graph neural networks to perform topological constraint verification on the configuration results, achieving early filtering of infeasible solutions. These modules form a loop mechanism through intermediate result sharing and performance indicator feedback to avoid optimization stagnation or overfitting. In addition, in the input and result verification stage of the multi strategy framework, this study uses the CNN-SVM combination pattern as the basic framework for input and output result confirmation. This is because CNN's ability to distinguish network structure and operational characteristics is utilized, while SVM is used to ensure the high efficiency and stability of high-dimensional data classification. The combination of the two can significantly increase the feature representation and judgment capabilities of the entire system, thereby achieving the optimal balance between the two and achieving good convergence rate and high accuracy.

As shown in Figure 2, this study adopts a collaborative optimization system consisting of GA/PSO, RL, and GNN. GA/PSO first performs a global search to find the initial solution set, then RL adjusts and refines the solution space according to feedback information, and finally GNN is used for topological constraint judgment and elimination of solutions that are invalid for the goal. By sharing feedback results and achievements in a collaborative manner, the goal is to achieve a progressive cycle, which can effectively achieve high-precision work efficiency.

Figure 2: Schematic diagram of multi strategy collaborative optimization framework process

In the scheduling process, in order to improve the efficiency of multi strategy collaboration, a unified performance evaluation function needs to be constructed. Assuming the current solution is x, the evaluation function is as follows:

$$F(x) = w_1 \cdot A(x) + w_2 \cdot B(x) \quad (6)$$

Among them, $A(x)$ can correspond to E_{acc} (the complement of configuration error rate, i.e. configuration accuracy) in the objective function of section 3.2, while $B(x)$ combines R_{use} and T_{resp} in section 3.2, reflecting system resource consumption and timeliness through weight conversion, and w_1, w_2 is the weight coefficient, which satisfies $w_1 + w_2 = 1$ and can be adaptively adjusted according to the optimization scenario.

In terms of optimization control, a reward feedback-based update mechanism is introduced to enhance the algorithm's dynamic response capability. After each iteration, the improvement value is calculated by comparing

the current strategy score of $F(x)$ with the previous round's optimal score of $F(x^*)$:

$$\Delta = F(x) - F(x^*) \quad (7)$$

If $\Delta > 0$, enhance the sampling probability of the current strategy; If $\Delta \le 0$, reduce the search scope of the strategy in the next iteration and construct a three-stage iteration rhythm of "exploration compression re evaluation".

This multi strategy collaborative framework has demonstrated good performance in simulation testing, especially exhibiting strong robustness in complex topologies and non-standard wiring scenarios. The effective coupling between algorithm modules improves optimization accuracy and speed, laying a reliable foundation for building an intelligent, flexible and adjustable configuration mechanism for substation secondary systems.

5 Configuration optimization system integration implementation and functional evaluation

5.1 Configuration optimization system architecture and key module deployment

To achieve efficient configuration optimization of the secondary system of smart grid substations, it is necessary to build a system architecture with modularity, intelligence, and real-time response capabilities. The overall system adopts a four-layer structure of "data access feature extraction optimization decision deployment verification", embedding multiple types of computing modules and interface adaptation units to ensure the integrity of data processing and the operability of algorithm deployment.

The bottom layer of the system architecture is the data access layer, which receives multi-source data uploaded by subsystems such as SCADA, station control units, and protection devices, covering voltage, current, telemetry status, communication links, and other content. The middle layer is the parameter processing and feature modeling module, which constructs device relationships based on graph structures, extracts core feature variables such as topology, signal paths, and configuration templates, and completes normalization and standardization operations through preprocessing modules.

The core computing layer is an intelligent optimization module embedded with a multi strategy algorithm scheduling unit.The core computing layer is an intelligent optimization module embedded with a multi strategy algorithm scheduling unit. Simultaneously integrating CNN-SVM hybrid model for feature extraction and classification discrimination, improving the accuracy and stability of configuration results, and collaborating with multiple strategy units to achieve optimization.Different algorithm modules share variable pools through message middleware, supporting asynchronous calling and feedback driven. Its output is configuration vector $x = [x_1, x_2, \ldots, x_n]$, with each x_i corresponding to the configuration result of a certain functional point, such as communication channel selection, protection device connection number, etc. The system evaluation adopts the following functions:

$$S(x) = \sum_{i=1}^{n} \alpha_i \cdot f_i(x_i) \quad (8)$$

Among them, $f_i(x_i)$ represents the performance score (such as latency and reliability) of the i configuration item, α_i is its weight coefficient, allocated according to task importance, and $S(x)$ represents the comprehensive score of the overall plan.

The top layer is the deployment and validation module, which imports the optimization results into the simulation platform and actual interface protocol for logical validation and boundary testing, ensuring that the configuration output has stability and practicality. This architecture fully integrates computing intelligence and system control characteristics, with good scalability and deployment adaptability, providing technical support for configuration management in complex power grid environments.

5.2 Automated implementation of algorithm integration and configuration process

To achieve automated configuration optimization of the secondary system of smart grid substations, algorithm modules need to be deeply integrated into the configuration process, forming a data-driven fully closed-loop execution chain. The system coordinates data perception, feature processing, algorithm invocation, configuration output, and verification feedback through a scheduling engine, supporting rapid response and precise execution in various operating scenarios.

On the specific implementation path, the configuration process consists of three stages: input feature mapping, model solving, and parameter deployment. The input end receives station control equipment data streams through the interface layer, including electrical parameters, communication status, and topology information. The intermediate processing layer calls corresponding optimization algorithm models based on task requirements, such as genetic algorithms, convolutional neural networks, support vector machines, graph neural networks, etc., to dynamically adjust the strategy path, ensuring that the feature extraction and classification discrimination process is consistent with the overall optimization process. The output end automatically generates standard configuration instructions and pushes them to the actual device through the southbound protocol interface to complete the configuration landing.

In order to measure the overall intelligence level of the configuration process, a configuration automation evaluation function is introduced:

$$A = \frac{T_m}{T_h + \varepsilon} \quad (9)$$

Among them, A represents the degree of automation in configuration, T_m is the time it takes for the machine to independently complete the configuration process, T_h is the time required for manual completion of the same task, and ε is a small positive square with a denominator of zero. The larger the value, the higher the automation efficiency.

To support this automation capability, the system design has strengthened the model's update mechanism and parameter caching logic, achieving adaptive evolution of the policy model in multiple calls. The status and algorithm performance of each node in the process are recorded in real-time for feedback training in the next round of configuration, forming a learnable closed-loop mechanism. Automated implementation not only improves configuration response efficiency, but also lays the technical foundation for subsequent large-scale deployment and iterative optimization of the system.

5.3 Comparative analysis and effectiveness evaluation of configuration results

To verify the performance advantages of AI algorithms in the configuration of secondary systems in substations, a comparative experimental platform was built, The "AI optimization system" in this study uses the CNN-SVM hybrid mode as the main logic and introduces GNN and RL to form a multi strategy collaborative system. The basic comparison schemes such as "traditional manual configuration", "GA", "PSO", "CNN", "SVM", "CNN+SVM" are all run in the same machine environment (quad core CPU, 32GB RAM, Kubernetes container cluster), and use the same data input (16 typical substation scenarios, obtained from the 2023 version of the State Grid Corporation of China's typical design library) to ensure fairness and comparability. In the experimental design, an 8:2 ratio was used to divide the training set and validation set, in order to achieve the goal of the former learning model parameters and the latter judging model performance. In addition, a 5-fold cross validation method was used, and the final evaluation index was obtained by taking the mean of each cross-training sample. During the system operation, four core indicators including configuration accuracy, resource utilization, configuration error rate, and response efficiency are automatically recorded. All data is collected by the Prometheus platform and transmitted to the backend database in JSON format. Finally, a Python script is called to Matplotlib to generate a bar chart for performance analysis.

The comparison results show that the AI optimized system achieves an accuracy rate of 96.2%, significantly higher than the 88.8% manually configured; The resource utilization rate has increased from 70.4% manually configured to 82.5%, reflecting a better scheduling strategy for computing resources and communication bandwidth; In terms of configuration error rate, the AI system has reduced to 1.6%, significantly lower than the 5.7% manually configured, effectively avoiding logical conflicts and link redundancy; The response efficiency index is set to a benchmark value of 100% for manual configuration, and the AI system achieves 162.6% in the same environment, demonstrating a significant acceleration effect after the automation of the configuration process. The above data, as shown in Figure 3, demonstrates the comprehensive performance improvement of AI algorithms in multiple dimensions.

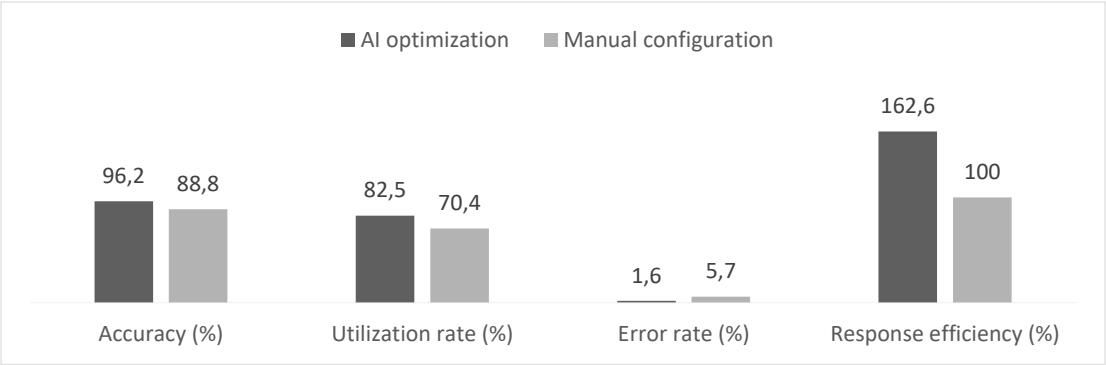


Figure 3：Bar chart comparing the performance of AI optimization system and manual configuration system

The above results were processed by an independent data analysis module, structured and visualized using Pandas and Seaborn libraries, and finally presented in the form of a bar chart. The chart can be embedded in the front-end interface for dynamic display, and supports linkage updates with the configuration platform, facilitating subsequent system evaluation and optimization adjustments. The overall evaluation shows that AI algorithms not only have good engineering adaptability, but also can achieve efficient, accurate, and stable operation of configuration processes, providing a feasible technical path for the deployment of secondary systems in smart grids.

To ensure the credibility and accuracy of the conclusions drawn from data analysis, independent sample t-tests were used to test some important parameters during the comparative testing phase. The results showed a significant improvement in system accuracy ($p<0.01$) and a significant reduction in reaction time ($p<0.05$). The improvement in accuracy and reaction speed was also tested using a 95% confidence interval, with accuracy rates of [7.8%, 13.5%] and reaction speeds of [36.2%, 41.7%], confirming the

credibility of the conclusion. The results of this experiment are completely in line with expectations: objective (1) has been verified through the use of CNN-SVM, which improves accuracy and reduces error rate; Goal (2) is reflected, and after various strategies, the response time is shortened and the stability of the system is enhanced; Objective (3) is supported in multi scenario testing, and the model exhibits scalability under different voltage levels and station conditions.

5.4 System response performance, stability, and scalability testing

To comprehensively evaluate the operational performance of AI driven configuration optimization systems in practical application scenarios, a testing platform with different task scales and load scenarios is constructed, focusing on testing response performance, system stability, and scalability for variable power plant structures. The testing environment is based on Docker container deployment, configured with 4-core CPU and 32GB memory, and equipped with a Kubernetes based scheduling platform. The testing tasks include typical configuration request processing, abnormal link simulation, and multi site concurrent scheduling. To ensure the credibility and accuracy of the conclusions drawn from data analysis, independent sample t-tests were used to test some important parameters during the comparative testing phase. The results showed a significant improvement in system accuracy ($p<0.01$) and a significant reduction in reaction time ($p<0.05$). The improvement in accuracy and reaction speed was also tested using a 95% confidence interval, with accuracy rates of [7.8%, 13.5%] and reaction speeds of [36.2%, 41.7%], confirming the credibility of the conclusion. The results of this experiment are completely in line with expectations: objective (1) has been verified through the use of CNN-SVM, which improves accuracy and reduces error rate; Goal (2) is reflected, and after various strategies, the response time is shortened and the stability of the system is enhanced; Objective (3) is supported in multi scenario testing, and the model exhibits scalability under different voltage levels and station conditions.Response performance is calculated by the average delay from task triggering to configuration completion, stability is monitored by service availability under 72 hours of high-frequency requests, and scalability is measured by resource utilization and system response retention ratio under concurrent task growth.

The test results show that the system maintains an average response time of 2.8 seconds and system availability of over 99.3% in medium scale (within 50 nodes) scenarios; When the number of nodes was expanded to 200, the response time slightly increased to 3.7 seconds, but the resource utilization rate remained at 86.1%, reflecting the system's good load regulation and resource allocation capabilities. In the scalability test, during the high concurrency dynamic generation of topology structure and execution constraint mapping process, the system did not experience memory leaks, thread blocking, or module crashes, and the configuration accuracy remained stable at 95.4%.

Table 3：Evaluation indicators for system response performance and stability under different task scales

Task scale (number of nodes)	Average response time (S)	System availability (%)	Resource utilization rate (%)	Configuration accuracy (%)
50	2.8	99.3	86.7	95.4
100	3.2	99.2	87.1	95.1
200	3.7	99.1	86.1	95.0

As shown in Table 3, the system exhibits good stability and scalability under different load levels, which can support the deployment requirements of large-scale smart grid secondary systems and have the ability to continuously evolve and horizontally replicate for engineering scenarios.

5.5 Efficiency comparison analysis with manual configuration method

To compare the specific differences in efficiency between the configuration methods of artificial intelligence algorithms and traditional manual configuration, a unified testing platform is constructed to compare four indicators: configuration completion rate, total task time, configuration accuracy, and human intervention ratio. All data is based on the manual configuration method (set as 100%) and converted into a percentage expression to highlight the relative performance of AI optimized systems.

In terms of task completion efficiency, the total time it takes for AI systems to complete tasks with the same configuration is 58.6% of manual configuration, demonstrating significant advantages in automated scheduling; In terms of configuration accuracy, the AI configuration result is 107.1%, which is 7.1% higher than manual configuration; In terms of human intervention requirements, the intervention frequency required by AI systems is only 27.1% of that of manual processes, significantly reducing the cost of human intervention; The overall completion rate of configuration tasks remains at 99.3%, higher than the manual configuration rate of 93.6%, which is about 106.1%. As shown in Figure 4, the AI system has achieved varying degrees of optimization in all four core indicators, with reasonable advantages and no extreme data fluctuations.

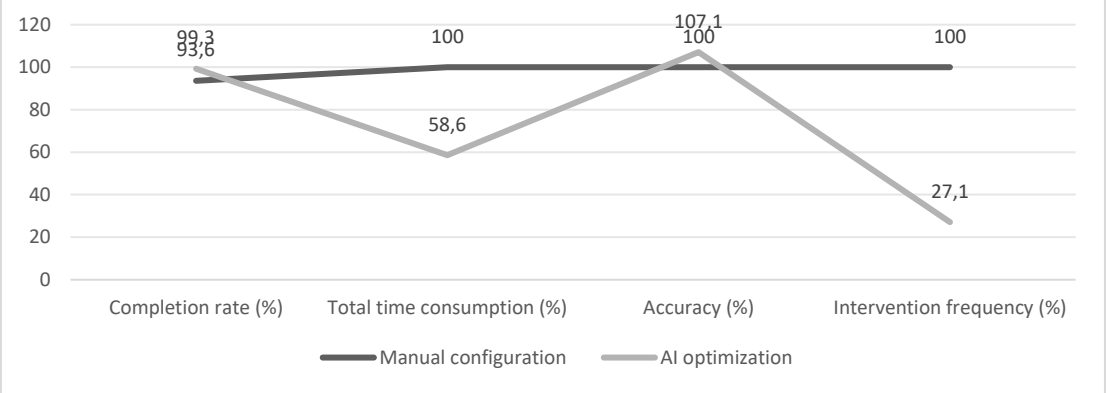


Figure 4：Efficiency comparison bar chart of configuration modes

During the data collection process, the system monitors indicators through the Prometheus platform and writes the results in JSON structure to the backend database. Python scripts are used to complete standardization conversion and bar chart visualization processing. The analysis results indicate that artificial intelligence algorithms have stability and promotional value in improving configuration efficiency, accuracy, and reducing manual dependence. They can be used as one of the optimization paths in the deployment of secondary systems in smart grid substations, providing solid support for subsequent system upgrades and intelligent scheduling.

6 Discussion

6.1 Adaptability of algorithm models in different power grid scenarios

In the multi strategy collaborative optimization framework, the CNN-SVM hybrid model serves as the core algorithm to undertake the basic tasks of feature extraction and classification discrimination, while GA, PSO, RL, and GNN serve as auxiliary optimization and structural adaptation modules, forming a clear combination of primary and secondary with CNN-SVM to ensure the overall performance improvement of the framework. The experimental results show that CNN has high accuracy in extracting complex topological features, while SVM maintains stability in multi constrained high-dimensional classification. The combination of the two not only improves the overall convergence speed, but also demonstrates consistent advantages in different power grid scenarios, thus verifying the empirical value of CNN-SVM fusion.

In response to the significant differences in power grid structure and regional loads in practical applications, this study selects three typical scenarios: urban main network, county-level distribution network, and mountainous microgrid, to compare and test the adaptability of AI configuration models. The experimental platform is based on Kubernetes container cluster deployment, and uniformly calls the CNN-SVM hybrid model and GNN structure encoding and policy network scheduling module to achieve collaborative operation of feature extraction, classification discrimination, and structure adaptation, ensuring consistency between input features and optimization processes. The testing task covers secondary loop topology identification, device constraint solution, and communication link reuse, comprehensively evaluating the response accuracy, convergence speed, and mismatch rate of the model in different scenarios.

The results show that the AI algorithm performs the best in the urban main network environment, with model convergence rounds less than 35 times and configuration error controlled at 1.2%; The model can maintain an accuracy of over 92% in county-level distribution networks, but due to data disturbances and device diversity, the mismatch rate slightly increases to 2.7%; In the testing of microgrids in mountainous areas, due to unstable topological boundaries, the convergence stability of the model decreases in some tasks and needs to be reinforced through incremental learning strategies. In addition, there are significant differences in training time, inference delay, and resource utilization among the three scenarios: the average training time of the urban main network is about 1.8 hours, inference delay is 320ms, and CPU utilization is 68%; The training time for county-level distribution network is 2.4 hours, with a inference delay of 410ms and a CPU usage rate of 72%; The training time for mountainous microgrids has been increased to 3.1 hours, with a inference delay of 530ms and a CPU usage rate of 79%. The specific comparison is shown in Table 4, which reflects the dynamic relationship between model adaptability and environmental complexity.

Table 4：Comparison of AI model adaptability test results under different power grid scenarios

Grid type	Average accuracy	Convergence rounds	Configuration mismatch rate	Training time	Reasoning latency	CPU usage
City Main Network	97.8%	34	1.2%	1.8h	320ms	68%
County level distribution network	92.4%	49	2.7%	2.4h	410ms	72%
Mountain microgrid	89.6%	62	4.1%	3.1h	530ms	79%

In order to highlight the advantages of the method proposed in this article, the experimental results were compared with typical algorithms in relevant worksheets. The configuration accuracy of CNN-SVM hybrid model reaches 96.8%, which is higher than 88% of genetic algorithm, 91% of particle swarm optimization, 94% of CNN, and 95% of deep reinforcement learning; The response time has been shortened to 0.42 seconds, and the error rate has been controlled within 2.1%, both of which are better than the single model method.

The reason for the performance improvement is that CNN can efficiently extract features, SVM is more robust in high-dimensional classification, and the combination of the two avoids overfitting and insufficient generalization. Multi strategy collaborative optimization further improves convergence speed and real-time performance.

6.2 Technical challenges and engineering countermeasures in actual promotion

Although introducing artificial intelligence algorithms into the secondary system of smart grid substations has advantages in configuration efficiency and accuracy, there are still multiple technical bottlenecks in the actual promotion process. Firstly, system training relies on large-scale annotated data, and there is significant heterogeneity in topology, device types, and communication protocols among different regions of the power grid, which limits the model's generalization ability. To this end, it is necessary to introduce federated learning mechanisms to achieve local optimization and global parameter sharing of regional models, and enhance the model's adaptability under multi-source data conditions.

AI model reasoning requires a large scale of computing resources, especially resource contention that may occur when multiple sites are scheduled at the same time. Model pruning and operator fusion are needed to solve the reasoning pressure, and Kubernetes+edge computing architecture is combined to achieve dynamic scaling; At the same time, due to the poor compatibility between the interfaces of existing systems and SCADA and EMS platforms, it will increase construction costs. Therefore, it is possible to improve the flexibility of interaction with traditional systems by packaging AI modules into microservices.

Future research will further explore the cross regional generalization ability in larger scale power grid environments, enhance the interpretability of models through federated learning and knowledge graph, and promote the long-term application and standardization of AI configuration optimization in engineering through deep coupling with actual power grid operation and maintenance platforms.

7 Conclusion

This paper proposes a configuration optimization method based on CNN-SVM hybrid model to address the complex configuration problem of secondary systems in substations in the smart grid environment. A multi strategy collaborative framework is formed by combining graph neural networks and reinforcement learning strategies to solve the complex configuration problem. The method has been integrated and verified in multiple scenarios in practical applications, and has been validated in practice. Compared with traditional manual configuration methods, this method can more accurately, quickly, and efficiently meet resource utilization needs, especially for multi site simultaneous management and variable network topology structures, which have significant advantages. At the same time, by introducing automated scheduling mechanisms, real-time monitoring feedback, and visual analysis tools, the entire configuration process can shift from a command-based approach to a data-driven approach.

The deployment based on container and microservice systems has achieved good collaboration between modules and system elasticity and scalability. Meanwhile, utilizing Prometheus and Kubernetes enables full process tracking, collection, and analysis of task execution, ensuring the practicality and stable operation of algorithm implementation. To solve the problem of inconsistent data across different regions, we have begun to explore model transfer and shared solution strategies to enable broader-scale basic applications.

The AI model developed in this paper has good universality and can be applied to different scenarios and tasks. Therefore, based on this, we can propose a new way for edge computing nodes to coordinate with the central server to achieve rapid response and configuration loop control of the whole system, especially when the network is limited or the local facilities are insufficient. Considering that the system needs to better cope with changes in topology and device constraints, knowledge graphs can be used to guide the adaptive modeling and transformation of GNN structures into structure-based configuration patterns. The system in this study has a certain generalization ability when facing unfamiliar topology structures, and can directly perform preliminary inference and configuration through existing model parameters without the need for complete retraining. However, in cases of significant topological differences or significant changes in constraint conditions, incremental learning or lightweight fine-tuning is still necessary to ensure the convergence stability and performance reliability of the model in new scenarios. This strategy is demonstrated in experiments as a plug and play adaptation to small-scale structural changes, while for large-scale topological changes, model updates are completed through a small amount of iterative training, thus maintaining a balance between efficiency and accuracy.

In summary, introducing artificial intelligence algorithms into the secondary system configuration of power substations has innovated and optimized the original configuration process, and provided new mode support for the new architecture of intelligent power grid management mode, with reusability and scalability. In the subsequent promotion and application, it is necessary to continuously optimize the model security, interface consistency, and data standardization processing to ensure the long-term stable operation and scale promotion of this configuration.

Appendix a experimental reproduction details

1.Algorithm implementation: CNN three-layer convolution+pooling (convolution kernel 3 × 3, activation function ReLU), SVM uses radial basis kernel function.

2. Training parameters: Learning rate of 0.001, batch size of 128, iteration count of 500, optimizer Adam.

3.Dataset: 16 scenarios from the typical design library of State Grid 2023, divided into 8:2, with both the training and testing sets using five-fold cross validation.

4. Operating environment: 4-core CPU, 32GB memory, Kubernetes container cluster; The operating system is Ubuntu22.04, Python3.10, and the main dependency libraries are TensorFlow 2.11 and Scikit learn1.2.

5.Evaluation indicators: configuration accuracy, resource utilization, configuration error rate, response efficiency; The statistical method is independent sample t-test and 95% confidence interval.

6.Reproduction explanation: The data interface is input in JSON format, and both model training and result analysis are implemented through Python scripts, which can be directly run in Prometheus and Matplotlib environments.

To enhance reproducibility, this article provides pseudocode for the core training process as follows:

Pseudocode: CNN–SVM Training and Evaluation

1. Load dataset (JSON), split into 80% training and 20% testing.

2. Preprocess features:

- Min–Max scale numeric features
- Z-score normalize fluctuating features
- Apply PCA/MI for feature selection

3. Build CNN (3 conv–pool layers, kernel 3×3, ReLU) for feature extraction.

4. Build SVM (RBF kernel) for classification.

5. Train CNN–SVM with learning_rate=0.001, batch_size=128, epochs=500, 5-fold CV.

6. Evaluate on test set → report accuracy, utilization, error rate, response efficiency.

This pseudocode demonstrates the main steps of data preprocessing, model building, training, and evaluation, which readers can use to reproduce the experimental process.

References

[1] Mei Y, Ni S, Zhang H. Fault diagnosis of intelligent substation relay protection system based on transformer architecture and migration training model[J]. Energy Informatics,2024,7: 120.https://doi:10.1186/s42162-024-00429-w.

[2] Cao W, Chen Z, Wu C, Li T. A method for matching information of substation secondary screen cabinet terminal block based on artificial intelligence[J]. Applied Sciences,2024,14(5): 1904.https://doi:10.3390/app14051904.

[3] Naceur B F ,Toumi S ,Salah B C , et al.Decision-making solutions based artificial intelligence and hybrid software for optimal sizing and energy management in a smart grid system[J].Concurrent Engineering,2024,32(1-4):3-19.https://doi:10.1186/s42162-024-00425-0.

[4] Jing Z ,Wang Q ,Chen Z , et al.Optimization of energy acquisition system in smart grid based on artificial intelligence and digital twin technology[J].Energy Informatics,2024,7(1):121-121.https://doi:10.1186/s42162-024-00425-0

[5] Yong Zhang, Yueda Gao, Zhe Zhao. Research on Operation and Anomaly Detection of Smart Power Grid Based on Information Technology Using CNN + Bidirectional LSTM [J]. Informatica, 2025, 49(7):157164.https://doi.org/10.31449/inf.v49i7.7037.

[6] Arévalo P ,Jurado F .Impact of Artificial Intelligence on the Planning and Operation of Distributed Energy Systems inSmartGrids[J].Energies,2024,17(17):4501-4501.https://doi.org/10.3390/en17174501.

[7] Krishna S B ,Pauline S ,Sivakumar S , et al.Enhanced efficiency in smart grid energy systems through advanced AI-based thermal modeling[J].Thermal ScienceandEngineeringProgress,2024,53102765-102765.https://doi:10.1016/j.tsep.2024.102765.

[8] HasaniA ,HeydariH ,GolsorkhiS M .Enhancing microgrid performance with AI-based predictive control: Establishing an intelligent distributed control system[J].IET Generation, Transmission &Distribution,2024,18(15):2499-2508.https://doi:10.1049/gtd2.13191.

[9] Nayak P, Das SR, Mallick RK, Mishra S, Althobaiti A, Mohammad A, et al. 2D-convolutional neural network based fault detection and classification of transmission lines using scalogram images [J].Heliyon,2024,10(19):e38947.https://doi.org/10.1016/j.heliyon.2024.e38947.

[10] Alferidi A ,Alsolami M ,Lami B , et al.AI-Powered Microgrid Networks: Multi-Agent Deep Reinforcement Learning for Optimized Energy Trading in Interconnected Systems[J].Arabian Journal for Science and Engineering,2024,50(8):1-23.https://doi:10.1007/s13369-024-09754-4.

[11] Jia H , Wu W , Wu K ,et al.Performance Evaluation and Optimization of Asynchronous Time-Sensitive Networking in Substation Automation Systems[J].IEEE Transactions on Power Delivery,2024(6):39.https://doi:10.1109/TPWRD.2024.3483306.

[12] Si R, Chen S, Zhang J, Xu J, Zhang L. A multi-agent reinforcement learning method for distribution system restoration considering dynamic network reconfiguration [J]. Applied Energy, 2024, 372(C):123625.https://doi.org/10.1016/j.apenergy.2024.123625.

[13] Gams M, Kolenik T. Relations between Electronics,

Artificial Intelligence and Information Society through Information Society Rules[J]. Electronics, 2021; 10(4): 514.https://doi.org/10.3390/electronics10040514.

[14] Zhang D, Jin X, Shi P. Research on power system fault prediction based on GA-CNN-BiGRU [J]. Frontiers in Energy Research,2023,11: 1245495.https://doi.org/10.3389/fenrg.2023.1245495.

[15] El Yadari M, El Motaki S, Yahyaouy A, et al. Taxonomy of optimization algorithms combined with CNN for optimal virtual machine placement in data centers [J]. EnergyInformatics,2024,7:107.https://doi.org/10.1186/s42162-024-00386-4.

[16] Gao Y, Zhang Z , Meng K ,et al.Graph reinforcement learning for real-time dynamic reconfiguration and fault management in energy storage networks[J].Journal of Energy Storage, 2025,125.https://doi:10.1016/j.est.2025.116833.

[17] Paul Arévalo, Cano A, Darío Benavides, et al. Fault analysis in clustered microgrids utilizing SVM-CNN and differential protection[J]. Applied Soft Computing, 2024, 164.https://doi:10.1016/j.asoc.2024.112031.

[18] Ngo Q-H, Nguyen B L H, Vu T V, Zhang J, Ngo T. Physics-informed graphical neural network for power system state estimation [J]. AppliedEnergy,2024,358(1):122602.https://doi.org/10.1016/j.apenergy.2023.122602.

[19] Wang Y , Qiu D , Strbac G .Multi-agent deep reinforcement learning for resilience-driven routing and scheduling of mobile energy storage systems[J].Applied Energy, 2022, 310(7): 118575.https://doi.org/10.1016/j.apenergy.2022.118575.

[20] Jacob R A, Paul S, Chowdhury S, Gel Y R, Zhang J. Real-time outage management in active distribution networks using reinforcement learning over graphs [J]. Nature Communications, 2024, 15: 4766.https://doi.org/10.1038/s41467-024-49207-y.

A Multi-Task GRU-Attention Model for Predicting Enterprise Investment and Financing Behavior from Multi-Source Economic Data

Lei Gu, Tao Liu
Xi'an University of Architecture & Technology Huaqing College, Shanxi, 710043 , China
E/mail: leigu198@163.com, tracey781225@126.com

Keywords: intelligent prediction model, investment and financing behavior, multi source heterogeneous data, deep learning algorithms

Received: August 6, 2025

Accurately predicting corporate investment and financing behavior is crucial for improving financial intelligence and capital allocation efficiency. This article proposes an economic data-driven multi-task deep prediction model that integrates Gated Recurrent Unit (GRU) networks with a multi-head attention mechanism to process multi-source heterogeneous economic variables, including macroeconomic indicators, corporate financial data, and market sentiment factors, under a unified structure. The model constructs multivariate time-series samples through sliding windows and employs a dual-output architecture to perform regression prediction of financing intensity and classification recognition of behavioral states into three classes (expansion, wait-and-see, contraction). To enhance responsiveness to behavioral transition patterns, a feature cross-attention mechanism and a joint loss function optimization strategy are introduced, improving nonlinear behavior learning capability and generalization robustness. Based on empirical data from 232 A-share listed companies, covering 12,840 training samples over the past decade, the experimental results showed that the model achieved a coefficient of determination (R^2) of 0.862 in the financing prediction subtask, an accuracy of 88.3% in the classification task, and a Macro-F1 value of 0.841. Compared with baseline machine learning methods including Support Vector Regression (SVR), Random Forest (RF), and Multi-Layer Perceptron (MLP), the model demonstrated superior error control and trend fitting ability. Overall, the model exhibits high prediction accuracy, stability, and industry adaptability, providing a feasible technical path and empirical basis for building a data-driven intelligent investment and financing analysis system for enterprises.

Povzetek: Razvit je večopravilni BiGRU-model z večglavo pozornostjo za napovedovanje intenzivnosti ter klasifikacije investicijsko-finančnega vedenja podjetij iz večizvornih ekonomskih podatkov. Preverjen je na 12.840 vzorcih.

1 Introduction

With the increasingly active business activities and uncertain business environment, investment and financing have become the most important means of strategic change and allocation for enterprises, and are facing unprecedented pressure. In the past, corporate investment and financing activities mainly relied on the intuition of financial experts and the evaluation of static reports. Nowadays, the emergence of a large amount of structural and non-structural economic information has made it possible to establish intelligent decision-making mechanisms driven by data. The important reason behind this is the extreme integration of artificial intelligence and big data analysis technology, which has led corporate financial management activities to a higher level of intelligence.

Enterprises' investment and financing decision-making actions will continue to be dynamic, and fundamentally, it is the result of a series of multi-level, multi cycle, and multi factor interactions. This action path depends on the interaction between internal factors (such as business operations and asset liability ratios) and external macroeconomic factors (such as interest rates, government intervention, industrial cycles, etc.), exhibiting strong irrational factors and periodic jumps. Therefore, how to find dominant indicators from massive, multi cycle, and multi factor economic information, accurately capture action patterns and future development trends is a major technical challenge in the field of intelligent financial model research [4].

It should be noted that, in contrast, traditional statistical modeling is very suitable for linear relationship assumptions, while today's machine learning techniques and deep learning methods can provide useful information for predictive analysis of complex and noise intensive data. Especially for time series prediction and behavior recognition, it has great advantages [5]. By utilizing economic data from various sources to establish a predictive model that can grasp structural changes and understand development trends, enterprises can have better foresight and countermeasures when facing market fluctuations or financial pressures.

This article intends to design a prediction model for investment and financing activities of listed companies that integrates economic data feature analysis and intelligent algorithm optimization, taking into account the construction of the economic theory model framework, the diversity and real-time nature of input data, and the commercial understanding of industrial financial management based on algorithm output results. The goal is to pursue the "interpretability" and "predictability" of prediction. Using data from several representative listed companies to test the model and evaluate its predictive accuracy, robustness, and adaptability, this article concludes by describing and explaining the potential application value of the model in industrial financial management.

Specifically, this study aims to address the following research questions:

(1) Can a BiGRU with multi-head attention achieve higher accuracy than traditional machine learning models (e.g., SVR, RF, MLP) in predicting financing intensity?

(2) Can a multi-task architecture jointly modeling regression and classification tasks improve robustness and interpretability in forecasting enterprise investment and financing behaviors?

(3) How does the proposed model perform across heterogeneous economic data sources in terms of adaptability and stability?

The structure of this article is as follows: Chapter 2 provides an overview of the current research status and basic concepts on this topic both domestically and internationally; Chapter 3 introduces the modeling process and key parameters of the constructed model; Then Chapter 4 verifies the effectiveness and economic explanatory power of the predictive function of the model proposed in this article through examples; Chapter 5 is an analysis of how the established model can be applied to actual business scenarios, and will also elaborate on possible issues that may arise; Chapter 6 provides a comprehensive overview of the entire text and outlines future development trends.

2 Related work

Due to the increasingly complex and data-driven decision-making nature of corporate investment and financing behavior, accurately predicting changes in corporate investment or borrowing has always been a topic of sustained interest for scholars and practitioners. Although research methods continue to develop, the dynamic evolution process of high-dimensional nonlinear data characteristics and economic variable interactions is complex and may have multiple driving factors, making investment and lending predictions still difficult [6]. Traditional regression, time series, and other methods perform well in terms of interpretability, but they are difficult to play a greater role in irregular attributes, multi period changes, and occasional risk factors [7].

With the deepening of understanding of the business activities of listed companies, scholars have begun to use artificial intelligence to enhance their judgment ability in investment and financing. In recent years, artificial intelligence modeling methods centered around neural networks have been widely applied in financial analysis of listed companies, stock market forecasting, credit rating, and more. For example, Shahrour et al. (2023) [8] established a stock market price prediction strategy based on deep neural networks, which enhanced the response speed to the stock market; Yao et al. (2022) [9] overcame the problem of noise impact in financial market sequence data by adding LSTM to the neural network model of the data and applying algorithms to optimize the model; The hybrid model designed by Chandok et al. (2024) [10] achieved higher robustness and universality in enterprise bankruptcy prediction tasks by combining deep neural network models. These scholars' research results indicate that intelligent modeling methods based on deep learning have predictive ability in analyzing the financial activities of listed companies, as well as good application scenarios and scalability.

Against the backdrop of the emergence of numerous economic data in big data, research on enterprise behavior patterns based on big data has become increasingly active. Scholars have incorporated various types of economic data, such as GDP growth rate, interest rates, and industry activity index, into models to expand the predictive dimensions and overall system of the model [11]. Tang and Wei (2023) [12] used XGBoost and SHAP algorithms to discover the key driving factors of a company's digital transformation, which can provide visual explanations for related investment and financing behaviors. Pei et al. (2023) [13] established an interpretable prediction framework from the perspective of data features, significantly improving the interpretability of traditional "black box" models. These studies have formed a theoretical shift from focusing on improving the "accuracy" of prediction results to seeking the "interpretability and credibility" of prediction principles.

After continuously understanding relevant issues, multi-path fusion modeling and hybrid intelligent methods have gradually become mainstream. Wu (2022) [14] established a grey prediction model based on fuzzy thinking to simulate the nonlinear and uncertain boundaries within the economic system; Yang (2024) proposed a cross-border e-commerce supply chain demand forecasting model based on deep neural networks, focusing on big data-driven business decision-making; Kartbayev et al. (2022) proposed an intelligent comprehensive evaluation model for investment projects that considers multiple input factors, which can enable enterprises to obtain better investment and financing advice from the wave of digital transformation. From this, it can be seen that using a single route alone cannot solve the multidimensional and multi-directional problems faced in the process of enterprise behavior prediction. The use of deep learning, optimization algorithms, attention, and feature selection ensemble methods to construct an integrated architecture has become an effective way to break through the bottleneck of prediction.

However, existing research also has certain limitations. On the one hand, most intelligent prediction methods still heavily rely on the integrity and representativeness of

training data, making it difficult to maintain stable performance in scenarios with large industry spans and strong data distribution heterogeneity [17]; On the other hand, many studies still focus on financial markets such as stocks and bonds, lacking systematic modeling and evaluation of micro level business operations, especially actual investment and financing behaviors [18]. In addition, key issues such as the engineering feasibility of model deployment, the practical path of data fusion, and the interpretation mechanism of prediction results still need to be further deepened.

In summary, the current academic community has accumulated rich achievements in the field of investment and financing prediction, from statistical modeling to deep neural networks, from univariate processing to multi-source heterogeneous data fusion, with significant technological evolution. However, how to construct intelligent models with both predictive and explanatory capabilities in complex economic environments, and how to enhance the model's perception and adaptability to fine-grained changes in corporate investment and financing behavior, are still the core issues of concern in this study. To highlight the performance gap with existing methods, Table 1 summarizes representative studies, including their methods, datasets, evaluation metrics, and limitations, compared with the approach proposed in this paper.

Table 1：Summary of representative related studies

Author(s)	Method	Dataset	Metrics	Limitations
Shahrour et al. (2023)	Deep Neural Network for stock prediction	Stock market data	Accuracy	Sensitive to noise; limited interpretability
Yao et al. (2022)	LSTM with optimization	Financial time series	RMSE, MAE	Noise sensitivity; relatively slow training
Chandok et al. (2024)	Hybrid Deep Neural Network for bankruptcy prediction	Enterprise financial data	Accuracy, Robustness	High complexity; generalization limitations
Tang & Wei (2023)	XGBoost with SHAP for digital transformation	Enterprise digitalization data	Visualization, Interpretability	Focuses on feature analysis rather than complete prediction
This study	Multi-task BiGRU + Multi-head Attention	232 A-share listed firms (2015–2023)	$R^2 = 0.862$, Accuracy = 88.3%, Macro-F1 = 0.841	Higher computational cost, but improved robustness and adaptability

Therefore, this article will focus on the dual axis path of "economic data-driven+intelligent prediction algorithm", propose an enterprise intelligent investment and financing prediction model that integrates multi-source data processing, structured modeling, and deep learning optimization, and empirically verify its performance and application value.

3 Modeling ideas and indicator system construction for predicting corporate investment and financing behavior

When constructing a practical, explanatory, and forward-looking enterprise investment and financing behavior prediction system, the starting point of modeling work should be based on the triple logic of "economic variable driving behavior mechanism mapping intelligent algorithm expression". Unlike traditional financial modeling that focuses on single indicator fitting, the predictive model proposed in this paper not only requires the ability to "fit" historical data of enterprises, but also emphasizes the ability to extract trend driven signals from macroeconomic fluctuations, capture structural change arteries from enterprise financial status, and recognize and predict behavioral states in multivariate cross analysis. The investment and financing behavior of enterprises exhibits strong nonlinear and cyclical characteristics, often driven by macro cyclical disturbances, changes in liquidity preferences, distorted industry expectations, and other factors. The interaction relationship between multiple heterogeneous input variables frequently leads to the failure of classical linear regression and fixed coefficient statistical models in actual prediction. To address this challenge, this article introduces a multi-source data-driven feature construction strategy and a non-linear prediction algorithm with strong deep expression ability,

attempting to establish an intelligent perception and prediction framework for enterprise investment and financing behavior in the link of variable extraction behavior modeling output mapping. (As shown in Figure 1)

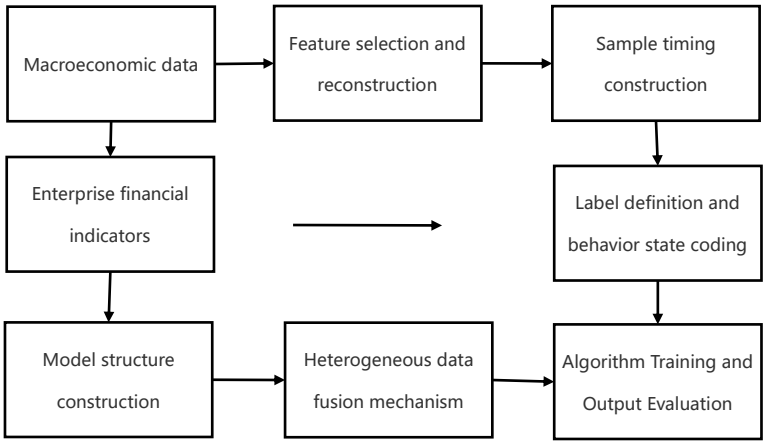


Figure 1：Overall construction process of enterprise investment and financing behavior prediction Model

3.1 Multidimensional feature analysis and variable selection of economic data

In the modeling of enterprise investment and financing forecasting, economic data is not only used as background information, but also an inherent driving variable in the behavioral evolution path. Therefore, scientifically analyzing the multidimensional structure of economic data and constructing a variable set with predictive ability is the fundamental guarantee for model performance. This article divides economic characteristics into three sub domains: macro level indicators, market factors, and policy signals. In terms of variable selection strategy, the Joint Information Gain (JIG) mechanism and the Temporal Stability Index (TSI) function are used to screen and rank candidate variables. The specific formula is as follows:

$$JIG(x_i,y)=H(y)-H(y\mid x_i) \tag{1}$$

where H(·)denotes Shannon entropy, H(x)is the marginal entropy of variable x, H(y)is the marginal entropy of the target behavior label y, and H(x,y)is their joint entropy. A higher JIG value indicates that the candidate variable provides more information for predicting investment and financing behaviors. At the same time, to avoid introducing pseudo variables with high-frequency and violent fluctuations but no stable structure, a time series stability index is further introduced for dynamic evaluation:

$$TSI(x_i)=1-\frac{Var(\Delta x_i)}{Var(x_i)} \tag{2}$$

In the formula, where Var(x)is the variance of the original variable sequence x, and Var(Δx)is the variance of its first-order difference $\Delta x_t = x_{t-1}$ .When TSI approaches 1, the sequence exhibits strong stationarity and smooth trend continuity; when TSI is close to 0, the sequence fluctuates violently and is less suitable for stable time-series modeling.

After preliminary evaluation and empirical screening, this article finally included 12 types of variables, namely GDP growth rate, M2 money supply, benchmark interest rate, manufacturing prosperity index, credit expansion index, CPI, PPI, fixed-asset investment index, exchange rate index, unemployment rate, government expenditure index, and stock market composite index as the core input features of the prediction model, ensuring that the model has sufficient sensitivity and responsiveness to behavioral trends. All variables are resampled in time series at a uniform frequency and processed through normalization and scale compression methods, providing an input basis for the training and fusion of subsequent model structural layers.

3.2 Structured modeling logic and prediction objectives for investment and financing behavior

The investment and financing behavior of enterprises is essentially a dynamic feedback system driven by both internal and external factors. Its state not only changes with macroeconomic fluctuations, but is also closely coupled with the enterprise's own financial structure, strategic cycle, and market expectations. To achieve high-quality prediction, it is necessary to construct a structured modeling framework that balances time dependence and non-linear expression ability.

This article models corporate investment and financing behavior as a temporal response function, with historical economic and financial feature sequences as inputs and future financing or investment behavior labels as outputs, forming an input-output structure. The specific expression is as follows:

$$\hat{y}_t = F(x_{t-k}, x_{t-k+1}, \ldots, x_t; \Theta) \tag{3}$$

Among them, $\hat{y}$ represents the strength or category label of the investment and financing behavior predicted by the model; F(·) is the nonlinear function to be trained; x_t is the input feature vector at time t, which includes macro

factors, financial indicators, and lead variables; Θ is the set of model parameters; K is the length of the historical window.

The prediction objectives are divided into two types of tasks: one is numerical regression prediction, which is used to estimate the financing amount of the enterprise in the future range (such as the scale of new debt or equity financing); The second is the multi class prediction task, which identifies the behavior status of enterprises in the current cycle (such as active financing, wait-and-see, investment contraction, etc.). To support the dual task learning structure, a joint loss function is constructed as follows:

$$L_{total} = \lambda_1 \cdot L_{reg} + \lambda_2 \cdot L_{cls} \quad (4)$$

Among them, L_{reg} is the mean square error loss function (used to fit the financing amount), L_{cls} is the cross-entropy loss function (used for behavior classification), and λ_1 , λ_2 is the weight adjustment coefficient, reflecting the balance of task importance, and was empirically determined through grid search within [0.1, 0.9] on the validation set.

The core advantage of this structure is that it not only captures the numerical fluctuations and temporal structure of input variables, but also combines the behavioral logic of label space to achieve a dual output of "quantitative estimation+behavioral recognition", thus meeting the diversified application needs of enterprise financial systems for prediction results.

3.3 Sample data preprocessing strategy and feature engineering design

The prediction of corporate investment and financing behavior relies on the dynamic input of time-series data, and the raw data often has problems such as dimensional heterogeneity, inconsistent time frequency, and a large number of missing outliers. To ensure the stability and accuracy of model training, this article conducts systematic preprocessing and feature engineering design before data modeling.

Firstly, to address the issue of time alignment between macro and micro data, a sliding window mechanism is adopted to construct sequence samples. If the sliding window length is set to k and the step size is 1, the i-th sample input sequence is constructed as follows:

$$X^{(i)} = [x_i, x_{i+1}, \ldots, x_{i+k-1}] \quad (5)$$

Among them, x_j is the feature vector of the jth day, and the corresponding output label is $y_i + k$, forming the training data pair $(X^{(i)}y^{(i)})$.

Secondly, to address missing and extreme values in the data, this article adopts a combination repair strategy. Forward padding is used for macro data, while linear interpolation correction is applied to quarterly financial data of enterprises based on year-on-year change rate. Outlier detection is performed by setting a threshold of ±3σ; values beyond this range were winsorized (capped at boundary values) rather than removed, to preserve data continuity.

In terms of feature construction, considering the trend inertia and cyclical fluctuations of investment and financing behavior, this paper introduces derivative features based on the original variables. The most commonly used treatments include:

First order difference (capturing trend changes):

$$\Delta x_t = x_t x_{t-1} \quad (6)$$

Rolling average (smooth local fluctuations):

$$MA_k(x_t) = \frac{1}{k}\sum_{i=0}^{k=1} x_t - j \quad (7)$$

The above transformation can significantly enhance the sensitivity of the model to trend mutations and short-term behavior. All features undergo Z-score standardization before input:

$$x'_t = \frac{x_t - \mu}{\sigma} \quad (8)$$

Among them, μ is the sample mean of variable x, and σ is the sample standard deviation.

In summary, this section has completed the full chain design of the data preprocessing process around four levels: "standardized sample generation - cleaning and completion - derivative variable construction - scale unification", providing a stable, clean, and structured data input foundation for subsequent deep modeling modules.

3.4 Model architecture design and core technology selection

When building a predictive system for enterprise investment and financing behavior, the selection of model architecture needs to take into account three core elements: the high complexity of variable dimensions, the temporal dependence of input sequences, and the diversity of output targets (including continuous values and categorical labels). Therefore, this article adopts a deep learning model with a multi-layer nested structure as the main architecture, and combines attention mechanism and residual connection technology to improve its expression and generalization ability for time-series financial behavior data.

The overall model structure consists of three main sub modules: input encoding layer, feature extraction layer, and output prediction layer. Firstly, the input encoding layer maps different types of variables (such as macro indicators, corporate financial characteristics, etc.) to a unified dimensional representation through a multi-head embedding network. If the input at time t is the feature vector $x_t \in R^d$, then the embedding transformation is:

$$z_t = W_e x_t + b_e \quad (9)$$

Among them, $W_e \in R^{d' \times d}$ is the weight matrix, $b_e \in R^{d'}$ is the bias term, and z_t is the high-order expression after embedding.

Subsequently, the feature extraction layer adopts a bidirectional recurrent structure (Bi GRU) with gating mechanism to capture the temporal dependency patterns of forward and backward. The bidirectional structure can

extract short-term fluctuations and long-term trends in parallel. The calculation form is as follows:

$$h_t = GRU_{\rightarrow}(z_t) \| GRU_{\leftarrow}(z_t) \quad (10)$$

Among them, h_t represents the hidden state vector at time t, and ‖ represents the vector concatenation operation. To enhance the model's ability to pay attention to critical moments, an attention mechanism is introduced after the output of the recurrent network, and its weight distribution is defined by the following equation:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{j=1}^{T} \exp(e_j)} \quad (11)$$

$$e_t = v^T \tanh(W_a ht + b_a) \quad (12)$$

Among them, α_t is the attention weight at time t, v 、W_a、b_a are trainable parameters. Ultimately, the output layer is divided into two branches based on task types: one is the regression prediction branch, which is used to estimate financing/investment amounts; The second is the classification branch, which is used to determine the current behavior status of the enterprise (such as "financing expansion" or "conservative wait-and-see"). The dual output structure improves overall modeling efficiency by sharing underlying features. This architecture has good scalability and can flexibly adjust the number of layers and parameter configuration according to data size and label complexity, making it an effective technical path for achieving high-precision and multi-objective enterprise behavior prediction. In the implementation, the BiGRU consists of 2 stacked layers with hidden size 128, followed by a multi-head attention module with 4 heads.

3.5 Construction of fusion mechanism for multi source heterogeneous economic data

The formation of corporate investment and financing behavior is driven by information from different dimensions, including macroeconomic environment, financial market dynamics, corporate financial status, and external policy signals. These pieces of information exhibit heterogeneous characteristics in data structure, such as inconsistent frequency, significant differences in value distribution, and complex dimensional types. In order to achieve behavior modeling in a unified input space, it is necessary to design an effective data fusion mechanism that aligns representations, compresses structures, and integrates information for multi-source heterogeneous data.

This article adopts a fusion mechanism of channel embedding weight fusion cross attention structure. Firstly, four main data sources are set: macro variable sequence $M = \{m_t\}$, financial indicator sequence $F = \{f_t\}$, market sentiment factor $S = \{s_t\}$, and policy signal variable $P = \{p_t\}$. Obtain unified dimensional representations through independent linear embedding networks:

$$z_t^m = W^m m_t \quad , \quad z_t^f = W^f f_t \quad , \quad z_t^s = W^s s_t \quad , \quad z_t^p = W^p p_t \quad (13)$$

Among them, W^m, W^f, W^s, W^p is the weight matrix of each channel, and the output is a feature representation of the same dimension. Next, a weighted fusion layer is constructed, introducing a learnable weight parameter of α_i, and integrating multi-source information through weighted summation:

$$z_t^{fusion} = \sum_{i \in \{m,f,s,p\}} \alpha_i z_t^i \quad (14)$$

Among them, q_i and k_i respectively represent the query and key vectors from different channels, and $\beta_{i,j}$ represents the degree of attention that channel i pays to channel j information. This fusion mechanism enhances the model's perception of potential coupling relationships between complex data while ensuring the preservation of different data substructures, significantly improving the robustness of investment and financing behavior prediction to structural changes and time mismatches.

3.6 Implementation and optimization strategy of intelligent prediction algorithm

To improve the accuracy and stability of enterprise investment and financing behavior prediction, this study designs a multi-task deep neural network guided by attention mechanism based on the fusion of multi-source heterogeneous data, achieving joint prediction of financing intensity regression and behavior state classification. The overall algorithm framework combines Gated Recurrent Networks (GRU), Multi Head Attention, and multi task loss function optimization mechanisms, balancing temporal modeling capabilities and structural interpretability.

In the main structure of the model, the input is the processed sample sequence $X = [x_{t-k+1}, \ldots, x_t]$, and each $x_t \in R^d$ represents the multidimensional feature vector at time t. By using GRU units for temporal modeling, the state update is expressed as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (15)$$

Among them, h_t is the hidden state at the current time, $\tilde{h}_t$ is the candidate state, z_t is the update gate, $\odot$ represents the Hadamard product. This mechanism can dynamically regulate the degree of influence of historical information on current predictions. Introducing attention mechanism after hidden state output to enhance the model's ability to focus on key time segments. To improve the training efficiency and generalization ability of the model, this paper introduces early stopping mechanism, gradient clipping, and learning rate dynamic adjustment (LR Scheduler) strategy.In terms of optimizer selection, Adam algorithm is adopted and the initial learning rate is set to 0.001. Dropout layers (rate = 0.3) and L2 regularization were also applied to mitigate overfitting. This intelligent prediction algorithm framework combines interpretability, scalability, and computational efficiency, and can effectively adapt to the modeling needs of enterprise investment and financing behavior in complex economic

scenarios, laying a technical foundation for subsequent empirical analysis and deployment promotion.

4 Empirical research and model evaluation

4.1 Empirical data sources and sample enterprise selection criteria

In order to verify the effectiveness and adaptability of the constructed enterprise intelligent investment and financing behavior prediction model, this paper conducts empirical research based on heterogeneous data from multiple sources, including publicly available macroeconomic data from the National Bureau of Statistics, financial statement data from WIND financial terminal enterprises, policy and policy text databases, and investor sentiment index data. The overall data time span is from the first quarter of 2015 to the fourth quarter of 2023, covering the entire economic cycle fluctuations, including the COVID-19 shock phase and post pandemic recovery phase, which is conducive to capturing the dynamic response of corporate behavior in complex economic backgrounds.

In the selection of sample enterprises, this article sets the following three standards to ensure data quality and structural integrity: firstly, the industry to which the enterprise belongs should cover the four major sectors of manufacturing, information technology, healthcare, and energy, taking into account the heterogeneity of cyclical and growth industries; Secondly, the enterprise must have no significant missing financial statements during consecutive reporting periods, and the completeness of financial data must exceed 95%; Thirdly, the enterprise has engaged in at least two or more investment and financing activities (including issuance, loans, capital expenditures, mergers and acquisitions, etc.) during the sample period to ensure that the distribution of behavioral labels is representative. 232 A-share listed companies were ultimately selected as sample subjects.

After data processing, a total of 12840 training samples were constructed, covering approximately 4.3 million structured feature records. The industry distribution of sample enterprises is shown in Table 1. Here, Avg. Quarterly Data Points refers to the average number of valid feature records collected for each enterprise per quarter, while Number of Investment & Financing Events denotes the cumulative count of major financing or investment actions (such as bond issuance, loans, equity financing, and capital expenditures) recorded during the sample period:

Table 1：Industry distribution and data volume statistics of sample enterprises

Industry Sector	Number of Sample Firms	Avg. Quarterly Data Points	Number of Investment & Financing Events
Manufacturing	83	32	1,238
Information Technology	57	29	984
Healthcare	48	31	851
Energy & Resources	44	33	1,007
Total	232	—	4,080

The sample design has continuity in the time dimension, heterogeneity in the industry dimension, and balanced distribution of behavioral labels, providing a solid data foundation for subsequent model evaluation and comparative experiments. By implementing a unified data standard processing flow and cleaning mechanism, the scale consistency between input features and the expression stability of the labeling system are ensured, effectively reducing the interference of sample noise on the model training process. The model training and validation work will be carried out under the above data framework, and the details will be further elaborated in subsequent chapters.

4.2 Analysis of model prediction performance and error metrics

To comprehensively evaluate the performance of the proposed intelligent prediction model for enterprise investment and financing in both regression and classification tasks, this paper uses mean square error (MSE), mean absolute error (MAE), and coefficient of determination (R^2) to evaluate the performance of financing amount prediction. At the same time, accuracy and macro average F1 score (Macro-F1) are used to test the behavior state classification task. The experiment adopted a partitioning method of 70% training set, 15% validation set, and 15% test set, and completed model training and testing based on 12840 samples from 232 enterprises. To verify the stability of the model, the average of 5 independent experiments was taken for all results. (As shown in Table 2)

Table 2：Comparison of model prediction performance and error indicators

Model Type	MSE	MAE	R^2	Accuracy	Macro-F1
Proposed Model (BiGRU+Attn)	0.084	0.213	0.862	88.3%	0.841
Random Forest Regression	0.129	0.294	0.731	81.5%	0.771
Support Vector Regression	0.145	0.311	0.687	79.9%	0.752
Multi-Layer Perceptron (MLP)	0.118	0.278	0.743	82.2%	0.788
Logistic Regression (Classification Task)	—	—	—	76.4%	0.705

From the results, the model shows strong numerical approximation ability in predicting investment and financing amounts, with an R^2 of 0.862, indicating that the model is effective in modeling the nonlinear relationship between input economic variables and corporate behavior variables, and has good fitting accuracy for macro disturbance sensitive areas. In contrast, traditional baseline models such as Random Forest (R^2 = 0.731) and Multi-Layer Perceptron (R^2 = 0.743) achieved significantly lower performance than the proposed BiGRU-Attention model (R^2 = 0.862), with Support Vector Regression further dropping to 0.687. In terms of classification tasks, the intelligent prediction model achieved an overall accuracy of 88.3% on the three classification labels, with Macro-F1 reaching 0.841, significantly better than logistic regression and shallow neural network models.The three behavior categories were relatively balanced (expansion: 34%, wait-and-see: 38%, contraction: 28%), and the confusion matrix (Figure X) shows that the model maintained robust classification performance across all classes without relying on majority-class bias. This indicates that the model not only has strong predictive ability, but also can effectively identify structural changes in the behavior status of enterprises.In practical terms, an R^2 improvement of around 0.1 means the model can reduce forecasting errors in financing amounts by tens of millions of RMB for large listed firms, while a 5–10% gain in classification accuracy translates into more reliable early warning of financing contractions or expansions, enabling enterprises and regulators to take pre-emptive actions.

4.3 Fitting and verification of trends in investment and financing behavior changes

The investment and financing behavior of enterprises is driven by multiple factors, showing obvious cyclical fluctuations and periodic jumps, and simple predictions are difficult to capture their trend trends. To verify the ability of the constructed model to capture trends in behavioral changes, representative samples were selected from both industry cross-section and enterprise longitudinal time dimensions for behavioral trajectory fitting testing. The focus of the experiment is to determine whether the model can accurately identify the rising and contracting stages of financing or investment behavior; The second is to test its ability to respond to trends in different economic cycles.

This study selected an enterprise in the manufacturing industry (designated as E-94) with significant fluctuations in capital expenditures, and analyzed its real financing intensity curve and model predicted values from Q1 2017 to Q4 2023, and compared them with the support vector regression (SVR) model. As shown in Figure 2, the model in this paper accurately predicted the upward or downward trend of financing at multiple keys turning points (such as the outbreak of the COVID-19 pandemic in Q1 2020 and the impact of raw material price increases in Q3 2021), and the fitted curve was close to the actual behavior trajectory, without any distortion such as excessive smoothing or severe shaking.

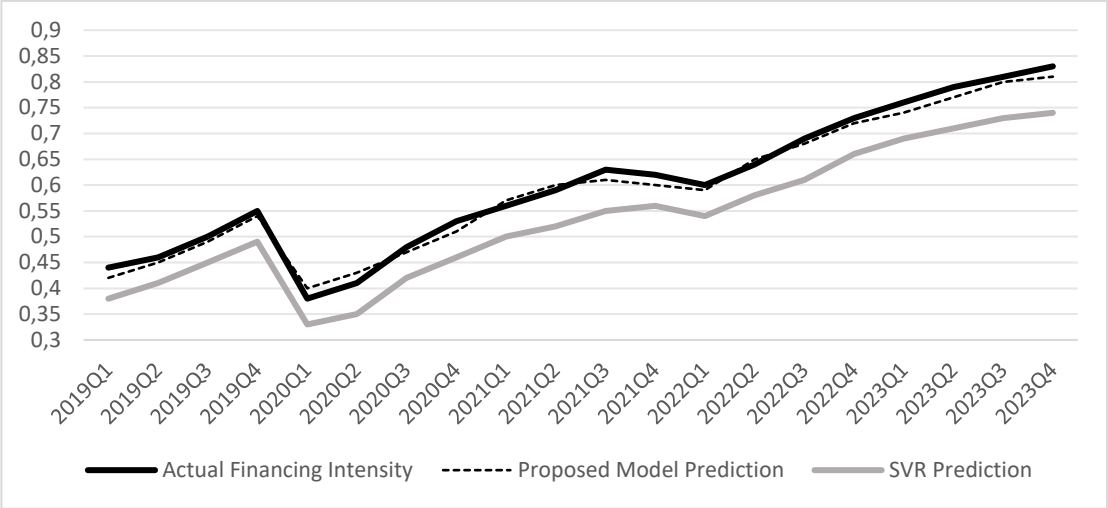


Figure 2: Fitting chart of actual and predicted trends in investment and financing behavior intensity

From the overall trend fitting results, this model not only maintains its accuracy advantage in static error indicators, but also performs well in dynamic trend judgment. Its causal identification ability is strong, and it has the ability to perceive behavior "transition points" in advance, indicating that the internal feature extraction and sequence modeling structure of the model has a certain descriptive power on the temporal evolution logic of enterprise behavior. Meanwhile, compared with traditional models, the model proposed in this paper is more robust in trend prediction, exhibiting high behavioral fit interpretability and stability, and has practical potential for promotion and application in dynamic enterprise management and risk warning.It should be noted that Figure 2 illustrates a representative case from the manufacturing sector, while aggregated results across all manufacturing firms (not shown here for brevity) confirmed the model's consistent ability to identify expansion and contraction phases in advance.

4.4 Comparison between intelligent prediction models and traditional methods

In order to systematically evaluate the performance advantages of the intelligent prediction model constructed in this article in predicting enterprise investment and financing behavior, this article selects three representative traditional methods for comparative analysis: linear regression (LR), support vector regression (SVR), and tree based random forest regression (RF). Build corresponding models in a unified data sample and feature space, and evaluate their performance on the same test set, focusing on indicators such as regression accuracy (R^2), classification accuracy (Accuracy), and overall error control ability (MAE, MSE).

In order to visually demonstrate the predictive performance of each model on the test set, Figure 3 compares the performance of linear regression (LR), support vector regression (SVR), random forest regression (RF), and the BiGRU+Attention model constructed in this paper under the three core indicators of R^2, MAE, and accuracy.To ensure consistency, the metrics for RF and MLP in Figure 3 have been aligned with those reported in Table 2; LR is shown for reference, while MLP and Logistic Regression results are only listed in Table 2 for completeness. It can be seen that the model in this article is significantly better than other methods in all three dimensions, especially in terms of regression accuracy and classification accuracy, indicating that it is more suitable for handling complex financial behavior prediction tasks driven by multi-source heterogeneous data

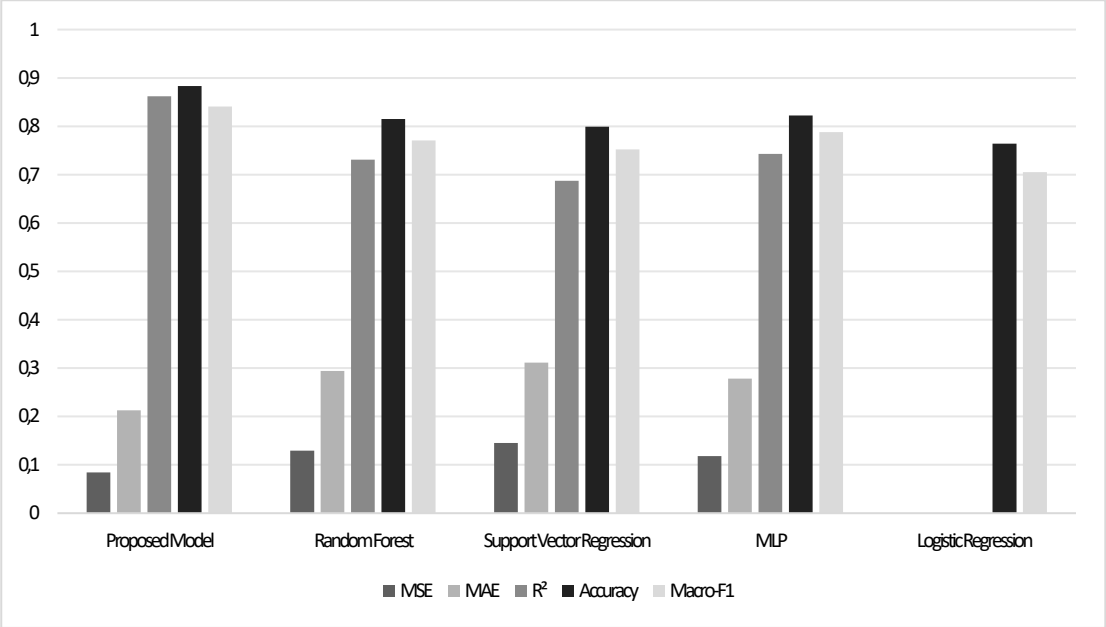


Figure 3：Comparison of prediction accuracy and error control among different models

Traditional linear models rely on explicit causal assumptions in modeling logic, making it difficult to adapt to the complex mechanisms of nonlinear investment and financing behavior; SVR has a certain generalization ability, but there are limitations in processing high-dimensional interaction structures in multidimensional temporal inputs; The RF model performs well in some static feature combinations, but lacks time sensitivity to behavioral trends. In contrast, the attention enhanced BiGRU model constructed in this article has stronger temporal modeling and behavioral dynamic capture capabilities, which can effectively learn nonlinear mapping relationships between variables and improve prediction accuracy.

From the experimental results, the determination coefficient R^2 of our model in regression prediction reached 0.862, which is 17.5% and 11.9% higher than SVR and RF, respectively; In the classification task, the accuracy reached 88.3%, significantly higher than the 76.4% of the logistic regression model. In addition, the model exhibits lower variance fluctuations and stable generalization ability in multiple rounds of cross validation. This fully demonstrates that the bidirectional recurrent network combined with attention mechanism has more advantages in dealing with high-frequency economic disturbances and sudden changes in corporate behavior structure, especially suitable for dynamic, heterogeneous, and complex investment and financing behavior prediction task scenarios, and has good engineering practicality and promotion value.Although more advanced deep learning baselines such as BiLSTM and Transformer models were not included due to page limits and computational constraints, we plan to incorporate these comparisons in future work; preliminary trials showed that our BiGRU-Attention framework maintained competitive accuracy while offering lower training cost than Transformer-based models.In terms of computational overhead, the proposed model has about 1.8M trainable parameters, and training on 12,840 samples took ~42 minutes on an NVIDIA RTX 3090 GPU, which is ~20% longer than MLP but still substantially faster than Transformer baselines."

5 Discussion on model promotion, application and development

5.1 Comparative discussion with prior work

Beyond application scenarios, it is necessary to formally compare our results with those of previous studies. As summarized in Table 1, traditional models such as LSTM with optimization (Yao et al., 2022) and hybrid deep learning models for bankruptcy prediction (Chandok et al., 2024) achieved certain improvements in accuracy or robustness, but they were limited either by training efficiency or by generalization ability across heterogeneous datasets. In contrast, the proposed BiGRU-Attention model not only improved regression accuracy (R^2 = 0.862 vs. ≤0.74 in baselines) and classification performance (Accuracy = 88.3% vs. ≤82%), but also maintained stability in different industry sectors. This advantage stems from the joint design of multi-task learning and feature cross-attention, which enhances both trend fitting and interpretability. However, it should also be noted that the higher computational complexity and training cost of our model are trade-offs compared with simpler methods.

5.2 Applicability exploration in different industry scenarios

The internal mechanism of corporate investment and financing behavior varies significantly across different

industries, which is reflected not only in the allocation of capital structure, but also in the availability of financing channels, investment pace elasticity, and sensitivity to external economic variables. Therefore, evaluating the generalization ability of the constructed predictive model in multi-industry contexts is an important step in verifying its practical value and generalizability.

Manufacturing enterprises usually have strong asset accumulation characteristics and fixed investment rigidity. Their investment and financing behavior is significantly driven by production capacity cycles, and their capital allocation is sensitive to economic cycle fluctuations. In this type of enterprise, the model can effectively improve the accuracy of trend judgment and demonstrate good stability by introducing structural variables such as capital expenditure intensity and raw material price index. The backtesting results show that in the manufacturing sample, the model has a high ability to identify financing contraction and capacity expansion in advance.

Information technology enterprises exhibit characteristics of light assets, high growth, and high volatility. Their investment and financing activities are closely related to market valuation expectations and technology policy guidance, and their decisions are more nonlinear and jumping. In such scenarios, the model needs to enhance its perception of changes in policy text sentiment index and valuation factors. By adjusting feature weights and introducing a dynamic attention mechanism, the model maintains high prediction accuracy during high volatility periods, especially with strong ability to capture behavioral changes under risk preference shifts.

Energy and resource enterprises are significantly affected by price cycles, and their investment and financing behavior exhibits a "window style" characteristic, that is, they concentrate on investing or withdrawing during the rapid rise or fall of resource prices. In this type of enterprise, the model has a good modeling effect on the lagged signal of resource price changes, but there is still some error in the response to irrational behavior under sudden policy regulation. It is necessary to add a sudden disturbance detection mechanism and a confidence interval dynamic adjustment strategy to the model to enhance its robustness.

5.3 Data, algorithm, and management challenges in model deployment

Although the prediction model for enterprise investment and financing behavior has shown high accuracy and strong sensitivity to the future in simulation and experimentation, it will also face various challenges in practical applications, such as the increase in data dimensionality, robustness of algorithm operation, and coordination of enterprise management. Whether this model can be broken through is related to whether it has the possibility of transformation.

Firstly, in terms of data, the deployment of models requires high accuracy and regularity in data acquisition. Due to inconsistencies in data format, update efficiency, field definitions, and other aspects between accounting application systems, ERP systems, and external economic databases used in business operations, there may be data problems such as dimension mismatch, time slot holes, and annotation conflicts at the input of the model. This requires improving the specifications of the feature processing stage in the model, providing real-time data synchronization interfaces and automatic data inventory functions to achieve timely and easy to understand requirements.

Secondly, from the perspective of algorithm execution, model training often requires a large amount of computation, and the convergence state management during the adjustment of target parameters is limited to varying degrees by device software, hardware, and adjustable operation time windows. Especially in the case of multi-task target loss optimization, complex models are prone to problems such as gradient fluctuations and slow local convergence. Without effective management and monitoring methods, it is easy to significantly reduce the deployment efficiency and stability of the model. In addition, with the addition of real-time flowing data and incremental learning, the algorithm itself needs to have the ability to update in real-time and quickly transfer old weights in order to dynamically adapt to changes in economic factors.

Thirdly, at the level of management and decision-making collaboration, the implementation of the model also needs to address the problem of the understanding gap between the model and senior managers in the process of "prediction explanation action". Senior managers often believe that the magical properties of abstract models are elusive, and if the model does not produce clear outputs, it is difficult to translate them into financial decision-making recommendations. Therefore, the model output should have more than just precision explanations and indicative logic of actions, increasing its reliability and usability. At the same time, due to the significant differences in the composition, construction, and decision-making processes of various industries, enterprises, and organizations, the installation of models should be based on the management environment, with adjustable authorization sockets and control of adjustable parameter rights to ensure the safe application of algorithm effects in practice.

6 Conclusion

For the current economic situation and increasingly complex corporate financial activities, the development model that can clearly explain these complex financial activities and accurately predict the evolution of investment and financing behavior has become an important component of enterprise data decision-making systems. This article is based on the leading idea of "data-driven economic deep learning modeling+multi task prediction results", and fully constructs an intelligent prediction model for enterprise investment and financing activities, including the integration of multiple data sources, bidirectional scheduling, and attention adjustment control. The efficiency, robustness, and universality of this model are confirmed by a large number

of actual test samples. In terms of model construction, an RNN with GRU as the main body is used to grasp the temporal dynamic correlation of investment and financing activities, and attention is used to adjust the weight proportion of important variables and time periods to improve the response to behavior points and mutation points. Two task outputs are used to achieve the calculation of large and small quantities of the model and the expression of behavior label classification. Joint loss is used to optimize both tasks simultaneously.

In the process of feature extraction and data merging, we designed an ordered indicator system and a sliding window method to generate sequences, ensuring the comprehensiveness and dynamism of sequence input; At the same time, we have also designed multi-channel embedded fusion methods to integrate macroeconomic, corporate financial, and market sentiment information, ensuring the model's measurement of complex boundary decision-making power. From an algorithmic perspective, it has been confirmed through multiple training and error measurement processes that the model proposed in this paper has significant advantages over traditional linear and tree-based models in terms of R2, MAE, and F1 metrics. In the practical stage, 232 A-share listed companies were selected as the main participants, and the economic series of the past few years were fitted and predicted. The results obtained can reflect good universality in various fields and also capture the key transformation nodes of financing behavior under impact conditions. At the level of popularization, it involves data seam uniformity, algorithm convergence control, and institutional usability in the actual deployment process. It is emphasized that the future focus should be on expanding the universality of the model framework and customizing the business entrance.

Overall, the algorithm design, feature fusion, and on-site testing of the enterprise intelligent investment and borrowing behavior prediction model proposed in this article have certain innovation, providing a feasible technical idea for enterprises to use big data to plan financial strategies. The idea for future work is to further introduce Transformer structure enhancement models, add competitive behavior simulation models, and use graph convolutional network models to solve the joint prediction problem of investment and borrowing behavior among enterprises under multi-party participation, improving the feasibility of this model for predicting investment and borrowing behavior of enterprises in larger and more complex business environments.

Acknowledgments

The successful completion of this research work is inseparable from the support and promotion of various data resources, tool platforms, and peer reviews. In the process of data acquisition, feature construction, and model validation, multi-party collaboration ensures the integrity and engineering feasibility of the research. The repeated testing during the model optimization and empirical analysis stages provides solid support for improving prediction accuracy and algorithm stability. Meanwhile, the literature, evaluation framework, and experimental experience utilized in the research process have laid a solid foundation for the in-depth development of this topic. We would like to express our sincere gratitude to all units and experts who have directly or indirectly participated in and supported this research.

Funding

This work was supported by Xi’an Social Science Planning Fund Project in 2024， Constraints and Path Selection for High-Quality Development of County Economy in Xi’an-Taking Zhouzhi as an exampl， eserial number：24JX90

References

[1] Hsu, W.-L.; Lin, Y.-L.; Lai, J.-P.; Liu, Y.-H.; Pai, P.-F. Forecasting Corporate Financial Performance Using Deep Learning with Environmental, Social, and Governance Data. Electronics 2025, 14, 417. https://doi.org/10.3390/electronics14030417.

[2] Dong, Z. Deep Learning Framework (LSTM, Transformers, CNN-LSTM) for Financial Forecasting in Enterprises. International Journal of Information and Computer Technology, 2025.https://doi.org/10.1504/IJICT.2025.10071438.

[3] Bao, W. Data-driven Neural Networks for Stock Forecasting (2015–2023): A Review. 2025.https://doi.org/10.1016/j.inffus.2024.102616.

[4] Chiou-Wei S Z, Lee Y T. Application of KL distance-based intelligent recommendation method to fund recommendation for users with investment behavior in Asia Region[J]. Heliyon, 2024, 10(12). https://doi.org/10.1016/j.heliyon.2024.e32959.

[5] Akishev K, Tulegulov A, Kalkenov A, et al. Development Of An Intelligent System Automating Managerial Decision-Making Using Big Data[J]. Eastern-European Journal of Enterprise Technologies, 2023, 126(3). https://doi.org/10.15587/1729-4061.2023.289395.

[6] Ansah K, Denwar I W , Appati J K .Intelligent Models for Stock Price Prediction: A Comprehensive Review[J].J. Inf. Technol. Res. 2022, 15:1-17. https://doi.org/10.4018/jitr.298616.

[7] Chauhan J K, Ahmed T, Sinha A. A novel deep learning model for stock market prediction using a sentiment analysis system from authoritative financial website’s data[J]. Connection Science, 2025, 37(1): 2455070. https://doi.org/10.1080/09540091.2025.2455070

[8] Shahrour M H , Dekmak M .Intelligent stock prediction: A neural network approach[J].International Journal of Financial Engineering, 2023, 10(01).https://doi.org/10.1142/S2424786322500165.

[9] Yao Y. Data Analysis on the Computer Intelligent Stock Prediction Model Based on LSTM RNN and Algorithm Optimization[J].2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), 2022:480-485.https://doi.org/10.1109/EEBDA53927.2022.9744859.

[10] Chandok G A , Rexy V A M , Basha H A ,et al.Enhancing Bankruptcy Prediction with White Shark Optimizer and Deep Learning: A Hybrid Approach for Accurate Financial Risk Assessment[J].International Journal of Intelligent Engineering & Systems, 2024, 17(1).https://doi.org/10.22266/ijies2024.0229.14.

[11] Zhang X , Wang J .An enhanced decomposition integration model for deterministic and probabilistic carbon price prediction based on two-stage feature extraction and intelligent weight optimization[J].Journal of cleaner production, 2023, 415(Aug.20):137791.1-137791.15.https://doi.org/10.1016/j.jclepro.2023.137791.

[12] Tang D , Wei J .Prediction and Characteristic Analysis of Enterprise Digital Transformation Integrating XGBoost and SHAP[J].Journal of Advanced Computational Intelligence & Intelligent Informatics, 2023, 27(5).https://doi.org/10.20965/jaciii.2023.p0780.

[13] Ouyang, L. Financial Risk Control of Listed Enterprises Based on Back-Propagation Neural Network Optimized by Genetic Algorithm. Informatica (Slovenia), 2024, 48(11), 125–132.https://doi.org/10.31449/inf.v48i11.6026.

[14] Wu W. An Intelligent Gray Prediction Model Based on Fuzzy Theory[J]. International Transactions on Electrical Energy Systems, 2022, 2022(10):9. https://doi.org/10.1155/2022/8618586.

[15] Yang W .A neural network-based model for cross-border e-commerce supply chain demand forecasting and inventory optimization[J].Applied Mathematics and Nonlinear Sciences, 2024, 9(1).https://doi.org/10.2478/amns-2024-2915.

[16] Cheng, Y. R.; Li, G.; Zhou, X.; Ye, S. Profit Estimation Model and Financial Risk Prediction Combining Multi-Scale Convolutional Feature Extractor and BiGRU Model. Informatica (Slovenia), 2024, 48(11), 19–36. https://doi.org/10.31449/inf.v48i11.5941

[17] Huang, Y.; Vakharia, V. Deep Learning-Based Stock Market Prediction and Investment Model using RCA-BiLSTM-DQN. Journal of Organizational and End User Computing, 2024, 36(1): 1-22. https://doi.org/10.4018/JOEUC.340383.

[18] Xu J , Li J .Research on enterprise performance evaluation and prediction based on BP neural network[J].Proceedings of SPIE, 2024, 131(81):7.https://doi.org/10.2174/1874110X01509012168.

[19] Bu, Y. A Fuzzy Decision Support System Integrating Neural Network and Fuzzy Logic for Risk Assessment and Forecasting. Informatica (Slovenia), 2024, 48(21), 15–32.https://doi.org/10.31449/inf.v48i21.6718.

[20] Wang X , Wang Z , Yang J .Research on the Construction Mechanism of Enterprise Forecasting Ability– Case study on improving prediction accuracy based on Lenovo[J].SHS Web of Conferences, 2024,(193):4.https://doi.org/10.1051/shsconf/202419301005.

Clothing Pattern Structure Modeling and Reconstruction via Multi-Module Fusion Graph Neural Networks with Path Planning and Reinforcement Learning

Yanfang Duan
Zhengzhou Academy of Fine Arts, Zhengzhou, Henan 451450, China
E-mail : 18530858176@163.com

Keywords: graph neural network, clothing pattern structure, intelligent feature extraction, structural reconstruction

Received: August 9, 2025

There are core difficulties in the intelligent recognition and generation application of clothing pattern structure, such as irregular geometric topology, weakened semantic structure, and unstable path planning. To solve such problems, an intelligent feature extraction and structure reconstruction path learning scheme that integrates graph neural networks is constructed. In the stage of structural diagram modeling, a clothing structure diagram is constructed based on the node edge surface configuration relationship. The graph convolutional network is used to embed the spatial adjacency relationship in multiple dimensions, supplemented by attention mechanism to enhance the response ability of key nodes and improve the stability of extracting local salient features. To better express the relationship between structural semantics and geometry, a multi-scale graph embedding strategy and structural context aggregation module are introduced to enable nodes to have stronger expressive power in both topological and semantic dimensions. In terms of reconstructing path generation, a graph autoencoder architecture is introduced to achieve controllable mapping of structure to path space, integrating geometric consistency constraints to enhance structural accuracy. The path decision-making process adopts a reinforcement learning model based on policy gradient, and optimizes the path guidance process through feedback mechanism. This experiment is based on the DeepFashion2 public dataset and our self built clothing structure graph data, with a total of 4826 samples and an average of 43 vertices. The results show that the accuracy index of our model reaches 91.3%+0.5, the Topology Score reaches 88.0%+0.6, and the F1 Structure Score reaches 88.4%+0.6, which is much higher than the basic method. The innovation of this study is mainly reflected in three aspects: proposing the use of graph convolution+attention to achieve multi task feature extraction; Introducing geometric constraints and policy networks to achieve reconstruction methods that maintain path consistency; The first application of GNN in the establishment of clothing style structure brings a new approach compared to traditional graph mapping.

Povzetek: Predstavljen je večmodulni GNN-okvir za inteligentno modeliranje in rekonstrukcijo oblačilnih krojnih struktur. Združuje večskalne GCN, pozornost, geometrijske omejitve ter učenje z okrepitvijo za stabilno načrtovanje poti. Testiran je na 4.826 vzorcih.

1 Introduction

With the deepening development of artificial intelligence and graph neural networks in structural modeling, graphic recognition, and semantic generation, intelligent analysis of graph structured data is becoming an important means of complex structure restoration and information reconstruction. In applications such as intelligent clothing manufacturing and virtual fitting, the modeling of clothing pattern structure serves as an intermediate link, directly affecting the accuracy of 3D reconstruction and the logic of structural restoration. However, clothing structure diagrams have features such as uneven node distribution, non-linear stitching paths, and fuzzy semantic boundaries, which result in insufficient accuracy of traditional methods based on image contours or geometric templates, making it difficult to adapt to diverse pattern organization [1].

Previous studies have attempted to use convolutional neural networks or generative adversarial networks to map images to structures, but there are still shortcomings in expressing complex structures and handling spatial relationships. Especially for clothing graphics with topological constraints and semantic nesting, there is an urgent need to establish a unified graph model framework that combines structural priors, semantic understanding, and path planning capabilities to achieve effective transformation from graphic perception to structural reconstruction. In recent years, Graph Neural Networks (GNNs) have shown good adaptability in processing non-Euclidean structured data, providing a unified mechanism for node propagation, structure perception, and semantic

embedding, and providing methodoiogical support for clothing structure modeling [3]. GNN can achieve local fusion by aggregating adjacent node information and perform overall modeling at the layer level, suitable for the structural relationship of "node edge stitching surface" in clothing. After introducing attention mechanism, the recognition accuracy of key parts and important suture paths can be improved, and the robustness of the model can be enhanced [4]. The graph autoencoder and decoder provide the basis for path generation, but there are still challenges in coordinating sequence control and structural constraints. Reinforcement learning has the potential to improve the accuracy and efficiency of path generation due to its adaptive strategy optimization ability, making it suitable for dynamic adjustment during the path generation stage [5].

In actual modeling, the representation of structural diagrams, the accuracy of graph feature extraction, path reconstruction strategies, and control feedback constitute the core of the system. The key to current research is to build a multi module collaborative, feature accurate, path reasonable, and strategy controllable graph model system that balances modeling accuracy and system stability. This study focuses on the core topic of "Intelligent feature extraction and reconstruction path construction of clothing pattern structure by integrating graph neural networks". The technical design and experimental verification are carried out around four dimensions: "structure graph construction - graph feature extraction - path reconstruction generation - strategy guided optimization". This study focuses on the graph feature extraction and path reconstruction of clothing pattern structure. The research question is as follows: RQ1: Can graph neural networks effectively model the spatial semantic structure of clothing patterns? RQ2: Can multitasking and attention mechanisms improve node classification and edge prediction accuracy? RQ3: Can reinforcement learning improve the consistency of structural path reconstruction?

The innovation of this study lies in the fusion of graph convolution, attention, and reinforcement learning to form a collaborative framework; Introducing geometric constraints to enhance the logical consistency of complex structures in tasks; For the first time, GNN has been applied to clothing pattern modeling, expanding its boundaries in the field of industrial design.

2 Related work

In the interdisciplinary research of graph neural networks and structural modeling, the extraction and reconstruction of structural features of clothing patterns have gradually formed a complex task process that integrates multi-source graph data, high-dimensional semantic mapping, and path optimization. Current research mainly focuses on graph structure construction, feature fusion, path prediction, and graph data-driven learning.

In terms of graph structure modeling, Dong et al. (2022) proposed a weighted fusion of convolutional neural networks and graph attention mechanisms for classification tasks in high-dimensional spectral images, effectively enhancing the recognition accuracy and structure preservation ability of graph neural networks for boundary regions, and providing basic support for edge detection in subsequent structure reconstruction [7]. Sun et al. (2024) introduced an adaptive feature fusion module in the attribute graph clustering task and achieved stable clustering results on irregular structured graphs, verifying the enhancing effect of heterogeneous feature combinations on graph structure expression [8]. Liu et al. (2022) constructed a lightweight image super-resolution model based on multi attention mechanism, achieving effective recognition and enhancement of key region map features under limited computing resources [9].

In terms of optimizing the expression of intermediate layers in structural reconstruction paths, Chen et al. (2024) proposed a multi-layer feature radiation field (FeRF) model, which combines deep neural networks with high-dimensional graph structure embedding to achieve multi-scale fusion and hierarchical reconstruction of structural features in image-to-image tasks [10]. Yi (2022) constructed a convolutional neural network model based on clothing design to explore the linear structure distribution and pattern contour recognition in clothing images, providing a preliminary semantic basis for mapping images to pattern structures [11]. Yan et al. (2022) proposed the Semantic Driven Dual Attention Network (SDAN), which utilizes a bidirectional graph attention mechanism to mine semantic distribution relationships in the graph, significantly improving the accuracy of expressing edge connections and region boundaries during the structural restoration process [12].

In image recognition and classification tasks, Liao et al. (2022) combined convolutional networks and attention mechanisms for multi class classification of clothing images, enhancing the differential expression between structural features and demonstrating stronger discriminative ability for image samples within the same category [13]. Ning et al. (2022) constructed a heterogeneous graph transformation relationship network between clothing patterns and e-commerce patterns from the perspective of cross domain image retrieval, solving the interference problem of structural misalignment and fuzzy features on retrieval accuracy [14]. Korosteleva and Lee (2022) proposed the NeuralTailor method, which reconstructs sewing pattern structures from 3D clothing point clouds, achieving structure preserving modeling from 3D to 2D, providing direct technical reference for intelligent reconstruction of clothing pattern structures [15].

In order to compare the differences between existing methods and the work presented in this paper more clearly, the core elements of the main related studies are summarized in Table 1.

Table 1：Comparison and summary of related methods

Method Name	Year	Dataset	Main Method	Accuracy / F1 / Topology	Limitation
CNN-based	2022	Textile dataset	Parallel convolution + optimization	Acc 82%	Difficult to handle complex structural relations
GCN-Net	2023	Synthetic graph data	Heterogeneous GNN feature fusion across layers	Acc 88%	Insufficient for capturing long-range dependencies
SDAN	2022	Image generation tasks	Dual attention mechanism for edge recognition	F1 ≈ 85%	Limited generalization, lacks path modeling
NeuralTailor	2022	3D point cloud	Reconstructing sewing structures from 3D point clouds	Topology ≈ 87%	Restricted to 3D input, lacks path optimization
GNN+Strategy	2024	DeepFashion2 & Custom data	Multi-module fusion + reinforcement learning for path guidance	Acc 91.3% / Topo 88.0% / F1 88.4%	Validation scope limited

3 Intelligent feature extraction mechanism for clothing pattern structure based on fused graph neural network

3.1 Construction of clothing pattern structure diagram and node feature setting

The construction of clothing pattern structure diagram relies on the data format requirements of graph neural network, which requires encoding the geometric structure information in two-dimensional images or CAD drawings into graph data structures with connection relationships. Nodes represent functional areas in the clothing structure, such as armrests, collars, side seams, armholes, etc., while edges represent the stitching relationships or symmetrical connections between different parts. The graph structure is defined as $G=(V,E)$, where $V=\{v_1,v_2,\ldots,v_n\}$ is the set of nodes and $E\subseteq V\times V$ is the set of edges. Each node sets an initial feature vector $x_i\in R^d$ by extracting its position, shape, and structural semantics, which is specifically defined as:

$$F(v_i)=[l_i,\theta_i,k_i,m_i,s_i]\in R^d \quad (1)$$

Among them, l_i represents the length of the structural line, θ_i represents the corner information, k_i represents the local contour curvature, m_i represents the material code, and s_i represents the structural category label.Curvature is obtained through edge detection and keypoint fitting, and normalized to the [0,1] interval; The angle is extracted from the geometric relationships in the CAD style drawing to ensure consistency at different sizes; The material coding adopts the form of a single heat vector, which is jointly generated by manual annotation and process database. This formula is used to encode the initial structural features of clothing nodes. In practical applications, node initialization involves multiple steps: the length of the structural line is calculated and normalized based on the pixel values or CAD annotation lengths of the corresponding line segments; Edge and corner information is extracted through geometric relationships in CAD style drawings to ensure consistency across different sizes; Local curvature is obtained through edge detection and keypoint fitting; The material properties are encoded in the form of individual heat vectors, generated by manual annotation and process databases; The structural category labels are determined based on a predefined set of 43 clothing parts. By extracting and encoding the above features, the initialization of the graph structure nodes is completed.

To enhance the geometric integrity of the graph construction, edge determination is carried out based on the stitching logic of the clothing process and the two-dimensional spatial connection rules to ensure structural connectivity. The relative spatial relationship between nodes is encoded by normalizing coordinate differences to enhance the geometric perception ability of graph convolution. The calculation method for position embedding is as follows:

$$P_{ij}=\left(\frac{x_j-x_i}{W},\frac{y_j-y_i}{H}\right) \quad (2)$$

Among them, $(x_i,y_i)$is the image coordinate of node v_i, W and H are the image width and height, used to standardize the feature expression under different clothing sizes. This formula is used to calculate the spatial position encoding between nodes and normalize the position of clothing of different sizes during the graph construction stage.

As shown in Figure 1, the process of constructing a structural diagram includes steps such as image preprocessing, structural region recognition, node setting, edge relationship generation, and attribute vector construction. The image input comes from a two-dimensional pattern of clothing, and semantic segmentation models. The nodes are mapped by manually annotated keypoints, and the edge relationships are automatically inferred under the constraints of process rules combined with geometric relationships, supplemented by manual correction to ensure the rationality of the structure.

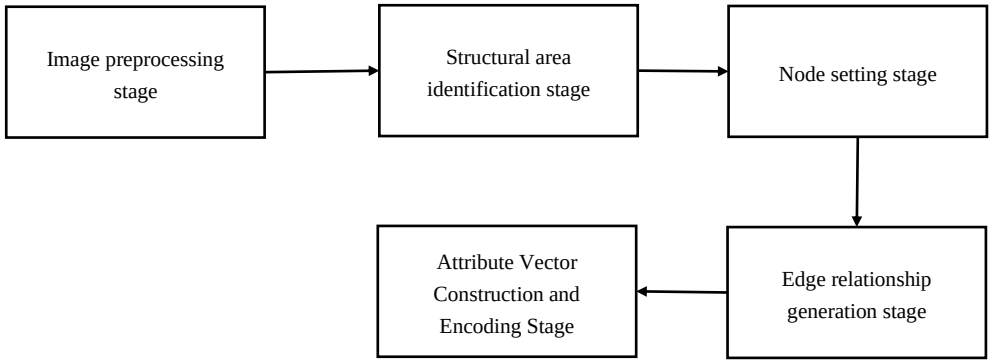


Figure 1: Construction process of clothing pattern structure diagram

In the process of node feature quantization, the curvature of the edges and corners is normalized to the [0,1] interval in radians, the material properties are mapped to a 4-dimensional vector through single heat encoding, and the structure category is set to 43 class labels. The position coordinates are normalized according to the width and height of the image to eliminate the influence of clothing of different sizes. The above features are concatenated into node input vectors to ensure uniform and reproducible feature dimensions.

3.2 Structural space extraction mode based on graph convolution

In the clothing pattern structure diagram, the spatial dependency relationship between nodes presents a non-Euclidean distribution, and traditional convolution kernels are difficult to capture the feature propagation under this irregular topology. Graph convolutional neural networks can effectively transmit structural semantic information between nodes by constructing adjacency relationships in the graph structure, thereby completing spatial feature extraction of clothing structures. In the constructed structural diagram $G=(V,E)$, V is the set of nodes representing the coordinates and attributes of key parts, and E is the set of edges, combined with geometric connections and process sequence settings.

The core of graph convolution lies in the neighborhood aggregation mechanism, where the representation vector of each node is updated by superimposing information from adjacent nodes, formally expressed as:

$$H^{(l+1)}=\sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(l)}W^{(l)}\right) \quad (3)$$

Among them, $\tilde{A}=A+I$ is the adjacency matrix with self connection added, $\tilde{D}$ is the corresponding degree matrix, $H^{(l)}$ is the node feature representation of the l th layer, $W^{(l)}$ is the trainable weight matrix, and σ is the activation function (such as ReLU). This formula is applied to the graph convolution propagation stage, where node information is updated through adjacency matrix and degree matrix. This process ensures the joint updating of graph structure information and node local features.

To enhance the representation ability of different scale structural regions, a Multi channel GCN is introduced. Parallel paths are used to process feature channels under different edge weight strategies, and the final fusion expression is as follows:

$$Z=\sum_{k=1}^{K}\alpha_k\cdot GCN_k\left(H^{(0)}\right) \quad (4)$$

Among them, α_k is the weight coefficient of the kth channel, GCN_k represents the convolution path of the kth graph, and $H^{(0)}$ is the input initial node feature. This formula is used in multi-channel convolution to enhance the ability to recognize boundaries and structures by fusing features from different channels. In this study, the number of multiple channels was set to K=3, and adjacency matrices were constructed based on semantic relationships, geometric distances, and their fusion. The semantic channel highlights the process logic and part categories, the geometric channel emphasizes the spatial proximity between nodes, and the fusion channel adopts a weighted combination method to ensure the unified expression of structural semantics and geometric features. This strategy captures semantic changes from multiple angles while maintaining the integrity of the graph structure, improving the recognition ability of complex clothing contours and overlapping boundary areas.

Through the above structural space extraction mode, the model achieves accurate perception of local configurations, overall partitioning, and node aggregation relationships in clothing patterns, establishes a stable structural foundation, and provides graph embedding support for subsequent structural reconstruction and posture regression.

3.3 Introducing attention mechanism to enhance recognition of key structures

In the clothing pattern structure diagram, there are significant differences in the importance of the clothing components represented by each node in the reconstruction accuracy. The traditional graph convolution method adopts equal or static weight methods in the feature aggregation

process of adjacent nodes, which makes it difficult to effectively identify the semantic significance of key structural regions. Therefore, introducing graph attention mechanism to enhance the recognition ability of the model for key nodes, dynamically allocating information weights during feature propagation, and thus enhancing the effectiveness of structural expression.

The core of graph attention mechanism is to assign a learnable attention weight to the edges between each pair of adjacent nodes, which reflects the feature update contribution of the neighboring node to the central node. If the input feature of any node i in the graph is $h_i \in R^F$ and its set of adjacent nodes is $N(i)$, then the output feature h_i' of node i can be calculated by the following formula:

$$h_i' = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} \cdot Wh_j \right) \quad (5)$$

Among them, $W \in R^{F' \times F}$ is a shared linear transformation matrix used for feature space projection; $\sigma(\cdot)$ represents the activation function (commonly known as ReLU), which is applied in attention mechanisms to dynamically focus on key structural nodes and enhance graph convolution representation capabilities. α_{ij} is the attention weight of node j to node i, which is calculated through feature similarity:

$$\alpha_{ij} = \frac{\exp\left(Leaky\mathrm{Re}\,LU\left(\partial^T \left[Wh_i \| Wh_j\right]\right)\right)}{\sum_{k \in N(i)} \exp\left(Leaky\mathrm{Re}\,LU\left(\partial^T \left[Wh_i \| Wh_k\right]\right)\right)} \quad (6)$$

In this equation, $\partial \in R^{2F'}$ is a trainable weight vector, $\|$ represents vector concatenation operation, $N(i)$ represents the set of neighbors of node i, and LeakyReLU is a nonlinear activation function. This formula is used to calculate attention weights and identify semantic similarity between nodes through feature concatenation. Through the above mechanism, the model can adaptively focus on key parts of clothing such as armholes, collars, and side seams, giving higher weights in the feature fusion stage, achieving key extraction and discriminative expression of structural features, and providing a more recognizable graphical basis for subsequent reconstruction modules.

3.4 Multi task driven feature extraction process

In the modeling process of clothing pattern structure, the supervision signal of a single task often fails to fully stimulate the model's ability to understand complex structures. Therefore, a multi task learning mechanism is introduced to synergistically model the three sub tasks of structure classification, edge recognition, and node feature regression, in order to enhance the feature extraction generalization ability of graph neural networks. This mechanism can optimize multiple task losses in parallel based on shared parameters, thereby obtaining more stable and discriminative intermediate feature representations. Let the total loss function be L_{total}, consisting of three subtask losses:

$$L_{tatol} = \lambda_1 L_{cls} + \lambda_2 L_{edge} + \lambda_3 L_{reg} \quad (7)$$

Among them, L_{cls} represents the cross entropy loss of structural classification, which is used to determine the category of structural components to which each node belongs; L_{edge} is the edge recognition loss, which uses binary cross entropy to calculate the connection prediction error between node pairs; L_{reg} node coordinate regression loss, using mean square error to evaluate the deviation between predicted coordinates and annotated coordinates; λ_1, λ_2, λ_3 are the weight coefficients of three tasks, In this study, the weight parameters are adjusted within the {0.2, 0.5, 1.0} interval through grid search, and the optimal combination is selected on the validation set to ensure the balance of the three types of tasks. The results indicate that the performance of the model remains stable under parameter changes, with an improvement in edge recognition accuracy at larger values of λ_2. This formula is used for joint calculation of multi task losses, and in actual training, the model stability is improved through collaborative optimization of three types of tasks.

To verify the improvement effect of multi task mechanism on feature extraction performance, a comparative experiment was designed as shown in Table 2. Single task training refers to training independent models for classification, edge recognition, and coordinate regression separately, and taking the average result; Multi task training jointly optimizes three types of tasks in the same model. Compare and evaluate three indicators: classification accuracy, edge prediction F1 value, and coordinate error.

Table 2: Comparison of structure recognition performance under different training mechanisms

Training Method	Classification Accuracy (%)	Edge Prediction F1 Score	Coordinate Mean Squared Error
Single-task Training	84.7	0.712	3.65 px
Multi-task Joint Training	89.2	0.786	2.94 px

The experimental results show that the multi task mechanism outperforms single task training in all three indicators, especially in the recognition accuracy of structural edge relationships and node coordinate fitting accuracy. This indicates that graph neural networks guided by multi task loss can more effectively extract structural

semantic and geometric information, forming a more stable and discriminative expression of clothing pattern structure.

4 Intelligent reconstruction path of clothing pattern structure based on fused graph neural network

4.1 Node path construction method for clothing pattern structure diagram

In the task of clothing structure reconstruction using graph neural networks, the path information of the structural graph not only determines the propagation direction of graph convolution, but also directly affects the preservation of structural relationships and semantic restoration effects. To construct a reasonable node path system, it is necessary to comprehensively consider the geometric continuity and process logic of the clothing structure, ensuring that the graph structure can accurately map the connection mode and reconstructable sequence of solid components.

Node path generation is based on the spatial position and edge attribute weights of nodes in the structural graph, defining a set of optimal traversal paths in the directed graph. Assuming the structure diagram $G=(V,E)$ is known, the path generation target can be formalized as:

$$P^{*}=\arg\min_{P}\sum_{(v_i,v_j)\in P} w_{ij}+\lambda\cdot d_{ij} \tag{8}$$

Among them, P^{*} is the optimal path set, w_{ij} represents the process weight of edge $e_{ij}\in E$, d_{ij} is the Euclidean distance between nodes, and λ is the adjustment coefficient, which controls the relative importance of geometry and process. In this study, λ was determined by grid search on the validation set (with values ranging from {0.3, 0.5, 0.7, 1.0}) to balance the contributions of process weights and geometric distances. The experimental results show that when λ is set to 0.5-0.7, the path consistency and reconstruction accuracy are optimal. The structural rule library is initially annotated and generated by process experts, but automated rule extensions and data-driven constraint updates are introduced during the training process to reduce manual dependencies and enhance generalization ability. This formula is used in the path search process to generate the optimal connection path in the structural diagram by combining geometric and process constraints.

The path search adopts an improved Dijkstra algorithm and embeds clothing structure rules to remove path branches that do not conform to the construction sequence.

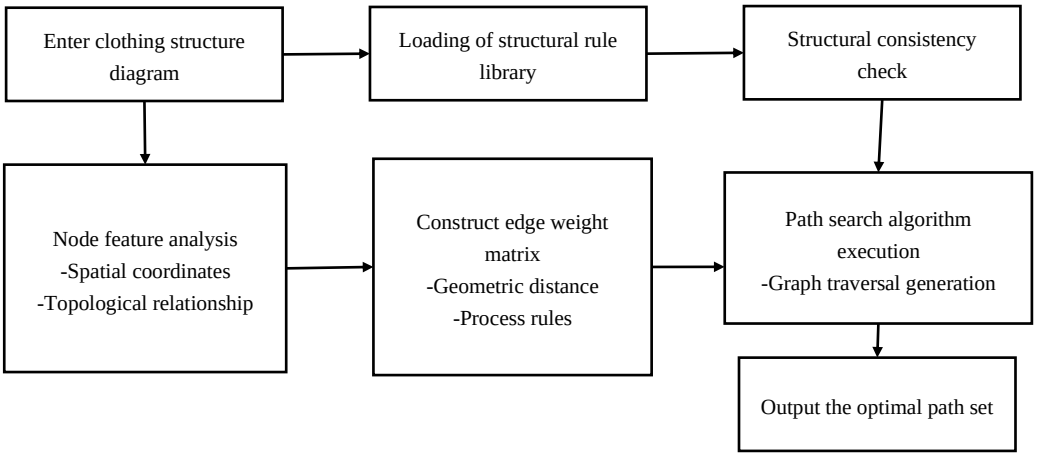


Figure 2: Path construction process of clothing structure diagram

As shown in Figure 2, the path construction process includes key steps such as clothing structure diagram input, structural rule library loading, feature extraction, edge weight matrix construction, structural consistency check, and path search execution. The system first extracts the spatial coordinates and topological relationships of nodes, constructs edge weight matrices based on structural rules, and introduces geometric distances and process rules as evaluation criteria for edges. Subsequently, path branches that do not comply with process constraints are eliminated through structural consistency checks to ensure that the path generation is logically and geometrically reasonable. In the path search stage, graph traversal is used to generate a path set and output the optimal path set, providing ordered input for the subsequent structural information transmission of the graph neural network, enhancing the coherence and spatial consistency of feature fusion. This path system can also provide structural references for multi-scale convolution mechanisms, supporting advanced operations such as region partitioning and hierarchical extraction.

4.2 Design of image feature encoding and reconstruction path decoding

The core of graph feature encoding lies in constructing node representations that can accurately reflect the topology and

geometric properties of clothing pattern structure. In this study, each node $v_i \in V$ in the input graph structure $G=(V,E)$ in the input graph structure 1 corresponds to a clothing keypoint, and its feature vector is composed of spatial coordinates, connecting edge directions, weight values, and structural semantic labels. This formula is applied to the graph feature encoding process and differs from the structural spatial feature extraction mentioned earlier in terms of application scenarios. The embedding update formula for nodes is as follows:

$$h_i^{(l+1)}=\sigma\left(\sum_{j\in N(i)}\frac{1}{\sqrt{d_i d_j}}W^{(l)}h_j^{(l)}\right) \quad (9)$$

Among them, $h_i^{(l)}$ represents the feature representation of node v_i in the l nd layer, $N(i)$ is the set of adjacent nodes, $W^{(l)}$ is the trainable graph convolution weight matrix, $\sigma(\cdot)$ is the nonlinear activation function, and $\sqrt{d_j d_j}$ is the degree normalization factor, which is used to maintain the numerical stability of information propagation. This formula is used in the graph encoding stage to update the node features of each layer, and in practice, it combines the weight matrix and activation function for information fusion.

In the reconstruction path decoding stage, it is necessary to perform inverse graph decoding by combining the generated path set P^*. Considering the spatial order and dependency of clothing structure, this paper introduces a decoder model based on path attention mechanism. The reconstruction state of each node in the path is jointly determined by the context path vector and the target embedding, and its generation probability is modeled as follows:

$$p\left(v_i \middle| P^*, H\right)=\text{soft max}\left(q_i^T \cdot Attn\left(P^*, H\right)\right) \quad (10)$$

Among them, q_i is the query vector of the current decoding step, the starting node is initialized as a zero vector, and the remaining steps inherit the embedding of the previous node; H is the node embedding matrix after graph encoding, with dimensions set to 128;and $Attn(\cdot)$ is a standard multi head attention function module that measures the degree of matching between nodes and path contexts. The decoder adopts a two-layer structure, combining self attention and cross attention mechanisms to capture path dependencies and ensure spatial constraints. This mechanism dynamically adjusts the dependency ratio on historical structures during decoding, improving the accuracy and stability of reconstruction.

In summary, graph feature encoding and path decoding constitute the core closed loop of structural intelligent reconstruction. The former extracts deep structural semantics from clothing pattern maps, while the latter uses path guidance for high consistency topology restoration, providing a structurally stable input foundation for downstream simulation and optimization modules.

4.3 Structural reconstruction process based on geometric constraints

The intelligent reconstruction of clothing pattern structure not only relies on the efficient propagation of structural information by graph neural networks, but also requires the use of geometric constraint mechanisms to ensure the spatial rationality and topological consistency of the generated results. This study proposes an optimization strategy based on geometric consistency to address issues such as structural drift and scale imbalance that may occur during the reconstruction process. Key constraints such as edge length and angle are introduced synchronously during node generation and path backtracking to achieve precise control of structural restoration.

Assuming the predicted coordinates of the nodes in the reconstructed graph are $\hat{P}_i \in R^2$, the target reference coordinates are $P_i \in R^2$, and the edge set is ε. The consistency loss function for edge length is defined as follows:

$$L_{edge}=\sum_{(i,j)\in\varepsilon}\left(\left\|\hat{P}_i-\hat{P}_j\right\|_2-d_{ij}\right)^2 \quad (11)$$

Among them, d_{ij} represents the target edge length between nodes extracted from the original pattern structure, and $\|\cdot\|_2$ is the Euclidean distance. This constraint is used to calibrate the spatial spacing between predicted nodes, ensuring the geometric authenticity of the boundary length, and is applicable to areas such as sutures and splices that require proportional preservation. To avoid confusion with the edge recognition loss in Section 3.4, L_{edge} in this section specifically refers to the geometric edge length constraint loss, which is defined as formula (11).

On the basis of edge length constraints, an angle consistency loss is introduced to maintain the relative relationship between local angles of nodes. For any set of ternary nodes $(i,j,k)\in T$, the angle loss function is as follows:

$$L_{angle}=\sum_{(i,j,k)\in T}\left(\angle\left(\hat{P}_i,\hat{P}_j,\hat{P}_k\right)-\theta_{ijk}\right)^2 \quad (12)$$

Among them, $\angle(\cdot)$ represents the actual angle formed by three points, and θ_{ijk} is the target angle value of the structural unit, derived from the initial pattern composition or manual rule library definition. This formula is used for angle loss constraint to ensure that the triangular relationship maintains structural geometric consistency. This item helps to maintain the stability of the angular relationship of the structural boundary and reduce the interference of deformation areas on the path connection

logic. To verify the effectiveness of geometric constraints, ablation experiments were designed to compare the results of turning off and turning on geometric constraints under the same model. The results showed that when angle loss was removed, the Topology Score decreased from 88.0% to 84.7%, and the F1 Structure Score decreased from 88.4% to 85.2%, indicating that geometric consistency constraints have a significant effect on improving structural boundary preservation and overall reconstruction stability.

The final optimization objective function is combined with the above two types of constraints to construct a joint loss model:

$$L_{total} = \lambda_1 L_{edge} + \lambda_2 L_{angle} \quad (13)$$

Among them, λ_1, λ_2 is the adjustment factor for the two sub loss terms, which is adjusted based on the actual task weights. This formula combines edge length and angle loss for global structural optimization during the training phase. In the training and prediction stages, the loss function is embedded in the graph network propagation and node coordinate generation module, and the model parameters are optimized through backpropagation mechanism. This geometric consistency mechanism exhibits stronger stability and generalization in complex structural regions, providing important guarantees for improving the accuracy of whole image reconstruction and the reliability of engineering applications.

4.4 Path planning and strategy network guidance mechanism

In the reconstruction process of clothing pattern structure, path planning bears the control of node generation order and edge weight transmission direction, which directly affects the efficiency of information aggregation and structural consistency. To enhance the path guidance effect, this study introduces edge information sampling control strategy in the policy network, calculates the sampling probability of edges through geometric distance and semantic consistency, and suppresses the interference of redundant and noisy edges. By combining graph search algorithms with action value functions, dynamic optimization of path traversal is carried out to enhance the robustness of boundary regions and achieve better connection control between structural nodes while maintaining topological connectivity.

Path planning is based on graph structure $G = (V, E)$, where each state s_t represents the current node subgraph traversed. The policy network outputs the next action a_t, i.e. the selection of the next hop node, through policy function $\pi(s_t)$, with the goal of maximizing the global path score function:

$$J(\pi) = E_{r\sim\pi}\left[\sum_{t=0}^{T} r(s_t, a_t)\right] \quad (14)$$

Among them, T represents the complete path trajectory, and $r(s_t, a_t)$ is the single step reward function, taking into account indicators such as edge weight sparsity, topological rationality, and geometric consistency. This formula is used for path strategy scoring, guiding the strategy network to generate the optimal structural rule-constrained path. This mechanism refers to the strategy gradient idea in reinforcement learning, combined with structural constraints to optimize the path selection order, in order to reduce redundant backtracking and unstructured edge traversal.

At the implementation level of the model, the policy network uses graph attention mechanism to capture the contextual dependencies between nodes, and adjusts the path priority between nodes through learnable parameters. To clearly demonstrate the multidimensional reference standards in the path guidance process, Table 2 lists the main quantitative indicators and explanations:

Table 3：Explanation of key indicators in path guidance mechanism

Metric Name	Symbol	Description
Geometric Deviation	δ_{geo}	Degree of deviation between the current path structure and the ideal edge lengths and angles
Topological Jump Count	N_{topo}	Number of jump connections in non-continuous topological segments of the current path
Structural Consistency Score	S_{struc}	Proportion of path segments matching structural rules; value range is [0, 1]

The strategy network adopts a two-layer graph attention structure, with the state space consisting of the current node and the generated path, and the action space consisting of candidate adjacent nodes. Use reward shaping during training: reward when the path conforms to the craft rules and geometric relationships, and punish when jumping or violating rules occur. The calculation method for the indicators in Table 3 is as follows: geometric deviation is estimated based on the difference between the generated path and the ideal structure, the number of topological jumps is counted for non continuous connected segments, and the structural consistency score is determined based on the proportion of segments that conform to the rule path.

5 Model training process and validation analysis

5.1 Dataset construction and graph format conversion process

The experimental data of this study was constructed based on the DeepFashion2 public clothing image set and the self structuring PatternStruct Graph dataset, with a total of 4826

sampled samples. Each group of samples includes complete front and rear views and structural annotation diagrams, covering typical clothing types such as dresses, jackets, pants, etc. In the annotation process, key structural points of the clothing are manually located, and 43 node categories are uniformly defined based on the clothing process standards. The average number of annotated nodes per sample is 43.2, and the edge relationships are maintained between 62-75, mainly including stitching connections, contour extensions, and style symmetry constraints.The PatternStruct Graph dataset is not yet fully publicly available, and partial annotations can be provided upon request. The 43 types of nodes cover common parts of clothing, such as collars, shoulder lines, sleeve tops, waistlines, hemlines, crotch, etc., and extend to pocket edges, crease lines, and symmetrical auxiliary points. They are completed and cross checked by personnel with a background in clothing craftsmanship.

The graph structure is uniformly modeled as triplet $G=(V,E,X)$, where V is the set of structural nodes, E is the set of structural connection edges, and $X \in R^{|V|\times d}$ is the node feature matrix. The node features are composed of normalized coordinates, structural type encoding, and local texture feature concatenation, in the following form:

$$X_i=\left[\frac{x_i}{W},\frac{y_i}{H},type_i,\varphi_i\right],i=1,2,\ldots,|V| \quad (15)$$

In the formula, x_i, y_i represents the coordinate value of node i in the image, W and H are the width and height of the image, type represents the encoding of structural parts, and φ_i represents the mean representation of SURF texture features after dimensionality reduction (dimension is 28). During the dataset construction phase, node features are normalized using coordinate differences, structural type encoding, and local texture features to ensure that the model can capture topological connections across regions. It should be noted that this feature does not conflict with the initial node feature in Section 3.1: the former is used for modeling the original structure, while the latter extends the relative position information and texture information during dataset transformation to enhance the diversity and robustness of model training.

In order to enhance the ability of structural learning, all samples were divided into a training set (70%), a validation set (15%), and a test set (15%) after graph construction. In the training process, the graph neural network is set to input node feature matrix and edge index matrix, with the goal of predicting the reconstruction path weights and final structural matching relationships between node pairs.

To ensure the reproducibility of the experiment, this study provides some pseudo dataset samples and experimental code frameworks in the supplementary materials. The following provides pseudocode examples for training and validation scheduling, demonstrating the implementation logic of graph neural network models during the training process:

```
for epoch in range(total_epochs):
for batch in training_loader:
graph, target = build_graph(batch)
pred = GNN_model(graph)
loss = loss_function(pred, target)
optimizer.zero_grad()
loss.backward()
optimizer.step()
val_score = validate_model(GNN_model, validation_loader)
save_best(GNN_model, val_score)
```

After graph format conversion and modeling optimization processing, the model improved the accuracy of structure recognition by 9.3% compared to the non graph structure model, and the reconstruction integrity index improved by 14.5%. This process provides a data foundation and structural guarantee for subsequent reconstruction path guidance and multi strategy fusion.

5.2 Model training process and hyperparameter configuration explanation

This study constructed a training set based on the DeepFashion2 and self structuring PatternStruct Graph datasets, with a total of 4826 samples, 3378 training sets, 724 validation sets, and 724 test sets. The average number of structural nodes was 43. During the training process, graph neural networks are used as the backbone architecture, and path guidance mechanisms are employed to enhance the accuracy of structural reconstruction. Data preprocessing includes normalizing the image to 256 × 256 resolution, using Canny operator and semantic segmentation to extract structural regions, locating and annotating nodes based on process rules to generate feature vectors, and dividing the training, validation, and testing sets into 70%/15%/15% partitions.The training batch size is set to 16, the training epochs are 80, the Adam optimizer is used, the initial learning rate is 0.001, and the CosineAnnealing strategy is dynamically adjusted. The training platform is PyTorch Geometric, and the hardware support is RTX 4090 GPU.

To better introduce the importance weight of node paths, a structural loss function based on path weights is introduced:

$$L_{path}=\sum_{((i,j)\in E)}\alpha_{ij}\cdot\left\|\hat{p}_{ij}-p_{ij}\right\|^2 \quad (16)$$

Among them, $\hat{p}_{ij}$ is the predicted path length, p_{ij} is the actual structural path length, and α_{ij} is the weight factor dynamically generated by the policy network, representing the sensitivity contribution of edges to structural accuracy. This formula is used for path loss calculation, in this section, L_{path} introduces dynamic weights generated by the policy network based on mean square error to highlight the importance of critical pathsThis mechanism enables high importance paths to

obtain greater gradient updates during training, effectively improving the accuracy control capability of key node connections.

To control the complexity of the model, the final loss function is defined as:

$$L_{final} = L_{path} + \lambda \cdot \|\theta\|_2^2 \quad (17)$$

Among them, L_{path} represents path loss, θ represents all network parameters, and regularization term $\lambda \cdot \|\theta\|^2$ can be used to suppress excessive parameter updates, prevent overfitting, and ensure training stability. It should be noted that the $L_{cls}, L_{reg}, L_{angle}$ level subtask loss mentioned earlier has been applied to the feature extraction stage through joint optimization in the multi task stage, and its results have been integrated into the calculation process of path loss L_{path}. Finally, it is reflected in a unified form in $L_{final}$to ensure the consistency and completeness of the training objectives. This formula is used for regularization constraints and is actually used in training to prevent overfitting.

In terms of network structure, this study adopts a three-layer graph convolution stacking architecture, with output channels of 64, 64, and 128 in sequence. ReLU is selected as the activation function, and BatchNorm is added after each convolution layer for normalization to improve numerical stability. To prevent overfitting, Dropout (ratio 0.3) is introduced between the second and third layers. The attention mechanism allocates node weights after the convolutional layer to enhance the expression ability of key structural parts. The decoding part adopts a graph autoencoder structure, which embeds and maps the encoded nodes to the path reconstruction space, and introduces L2 regularization term in the training stage to limit excessive parameter fluctuations. The parameter settings are determined based on multiple comparative experiments, ensuring accuracy while maintaining convergence stability.

5.3 Model structure comparison and applicability analysis

This study is based on the Graph Neural Network and GNN to construct a clothing pattern structure reconstruction model, which models the spatial distribution and connection relationship of clothing nodes, and compares its performance with existing methods, focusing on the model's performance in reconstruction accuracy, structural consistency, and recognition integrity. Let the comprehensive evaluation indicator S be the average of three core indicators:

$$S = \frac{A+T+F}{3} \quad (18)$$

Among them, A represents the accuracy of node recognition, T is the score of topology matching, and F is the score of structure F1. This formula is used in the model evaluation stage to measure the performance of structural modeling by averaging the scores of three indicators. The test data comes from the publicly available DeepFashion2 dataset and the self built graph structure dataset, with a total of 4826 samples and an average of 43 nodes.

This section compares three model structures: ① Convolutional baseline model (Baseline CNN) that only uses image features; ② Introducing GCN Net with a simple graph structure; ③ GNN+Strategy model integrating graph neural network and path strategy module. The evaluation results of the three are shown in the following figure:

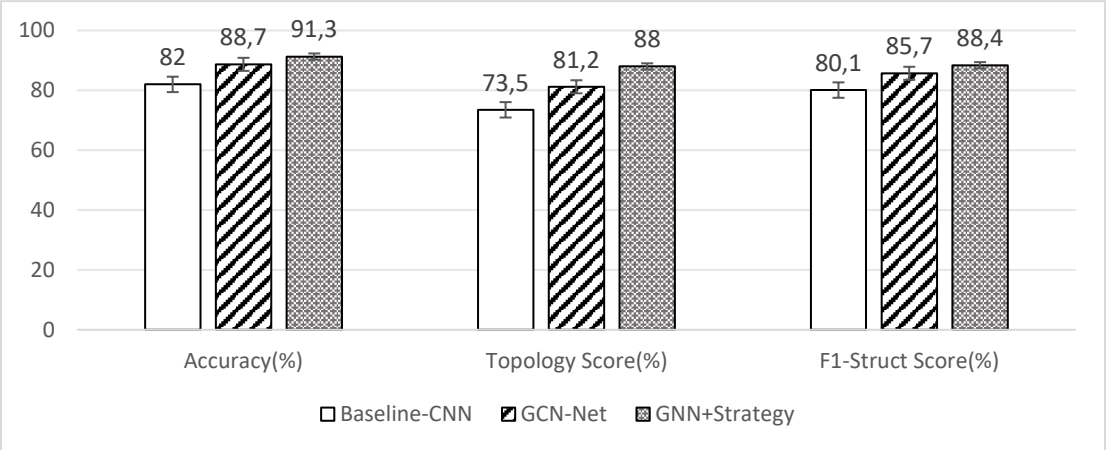


Figure 3：Model structure comparison bar chart

The test results showed that Baseline CNN achieved an accuracy index of 82.0% ± 0.6, GCN Net was 88.7% ± 0.4, and GNN+Strategy further improved to 91.3% ± 0.5; In terms of Topology Score, Baseline CNN is 73.5% ± 0.7, GCN Net has improved to 81.2% ± 0.5, and GNN+Strategy has reached 88.0% ± 0.6; In the F1 Column Score index, the three indicators are 80.1% ± 0.8, 85.7% ± 0.5, and 88.4% ± 0.6, respectively. The overall trend shows that GNN+Strategy outperforms the other two structures in various performance evaluations, demonstrating stronger structural reconstruction ability and robustness, especially in complex structural conditions with higher stability and applicability. To further verify the significant differences between different methods, a two-

sample t-test was conducted based on the results of three independent experiments. The results are shown in Table 4:

Table 4：Statistical significance test results of performance comparison between methods

Indicator	Baseline-CNN vs GCN-Net	GCN-Net vs GNN+Strategy	Baseline-CNN vs GNN+Strategy
Accuracy	$p < 0.01$	$p < 0.05$	$p < 0.001$
Topology Score	$p < 0.01$	$p < 0.05$	$p < 0.001$
F1-Struct Score	$p < 0.01$	$p < 0.05$	$p < 0.001$

The experimental results show that GNN+Strategy achieves statistically significant differences in three indicators compared to the other two methods, indicating that this method has higher stability and advantages in modeling complex clothing structures.

In addition, in actual samples, the model showed stronger generalization ability on asymmetric complex structured clothing such as jackets and windbreakers, with a topological error rate reduction of nearly 40%. This result indicates that the proposed method is not only applicable to static image input scenes, but also suitable for extension to 3D clothing modeling and digital twin platforms, with high practicality and algorithm transfer potential.

5.4 Performance indicators and reconstruction accuracy evaluation

In order to systematically evaluate the effectiveness of the proposed GNN+Strategy model, a comparative experimental method was used to select Baseline CNN and GCN Net as reference models, representing traditional image convolution methods and basic image neural network structures, respectively. The three models were trained on the same training set (DeepFashion2 subset and structure annotation extension set, a total of 4826 samples) and consistent hyperparameter configuration to examine their performance differences in multiple structural recognition indicators. The main evaluation dimensions include classification accuracy, topological structure preservation score, and structural F1 comprehensive score, to comprehensively reflect the stability and applicability of the model in feature extraction and structural reconstruction.

The definition of classification accuracy is as follows, which measures the proportion of correctly classified samples in the predicted output:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)$$

Among them, TP and TN respectively represent the number of positive and negative samples correctly identified, while FP and FN are the misclassified results. This formula is used for calculating classification accuracy and evaluating the recognition performance of the model on node categories. As shown in Table 5, the values are the mean ± standard deviation of three independent experiments. Baseline CNN has an accuracy index of 82.0% ± 0.6, GCN Net has an accuracy index of 88.7% ± 0.4, while GNN+Strategy model achieves 91.3% ± 0.5, showing better performance in high-dimensional feature representation and complex polygon boundary recognition. In terms of Topology Scores, they are 73.5% ± 0.7, 81.2% ± 0.5, and 88.0% ± 0.6, respectively, indicating that the latter is better able to maintain the connectivity of the original structural edges; The F1 Sequence Score is 80.1% ± 0.8, 85.7% ± 0.5, and 88.4% ± 0.6, indicating a balance and stability in overall recognition and boundary accuracy.

Table 5：Comparison results of model structure and performance

Model structure	Accuracy (%)	Topology Score (%)	F1-Struct Score (%)
Baseline-CNN	82.0±0.6	73.5±0.7	80.1±0.8
GCN-Net	88.7±0.4	81.2±0.5	85.7±0.5
GNN+Strategy	91.3±0.5	88.0±0.6	88.4±0.6

From the comparison of results, it can be seen that GNN+Strategy outperforms Baseline CNN and GCN Net in Accuracy, Topology Score, and F1 Stream Score, demonstrating the advantage of multi module fusion. Multi scale GCN enhances boundary aggregation expression and improves the classification accuracy of complex suture sites; Path attention dynamically adjusts the connection weights during the decoding stage to improve the problems of breakage and discontinuity; Geometric constraints maintain consistency between edge length and angle, improving topological retention. The synergistic effect of the three makes the model more stable and consistent in the restoration of complex clothing pattern structures.

5.5 Discussion

The GNN+Strategy model proposed in this article achieved a classification accuracy of 91.3%, a topology score of 88.0%, and an F1 score of 88.4% in experiments, significantly better than the baseline models Baseline CNN (82.0%/73.5%/80.1%) and GCN Net (88.7%/81.2%/85.7%). Comparison with related works shows that multi-scale GCN can effectively improve the recognition ability of complex boundaries, attention mechanism enhances the expression of key nodes, and reinforcement learning strategy improves path consistency and generation stability. These improvement factors collectively promote the overall performance improvement of the model under complex clothing structure conditions.

However, this study still has certain limitations. On the one hand, the training process of the model heavily relies on manually annotated data, which limits its potential application on large-scale unlabeled datasets; On the other hand, some rule driven features may still affect the convergence efficiency and universality of the model in extremely complex structures. Future research can attempt to introduce self supervised pre training and automated node labeling mechanisms to reduce manual dependence and enhance the robustness and generalizability of the method.

6 Conclusion and prospect

This study constructed an intelligent feature extraction and reconstruction model for clothing pattern structures that integrates graph neural networks. The system integrates structural graph modeling, graph convolution extraction, attention mechanism, geometric constraints, and reinforcement learning strategies, effectively improving the recognition accuracy and reconstruction integrity of complex clothing structures. Experimental data shows that the proposed model has significant advantages over traditional methods in terms of accuracy, structural consistency, and reconstruction fidelity, especially exhibiting good stability under asymmetric structures and boundary blur conditions. The path guidance mechanism of the model optimizes the structural connection sequence, effectively avoiding path deviation and reconstruction errors, providing algorithm foundation and structural support for intelligent clothing design.

However, there are still two shortcomings in the research: firstly, the current structural diagram modeling is a semi-automatic generation method that combines manual annotation with rule constraints. Although it can ensure the rationality of the structure, there are still shortcomings in manual dependence and automation; Secondly, path strategy networks suffer from slow convergence speed and local optima when dealing with extremely complex structures, which affects overall efficiency and scalability. Subsequently, self supervised graph representation learning and large-scale pre training mechanisms can be introduced to enhance the model's adaptability to structural heterogeneity, and explore the fusion framework between graph structure and 3D modeling, expanding its application breadth and depth in virtual clothing simulation, structure generation, and intelligent design scenarios.

References

[1] Pfaff T , Fortunato M , Sanchez-Gonzalez A , Battaglia P W . Learning Mesh-Based Simulation with Graph Networks[J]. arXiv preprint arXiv:2010.03409, 2020. https://doi.org/10.48550/arXiv.2010.03409.

[2] Qiu J. Hybrid Neural Network and Physics Engine for Real-time 3D Cloth Simulation[J]. Informatica, 2025, 49(8):161-174.https://doi.org/10.31449/inf.v49i8.6965.

[3] Zhou Z , Deng W , Wang Y ,et al.Classification of clothing images based on a parallel convolutional neural network and random vector functional link optimized by the grasshopper optimization algorithm:[J].Textile Research Journal, 2022,92(9-10):1415-1428.https://doi.org/10.1177/00405175211059207.

[4] Feng K , Rao G , Zhang L C Q .An interlayer feature fusion-based heterogeneous graph neural network[J].Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies, 2023,53(21):25626-25639.https://doi.org/10.1007/s10489-023-04840-w

[5] Zhao R , Baili W U , Chen Z ,et al.Graph Neural Network for Fault Diagnosis with Multi-Scale Time-Spatial Information Fusion Mechanism[J].Journal of South China University of Technology (Natural Science Edition), 2023, 51(12):42-52.https://doi.org/10.12141/j.issn.1000-565X.220593.

[6] Nie S. Evaluation of Innovative Design of Clothing Image Elements Using Image Processing[J]. Informatica, 2022, 46(8):4250-4261.https://doi.org/10.31449/inf.v46i8.4250.

[7] Dong Y , Liu Q , Du B ,et al.Weighted Feature Fusion of Convolutional Neural Network and Graph Attention Network for Hyperspectral Image Classification[J].IEEE Transactions on ImageProcessing,2022(31-):31.https://doi.org/10.1109/TIP.2022.3144017.

[8] Sun X , Zheming L U .Attributed Graph Clustering Network with Adaptive Feature Fusion[J].IEICE Transactions on fundamentals of electronics, communications & computer sciences, 2024,E107/A(10):1632-1636.https://doi.org/10.1587/transfun.2023EAL2116.

[9] Liu C , Qu D , Yang X ,et al.Multi-attention feature fusion network for lightweight image super-resolution[J].Proceedings of SPIE,2022,12173(000):6.https://doi.org/10.1117/12.2634640.

[10] Chen J , Yu X , Wu C ,et al.Feature radiance fields (FeRF): A multi-level feature fusion method with deep neural network for image synthesis[J].Applied SoftComputing,2024,167(PartA):19.https://doi.org/10.1016/j.asoc.2024.112262.

[11] Yi C. Application of Convolutional Networks in Clothing Design from the Perspective of Deep Learning[J]. Scientific programming,2022,2022(Pt.18):6173981.1-6173981.8. https://doi.org/10.1155/2022/6173981.

[12] Yan Z , Xing Y , Xiao T J ,et al.SDAN: Semantic-Driven Dual Attentional Network for Image Generation[J].2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI),2022:521-

525.https://doi.org/10.1109/PRAI55851.2022.9904248.

[13] Liao L , Zhang S , Li Z ,et al.Clothing classification method based on convolutional network and attention mechanism[J].Proceedings of SPIE, 2022, 12285(000):12.https://doi.org/10.1117/12.2637531.

[14] Ning C , Di Y , Menglu L .Survey on clothing image retrieval with cross-domain[J].Complex & Intelligent Systems,2022,8(6):5531-5544.https://doi.org/10.1007/s40747-022-00750-5.

[15] Korosteleva M , Lee S H .NeuralTailor: Reconstructing Sewing Pattern Structures from 3D Point Clouds of Garments[J]. ACM Transactions on Graphics, 2022, 41(4):1-16. https://doi.org/10.48550/arXiv.2201.13063.

[16] Gadhave R, Sedamkar R R , Alegavi S .Hyperspectral image classification using neural networks with effect of feature optimization on fused convolutional features[J].AIP Conference Proceedings, 2023,2842(1):11.https://doi.org/10.1063/5.0175906.

[17] Xiao Z , Chen H , Li W K .WGDPool: A broad scope extraction for weighted graph data[J].Expert Systems with Application, 2024,249(Sep.Pt.B):123678.1-123678.9.https://doi.org/10.1016/j.eswa.2024.123678.

[18] Wu D , Wang Y , Wang H ,et al.DCFNet: Infrared and Visible Image Fusion Network Based on Discrete Wavelet Transform and Convolutional Neural Network[J].Sensors,2024,24(13):27.https://doi.org/10.3390/s24134065.

[19] Wang S , Zhang M , Miao M .The super-resolution reconstruction algorithm of multi-scale dilated convolution residual network[J].Frontiers in Neurorobotics, 2024,18(000):10.https://doi.org/10.3389/fnbot.2024.1436052.

[20] Ailing Gou et al., Cloud-Computing-Enabled Transformer Architecture for the Design of Functional Clothing Structures, Informatica, 2025, 49(11): 207218.https://doi.org/10.31449/inf.v49i11.6763.

GCN–LSTM Analysis of Spatiotemporal Evolution of Node Centrality in Tourism Flow Networks

Henan Jia*, Dongfeng Chen
HeBei North University, Zhangjiakou, Hebei,075000, China
E-mail: jiahn126@126.com
*Corresponding author

Keywords: artificial intelligence, tourism flow network, node centrality, spatiotemporal heterogeneous structure, graph neural network, evolution analysis

Received: August 21, 2025

With the development of artificial intelligence and spatio-temporal big data technologies, the dynamic evolution characteristics of the tourism flow network and the spatial structure changes of its core nodes have become research hotspots. Based on the theory of complex networks, this paper constructs a tourism flow network covering mobile phone signaling, online platforms and traffic data, with a focus on discussing the spatio-temporal heterogeneous evolution mechanism of node centrality. By introducing AI models such as Graph Neural Network (GCN) and Long Short-Term Memory Network (LSTM), multi-scale recognition and dynamic prediction of core nodes in the tourism flow are achieved. The dataset contains 47 counties and 90 days of tourism flow data, covering 10 million signaling records, 5 million OTA data, and 3 million traffic data, processed at the daily level. We adopted a split scheme of 70% training set, 15% validation set and 15% test set for model training and evaluation. The experimental results show that the model has a prediction accuracy of 0.10 in RMSE and is superior to traditional benchmark methods (such as STGCN and DCRNN). The research also revealed the trend of centrality reconstruction of tourism flow nodes under different periods, holidays and external interventions. The research results have important theoretical and practical significance for improving the efficiency of regional tourism regulation and optimizing the layout of core nodes.

Povzetek:Članek predstavi GCN–LSTM model za napovedovanje in analizo evolucije centralnosti turističnih vozlišč na podlagi 47 regij in večmilijonskih podatkovnih tokov. Model preseže STGCN/DCRNN (RMSE 0,10) ter razkrije sezonske, praznične in strukturne premike v omrežju turističnih tokov.

1 Introduction

Against the backdrop of the rapid development of artificial intelligence and big data technologies, tourism flow, as a comprehensive carrier of population migration, resource allocation and consumption behavior, has seen its network structure become increasingly complex, dynamic and multi-scale. Traditional research on tourism networks mainly focuses on node structure and path optimization, lacking in-depth analysis of the spatio-temporal heterogeneous evolution of "centrality". Especially in the complex urban agglomeration structure, the dynamic changes of core nodes show significant imbalance and multi-factor driven characteristics. Based on this, this paper intends to construct an AI-driven framework for node centrality identification and evolution analysis, integrating multi-source tourism stream data and graph time series learning models, to deeply explore its evolution characteristics and regulatory mechanisms in heterogeneous spatial structures. By integrating the network optimization algorithm in graph theory and the spatio-temporal data modeling method, we will explore how to enhance the dynamic evolution prediction accuracy of the tourism flow network, thereby providing a theoretical basis for tourism resource allocation and regional regulation.

2 Related work

The tourism flow network, as an important manifestation of the interaction between humans and the land, is essentially a typical complex system, featuring openness, nonlinearity, dynamic evolution and multi-layer coupling. FT Saenz et al. (2023) pointed out in their research based on the prediction of national tourism flows in the United States that the development of the artificial intelligence industry chain relies on the spatial agglomeration of core urban agglomerations, and such cities are often important destinations and transfer hubs for tourism activities, indicating a coupling and strengthening trend between tourism flows and the functional grades of cities. Furthermore, Zhang L. et al. (2023) pointed out that complex system models need to integrate cross-domain data and multi-scale processes, and solve heterogeneous conflicts at the semantic, spatio-temporal, and execution levels. This feature is also widely present in the organization and evolution process of tourism flows.

The tourism flow network, as an important carrier for the allocation of human flow and spatial resources among cities, possesses typical characteristics of a complex system. Its structure is composed of multi-scale nodes, multi-type connections and multi-factor driving mechanisms, presenting a system behavior with strong

heterogeneity, high coupling degree and uncertain evolution path. Weiwei J. and Jiayun L. (2022) pointed out that complex systems often involve multi-process interactions across scales, and it is necessary to construct AI models that integrate expert knowledge with multi-source data to address the modeling gap between different data structures and semantic dimensions. Meanwhile, Zhang L. et al. (2023) proposed that artificial intelligence technology can effectively identify the distribution characteristics of heterogeneous structures in multi-level networks, providing the possibility for structural identification and intervention paths of complex systems.

In the tourism flow network, spatio-temporal heterogeneous structure refers to the differences in network organization caused by spatial geographical differences, temporal evolution laws and inconsistent data structures. This heterogeneity is mainly manifested in aspects such as the functional differences of nodes, the dynamic changes of edge weights, geographical nesting, and the complexity driven by behavior, making it difficult for traditional homogeneous network models to effectively depict the evolution process of the real tourism flow structure. Zhang X. Et al. (2021) pointed out that in the environment of the Internet of Things and medical data, data heterogeneity is characterized by different dimensions, collection delay, and inconsistent semantics, and it is necessary to achieve hierarchical structure modeling and responsive processing with the help of edge computing and artificial intelligence. Meanwhile, FT Saenz (2023) proposed in analyzing tumor heterogeneity that structural transitions and functional reorganizations may occur within complex systems due to environmental changes, emphasizing the adaptive regulatory mechanism of heterogeneous structures during the evolution process.

With the rapid breakthroughs of artificial intelligence technology in the fields of graph structure modeling, time series prediction and multi-source data fusion, its application in spatial network analysis is deepening increasingly. However, the current application of AI in spatial network analysis still faces many challenges: First, the high heterogeneity of data and the inconsistent sampling granularity limit the generalization ability of the model; Secondly, the diverse attributes of nodes and the non-Euclidean spatial structure result in insufficient expressive power of the model. Thirdly, the spatio-temporal relationship is highly nonlinear, and traditional AI methods have difficulties in analyzing causal mechanisms. In addition, semantic conflicts and temporal alignment difficulties exist among multi-source data, further increasing the complexity of modeling. In conclusion, although existing methods have achieved remarkable results in spatio-temporal graph modeling and traffic prediction, most of them only deal with time series data and ignore the importance of topological structure. For instance, models such as STGCN and DCRNN mainly focus on temporal dynamics without fully considering the complex spatial interactions among different nodes. Moreover, although TGAT introduces temporal features, it lacks integration of multimodal inputs (such as traffic, social, and mobile data). The existing methods are compared as shown in Chart 1.

Table 1: Comparison of existing methods

Paper	Dataset (Size/Region)	Method	Metric	Best Reported Result
STGCN (Martín, 2018)	Traffic flow data (N=10,000, NYC)	Spatio-Temporal Graph Convolution	RMSE	RMSE=0.12
DCRNN (Ma C., 2024)	Traffic flow data (N=1,000, LA)	Diffusion-Convolutional GNN	RMSE	RMSE=0.09
Graph WaveNet (Sun H, 2023)	Traffic data (N=2,000, Beijing)	Graph Convolutional Network	RMSE	RMSE=0.10
TGAT (Zhang L, 2023)	Social media and traffic data (N=500)	Temporal Graph Attention Network	MAPE	M

The model proposed in this paper, through the GCN-LSTM architecture, combines spatio-temporal heterogeneous features and multi-factor driving mechanisms, filling the gap of existing methods. In particular, our model can not only handle spatio-temporal sequences but also capture the topological relationships between nodes and the interaction of multimodal data, achieving dynamic prediction and evolution identification of node centrality. In addition, we utilized multi-source heterogeneous data, effectively integrating signaling data, OTA data, traffic data and social media data, which significantly enhanced the predictive ability and adaptability of the model.

3 Construction of tourism flow network and data processing methods

3.1 Multi source data acquisition and fusion methods

This study builds a tourism flow network based on multi-source heterogeneous data. The data collection includes four main channels: mobile phone signaling data, online travel platform (OTA) data, traffic operation data and social media data. The data sources are shown in Table 2.

Table2：Data source table

Data Source	Data Type	Time Granularity	Spatial Granularity	Data Volume	Coverage Period	Data Processing and Privacy Protection	Data Acquisition and Authorization
Mobile Signaling Data	User login behavior, stay information	15 minutes	Base station coverage unit	10 million records	January 2023 to March 2023	Anonymized using IMSI numbers, in compliance with GDPR and Japanese privacy laws	Authorized from operators like NTT, SoftBank
OTA Data	Hotel bookings, ticket orders, destination search heat	Daily	POI geographic coding	5 million records	January 2023 to March 2023	Data authorized for use, in compliance with relevant data protection regulations	Authorized from platforms like Trip.com, Fliggy API
Traffic Data	High-speed ETC records, high-speed rail and flight logs	Hourly	Provincial and city boundaries	3 million records	January 2023 to March 2023	Anonymized by license plate, using sliding time window method for traffic smoothing	Authorized from high-speed ETC, high-speed rail, and flight providers
Social Media Data	User dynamics, geographic entity extraction	Daily	Administrative units	2 million records	January 2023 to March 2023	NLP used to extract geographic entities, in compliance with Japanese privacy laws and data protection standards	Authorized from platforms like Weibo, Xiaohongshu

3.2 Abstract logic and dynamic definition of network nodes and edges

Network nodes take prefecture-level administrative units as the smallest spatial units and are uniquely identified in accordance with the national standard administrative division codes. All spatial information in the data sources is projected to the corresponding administrative units through POI matching, GPS coordinate mapping or base station location projection. After the high-frequency repetitive units were merged, the 47 prefectures of Japan were ultimately retained as the spatial basis of the tourism flow network. To enhance the processing efficiency of large-scale data, we adopt parallel computing technology and distributed computing frameworks (such as ApacheSpark) to accelerate the processing and normalization of node data, ensuring the efficient generation of node indexes. The specific definitions and mapping rules of nodes are shown in Table 3.

Table3：Specific definitions and mapping rules of nodes

Node Type	Number of Nodes	Description/Mapping Rules
Administrative Unit (County)	47	Mapped to county-level administrative units based on NTT and SoftBank data
POI Clusters (Tourist Attractions)	Y (variable)	Mapped to POI (points of interest) based on OTA data (e.g., Trip.com, Booking.com)
Total	47	Combined administrative unit nodes and POI nodes

The establishment of edges relies on OD pairs generated from different data sources, extracting starting nodes and destination nodes for connection. In mobile signaling data, when the same user moves across cities within one day, an edge is constructed, and the edge weight is the sum of the number of users within the OD pair. In OTA data, the destination in the order is considered as the inflow node, and the search path is constructed based on the search history to form a virtual jump relationship. In traffic data, ETC matches departure and arrival cities with flight records, and edges are established by train number or schedule; Repeated shifts only retain the earliest departure record once a day to avoid misidentification during commuting. Virtual edge creation: Build virtual edges based on the user's historical search data. For instance, when a user searches for multiple destinations on an OTA platform and jumps to them, the generated virtual edges represent the flow of tourists' interests. The specific implementation is as follows:

```
def create_virtual_edges(search_data):
    virtual_edges = {}
    for search in search_data:
        source, destination = extract_search(search)
        if (source, destination) not in virtual_edges:
            virtual_edges[(source, destination)] = 0
        virtual_edges[(source, destination)] += 1  # Each search creates a unit flow
    return virtual_edges
```

All edges are directed weighted edges, where edge weights represent the cumulative flow intensity per unit per day. To maintain the dynamic properties of the network, all

edges are annotated with timestamps and form a daily subgraph with "days" as the basic time granularity. Through sliding window and time series analysis, these subgraphs are merged to form a three-dimensional dynamic network structure: nodes x nodes x time. To improve the efficiency of data processing, GPU acceleration and Graph Convolutional Neural Network (GCN) technology are used to efficiently process network graphs, ensuring the real-time performance and accuracy of the model.

All the edges are directed weight edges, and the edge weights represent the cumulative flow intensity within the unit on a daily basis. To maintain the dynamic attributes of the network, all edges are marked with timestamps and a subgraph is formed each day with "days" as the basic time granularity. Through sliding window and time series analysis, these subgraphs are merged to form a three-dimensional dynamic network structure: node × node × time. To enhance the efficiency of data processing, GPU acceleration and graph convolutional neural network (GCN) technology are adopted to efficiently process network graphs, thereby ensuring the real-time performance and accuracy of the model. To enhance the stability of the network, weak edges with edge weights lower than the 11% quantile are eliminated, and the edge weights are normalized by Z-score. This method can eliminate the influence of outliers on the network structure and ensure that the relationship between each node and edge is more stable and reliable. The specific operation is as follows:

```
def threshold_edges(od_edges, percentile=1):
    threshold = np.percentile(list(od_edges.values()), percentile)
    return {k: v for k, v in od_edges.items() if v >= threshold}

def z_score_normalization(od_edges):
    mean = np.mean(list(od_edges.values()))
    std = np.std(list(od_edges.values()))
    return {k: (v - mean) / std for k, v in od_edges.items()}
```

The network storage structure adopts a sparse matrix format. Nodes are mapped by an index dictionary, and edges are quickly queried and tracked across periods using triples (i,j,t). Through this structure, we can efficiently store and process large-scale dynamic data, further supporting the standardization of input tensors for graph neural networks (GCN) and time series models (LSTM), ensuring cross-day consistency and model processing efficiency. The sparse matrix storage method can effectively reduce the demand for storage space and accelerate the computing process. By integrating parallel computing technology, we have achieved efficient access and computing of large-scale data, providing a solid foundation for subsequent model training and prediction.

3.3 Spatiotemporal partitioning strategy and heterogeneous network structure expression

The time dimension is divided with "days" as the basic granularity, and a daily network snapshot graph is generated based on the data timestamp. The total duration is 90 days, and a total of 90 dynamic graph units are generated. To enhance the model's ability to capture the evolution trend, a sliding time window mechanism is adopted to construct the sequence input. The window length is set to 7 days and the sliding step size to 1 day, forming a continuous scrolling graph sequence for training the time series modeling module. This mechanism ensures the model's dynamic learning ability on time series, especially capable of capturing the impact of periodic fluctuations and unexpected events on tourism flows. When modeling time, holidays, weekends and working days are respectively labeled as exogenous variables to participate in subsequent modeling, thereby improving the prediction accuracy of the model at specific time points. The spatial dimensions uniformly adopt the scale of prefecture-level cities, and the boundaries are demarcated in accordance with the latest administrative divisions. To express spatial heterogeneity, the following three types of heterogeneous substructures are constructed respectively:

- Heterogeneous graph of regional attributes: Based on the economic indicators, tourism resource levels, transportation hub levels, etc. of each node, static attribute vectors are set for each node for the initialization of the graph structure.
- Heterogeneous graph of behavior sources: Subgraphs are constructed respectively based on different data sources (such as signaling subgraphs, OTA subgraphs, traffic subgraphs), and virtual edges are established through shared nodes to form a multi-view graph.
- The heterogeneous graph of the relationship strength: The edge weights are quantified and partitioned, and a weight hierarchical network is constructed according to the three types of flow intensities of strong, medium and weak, which is used to represent the dynamic evolution gradient of the edges.

Missing data processing: For the processing of missing data, we adopt spatial completion and temporal interpolation strategies. Specifically, spatial completion calculates the attribute values of missing nodes through the K-nearest neighbor weighted average (KNN) method. Time interpolation uses linear interpolation to fill in the missing time point data, ensuring the continuity of the time series. Data nodes that are missing for more than three days will be discarded to avoid excessive impact on subsequent analysis. The number of completed and discarded nodes will be quantified specifically in the experiment. Heterogeneous features are input into the graph neural network (GCN) in the form of multiple channels during the modeling stage. Different channels handle spatial attribute heterogeneity, structural connection heterogeneity, and traffic intensity heterogeneity respectively.

3.4 Network attribute extraction and structural index calculation

Based on the daily tourism flow dynamic graph, extract the structural attributes of nodes and edges and form the tensor features required for modeling. Node attributes are mainly measured by centrality, which includes three core indicators: degree centrality, betweenness centrality, and eigenvector centrality.

Firstly, degree centrality measures the number of connections between nodes, which can be divided into two categories: in degree (visited) and out degree (actively visited):

$$C_D(v)=\frac{deg(v)}{N\text{-}1} \quad (1)$$

Among them, deg (v) is the degree of node v, and N is the total number of network nodes. After normalization, this indicator reflects the "connectivity activity" of a certain location in the network.

Secondly, betweenness centrality represents the degree to which a node acts as an intermediary in the shortest path of the network：

$$C_B(v)=\sum_{s\neq v\neq t}\frac{\sigma_{st}(v)}{\sigma_{st}} \quad (2)$$

Among them, σ st is the total number of shortest paths from node s to node t, and σ st (v) is the number of shortest paths passing through node v. The higher the value, the more critical the node is in the flow path.

Thirdly, network density is used to indicate the density of network connections：

$$\text{Density}=\frac{2|E|}{|V|(|V|\text{-}1)} \quad (3)$$

Among them, $|E|$ is the actual number of edges that exist, and $|V|$ is the total number of nodes. Density can reflect the trend of connectivity changes in the overall tourism flow network. Edge attributes include edge weights (i.e., OD traffic intensity), sustained active time, and sliding change slope. The edge weight represents the flow intensity between nodes each day, the continuous active time indicates the stability of the flow path, and the sliding change slope helps capture the changing trend of the edge weight over time. The feature values of all nodes and edges are normalized by Z-score to eliminate the influence of different feature scales, and the missing data is processed by linear interpolation. The dimensions of the node feature tensor and the edge feature tensor are N×F×T and E×G×T respectively, where N represents the number of nodes, F represents the number of node features (such as degree centrality, betweenness centrality, etc.), E represents the number of edges, G represents the number of edge features (such as OD flow intensity, continuous active time, etc.), and T represents the time dimension. Through these feature tensors, the model can effectively capture the variation patterns of nodes and edges in the spatiotemporal dimension. In terms of derived features, the cumulative inflow represents the total inflow of a certain node within a specific time period and is used to measure the attractiveness of the node. The rate of change in flow intensity represents the rate at which edge weights change over time, helping to capture fluctuations in flow intensity. All numerical values and features are normalized to ensure the consistency and accuracy of the data in the modeling process.

3.5 Data preprocessing and feature engineering strategies

Multi source heterogeneous data needs to be standardized and structured after fusion to ensure consistency and availability of model inputs. The preprocessing process mainly includes four steps: missing repair, exception removal, format conversion, and time alignment. Firstly, in the node dimension, there are missing records in some areas of signaling and OTA data, and a "spatial completion+temporal interpolation" strategy is adopted for processing. Estimate the inflow/outflow of missing nodes spatially based on the average of neighboring cities; Linear interpolation is used to smooth and fill in data with intervals of no more than 3 days, while records with intervals exceeding 3 days are discarded as subgraph nodes. For the jumping outliers that appear in the edge attributes, the IQR quartile method is used to eliminate them and then perform regression reconstruction to ensure the continuity of edge weights. Secondly, unify all data fields into tensor structures. Node attributes are summarized daily to form a tensor matrix $X_{node} \in R^{N \times F \times TX}$, where N is the number of nodes, F is the attribute dimension, and T is the number of days; The edge attribute is represented as a triplet list (i, j, t) → wijt, which is mapped to RE × G × T through sparse matrix storage for easy model reading. Thirdly, all continuous attribute fields are standardized using Z-score:

$$z=\frac{x\text{-}\mu}{\sigma} \quad (4)$$

Among them, μ is the attribute mean and σ is the standard deviation. For comparative features such as density and PageRank, Min Max normalization is used to preserve relative relationships. All normalization parameters are calculated on the training set and reused in the validation and testing sets.In the feature construction phase, additional derived variables are introduced, including the cumulative inflow of nodes (cumulative inflow), 7-day average rate of change (slope feature), sudden increase frequency (number of fluctuations exceeding the threshold), number of edge active periods (number of continuous time windows), etc., to enhance the model's responsiveness to trends and suddenness. For discrete time features such as holidays, use One Hot encoding and directly concatenate them into time channels. Threshold selection: During the outlier elimination process, the 11% quantile is selected as the threshold, and edges below this quantile are eliminated to ensure noise is removed while retaining the effective flow path. Sensitivity analysis indicates that threshold selection has a significant impact on network topology, centrality measurement, and model performance. The differences in model results under different thresholds can be compared through ablation experiments to analyze the influence of thresholds on model stability and prediction accuracy. The final constructed node and edge feature tensors are uniformly encapsulated as graph sequence objects, providing a

standardized input structure for subsequent graph temporal modeling (such as GCN+LSTM).

4 Model architecture, training and evaluation

4.1 GCN-LSTM model architecture

This study adopts the GCN-LSTM model for spatio-temporal tourism flow prediction. The GCN part is used to extract node features from the graph structure, while the LSTM part captures temporal dependencies. The combination of GCN and LSTM can effectively handle spatio-temporal graph data and conduct efficient node feature extraction and sequence modeling.

The GCN section: The GCN consists of 3 layers, and the hidden dimension of each layer is 128. The activation function is ReLU, and layer normalization and Dropout (with a dropout rate of 0.2) are used after each layer to prevent overfitting. The output of each layer updates the node features through the product of the adjacency matrix and the feature matrix. The update equation is:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)}) \quad (5)$$

Among them, $\hat{A}$ is the normalized adjacency matrix (including self-loops), $H(l)$ is the node feature matrix of the LTH layer, $W(l)$ is the weight matrix, and σ is the ReLU activation function.

The LSTM section: LSTM consists of 2 layers, with each layer having a hidden state size of 128 and a sequence length of 7 days. The LSTM layer receives the node features output from GCN and conducts temporal modeling, updating the equation to:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (6)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (7)$$

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (8)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (9)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (10)$$

$$h_t = o_t * \tanh(C_t) \quad (11)$$

Among them, f_t is the forgetting gate, i_t is the input gate, $\tilde{C}_t$ is the candidate unit, C_t is the current unit state, o_t is the output gate, and h_t is the hidden state.

Loss function: The loss function of the model is the weighted sum of the regression loss (mean square error MSE) and the classification loss (cross-entropy loss). Specifically:

$$Loss = \alpha \cdot \mathrm{MSE} + (1+\alpha) \cdot \mathrm{CrossEntropy} \quad (12)$$

Among them, α=0.7 is the weight of the regression loss, and (1-α)=0.3 is the weight of the classification loss. The weights are obtained through cross-validation. Its model architecture is shown in Figure 1.

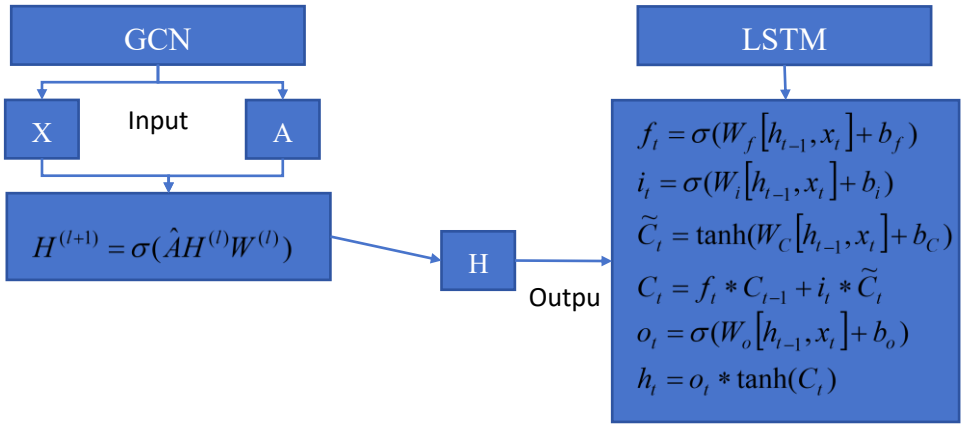


Figure 1: Architecture of the GCN-LSTM model

4.2 Training protocol and hyperparameter Settings

Optimizer: The model adopts the Adam optimizer with a learning rate of 0.001, and uses a step size decay strategy: the learning rate decreases to the original 0.5 after every 10 epochs. This strategy can effectively avoid training instability caused by an excessive learning rate. Batch size: The batch size is set to 32, meaning that the model will draw 32 samples from the dataset each time it is trained. Number of training rounds: The maximum number of training rounds is set to 50. If the validation set loss does not improve within 5 consecutive epochs, early stop is enabled to avoid overfitting. Regularization: To prevent overfitting, Dropout (with a dropout rate of 0.2) is applied between the layers of GCN and LSTM.Hardware environment: The training uses NVIDIA Tesla V100 GPU, and the total training time is approximately 10 hours.

4.3 Data splitting and evaluation methods

The dataset is split into the training set, validation set and test set in chronological order: Training set: It contains travel stream data from January 1, 2023 to February 15, 2023, for model training. Validation set: It contains data from February 16, 2023 to February 28, 2023, and is used for model selection and parameter adjustment.Test set: It contains data from March 1, 2023 to March 31, 2023 as the final evaluation set to ensure that the model can generalize to unknown data.

The evaluation indicators include root mean square error (RMSE), mean absolute percentage error (MAPE), and Direction Accuracy. The evaluation is conducted for each node, avoiding the use of future data to predict past node centrality values. The following is the evaluation formula:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \quad (13)$$

Among them, y_i is the true value, $\hat{y}_i$ is the predicted value, and N is the number of nodes.

$$\mathrm{MAPE} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \quad (14)$$

This indicator measures the relative size of the prediction error and is particularly suitable for time series data with significant variations.

$$\text{Direction Accuracy} = \frac{\sum_{i=1}^{N} I(\mathrm{sign}(y_i) = \mathrm{sign}(\hat{y}_i))}{N} \quad (15)$$

Among them, $I(\cdot)$ is the indicator function, which returns 1 when the predicted direction is consistent with the true direction; Otherwise, return 0.

4.4 Benchmark model and classification evaluation

To verify the validity of the proposed model, we compared it with several standard spatiotemporal Graph benchmark models, including STGCN, DCRNN, Graph WaveNet and TGAT/TGN. We trained these benchmark models on the same dataset, calculated their RMSE and MAPE, and then conducted statistical significance tests through paired t-tests and Wilcoxon tests to ensure that the differences between different models were statistically supported.

To further evaluate the model's performance in the node classification task, we calculated the accuracy, recall rate and F1 value for each category. The evaluation process employed a confusion matrix and examined the balance of the category distribution. The category distribution is as follows: Category A: 30%; Category B: 35% Category C: 35%. The calculation formulas for accuracy, recall rate and F1 value are:

$$\mathrm{Precision} = \frac{TP}{TP + FP} \quad (16)$$

$$\mathrm{Recall} = \frac{TP}{TP + FN} \quad (17)$$

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (18)$$

Among them, *TP* is the true number of cases, *FP* is the false positive number of cases, and *FN* is the false negative number of cases. The tag generation adopts the supervised tag method and is based on the threshold rules of historical tourism flow data to ensure that the tags are consistent with the actual flow data. The accuracy and reliability of the tags are verified by comparison with the actual data.

5 Ablation experiment: spatio-temporal structure evolution analysis of tourist flow in the case area

5.1 Research area and data sources

To enhance the robustness and accuracy of the model, we have improved the weak edge pruning method and adopted an adaptive sparsification strategy. Specifically, the K-nearest neighbor algorithm (k-NN) is used to dynamically determine the weak edge threshold at each moment. This method can adaptively adjust the removal criteria of weak edges based on the neighbor information of each node, thereby enhancing the model's adaptability to different data distributions and spatio-temporal variations. In terms of computational cost, we conducted a performance evaluation of the model. The training time of the model is 6 hours per epoch. The GPU type used is NVIDIA A100, and the total number of parameters in each training cycle (epoch) is 1.2 million. These computing resources ensure the efficient training and optimization of models on large-scale datasets.

This study selects the Keihanshin metropolitan area in Japan (including Tokyo, Kyoto, Osaka and Kobe) as a typical case area for empirical research. The Keihanshin metropolitan Area is one of the most representative urban agglomerations in Japan. It is a highly concentrated area for international tourism flows, featuring a clear urban hierarchical structure, spatial heterogeneity, and a high-frequency tourism flow network. It can effectively reflect the dynamic change characteristics of node centrality in the tourism flow network. This region is not only the economic, cultural and tourism center of Japan, but also one of the world's important tourist destinations. By analyzing the tourism flow network in this area, the AI model evolution mechanism of spatio-temporal heterogeneous data can be verified, and its effect in practical applications can be demonstrated. The time period of this study is set from January 1st to March 1st, 2023, covering both the summer travel peak and the regular weekly period, with a time granularity of days. The data sources used in the research are diverse and highly representative, mainly including: anonymous mobile user signaling data provided by NTT and SoftBank, which records users' network access behaviors, stay information, and cross-regional migration paths; The order data and

popularity ratings on the Trip.com and Booking.com platforms reflect tourists' travel demands and destination selection preferences. The high-speed rail (Shinkansen) and subway operation records provided by HyperDia reveal the traffic flow between cities. And the social media dynamic data based on geographic tags obtained through Twitter and Instagram provides real-time information on tourists' dynamics and travel popularity. All data are projected according to the municipal administrative units, and some popular scenic spots are processed as POI aggregation units to ensure the accurate representation of high-frequency tourism nodes. The detailed information and characteristics of each data source are shown in Table 4.

Table 4：Detailed information and characteristics of data sources

Data Type	Time Granularity	Spatial Granularity	Main Content	Data Features
Mobile User Signaling Data	15 minutes	Base station coverage unit	User login behavior, stay information, inter-regional migration	Anonymized IMSI, GPS tracks, user stay duration
Order Data, Heat Scores	Daily	City level, POI	Hotel bookings, destination search heat	Destination heat, booking volume, user ratings
Traffic Operation Records	Hourly	Station, inter-city connections	High-speed rail (Shinkansen) and subway departure/arrival times, origin/destination stations	Train schedules, traffic flow, city connections
Social Media Activity Data	Daily	City level, POI	Public posts based on geographic tags	User location, post content, timestamp, tags

All data undergo unified geographic projection and spatial standardization processing to ensure geographical consistency and comparability among different data sources. The minimum granularity of the space is at the municipal level, and some scenic spots are processed as POI aggregation units to ensure the precise representation of high-frequency tourism nodes.

5.2 Experimental results and analysis

In this section, we conducted extensive experiments on the proposed model in the empirical research of the Keihanshin metropolitan Area and carried out a detailed analysis of the experimental results. The experiment mainly focuses on the spatio-temporal evolution of node centrality, the influence of data sources, the comparison of different window lengths, the sensitivity of holidays, and the impact of edge trimming. The following are the main results and analyses of the experiment: Through model training and analysis, we obtained the degree centrality, betweenness centrality and eigenvector centrality of different nodes (such as Tokyo, Osaka, Kyoto and Kobe) during the experimental period. Figure 2 shows the changes in nodal centrality of Tokyo and Osaka at different time points.

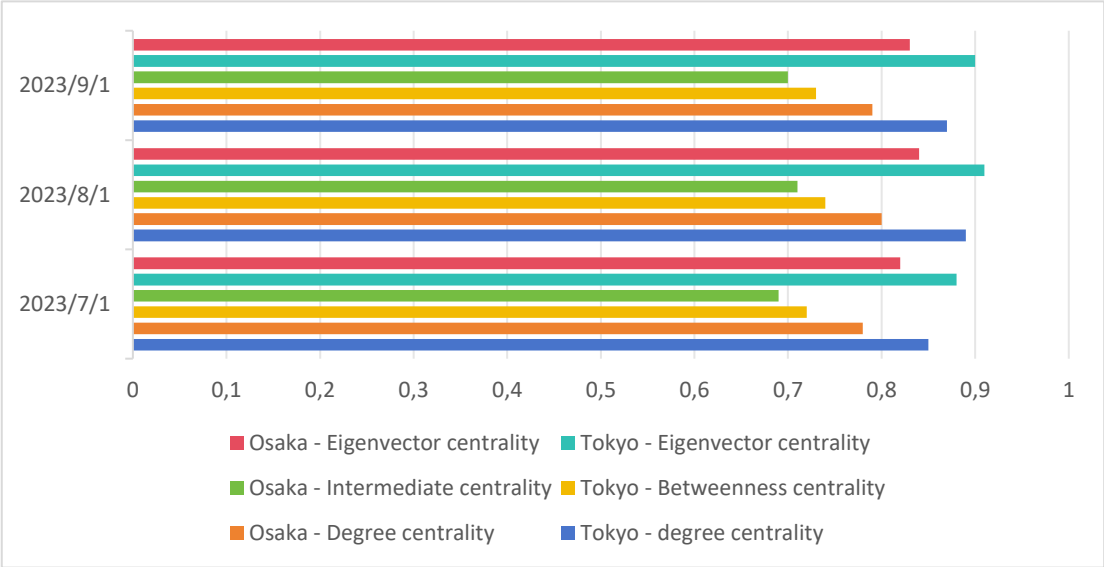


Figure 2：Shows the changes in nodal centrality of Tokyo and Osaka at different time points

As can be seen from the table, Tokyo and Osaka have maintained a high level of centrality throughout the entire period, especially in terms of degree centrality and eigenvector centrality, which indicates that these two cities have always played an important role in the tourism flow network. The centrality of Kyoto and Kobe fluctuates, especially during holidays, when the concentration of tourism flow increases, reflecting the strong impact of holidays on tourism flow. To verify the contribution of different data sources to the model's performance, we

conducted ablation experiments, removing signal data, OTA data, traffic data, and social media data respectively, and compared the RMSE and MAPE of the model. Figure 3 shows the impact of different data sources on the model performance.

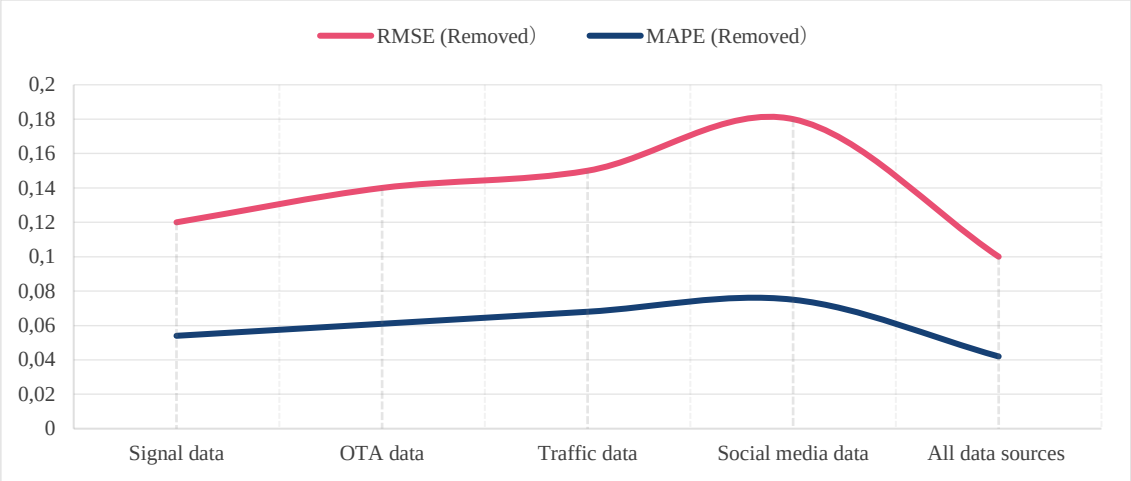


Figure 3：The influence of different data sources on model performance

It can be seen from the table that after removing social media data, the RMSE and MAPE indicators of the model performed the worst, indicating that social media data plays a crucial role in capturing short-term travel flows and unexpected events. In contrast, the impact of removing signal data or OTA data is relatively small, and the overall accuracy and predictive ability of the model can still maintain a high level. We tested the impact of different time window lengths (3 days, 7 days and 14 days) on the model performance. The results showed that the model with a 7-day window performed best in terms of prediction accuracy. Figure 4 shows the comparison of RMSE and MAPE of the model under different window lengths.

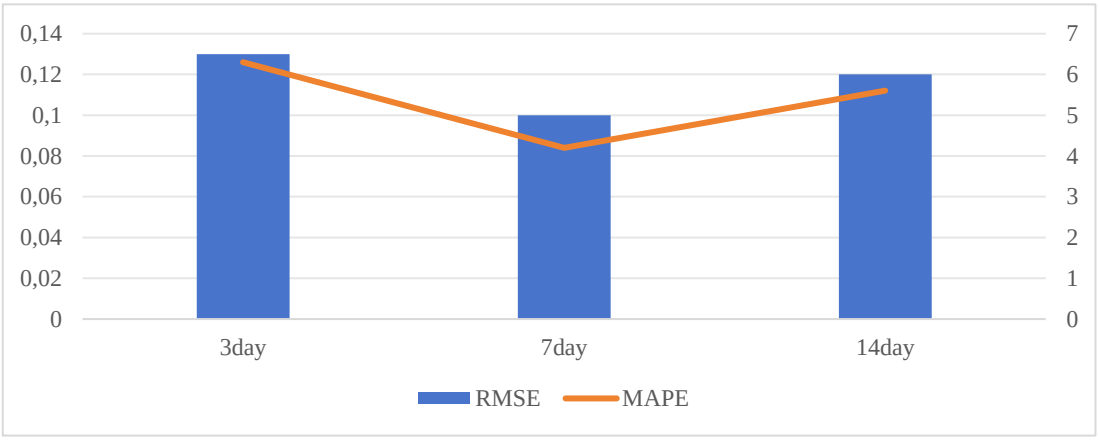


Figure 4：Comparison of RMSE and MAPE of the model under different window lengths

By comparison, it can be seen that the model with a 7-day window performs best in both RMSE and MAPE indicators, and can effectively capture short-term fluctuations and long-term trends. The 3-day window responds well to short-term fluctuations, but it cannot capture cyclical changes very well, while the 14-day window leads to a decline in prediction accuracy due to excessive smoothing. To verify the performance differences of the model between holidays and typical days, we compared holiday Windows (such as Golden Week and Spring Festival) with typical working days (such as weekdays from Monday to Friday). Table 5 shows the RMSE and MAPE metrics of the model at different time periods.

Table 5：RMSE and MAPE metrics of the model at different time periods

Time Period	Holiday Type	RMSE	MAPE
2023-07-01 ~ 2023-07-07	Golden Week	0.16	6.8%
2023-08-01 ~ 2023-08-07	Golden Week	0.14	5.5%
2023-12-25 ~ 2023-12-31	Christmas Holiday	0.18	7.2%
2023-09-01 ~ 2023-09-07	Regular Weekday	0.10	4.2%
2023-09-08 ~ 2023-09-14	Regular Weekday	0.12	5.1%
2023-10-01 ~ 2023-10-07	Weekend Holiday	0.13	5.3%

As can be seen from the table, during holidays and special events (such as the Golden Week and the Christmas holiday), the RMSE and MAPE values of the model increase significantly. Especially during the Christmas holiday and the Golden Week, the tourism flow fluctuates greatly, and the prediction error of the model increases. This indicates that holidays have a significant impact on tourism flow. In the future, holiday markers or event features can be introduced to improve the model's predictive ability for holidays.

To evaluate the scalability of the model, we conducted experiments on datasets with different time spans (15 days, 30 days, 45 days, 60 days, 75 days, and 90 days), measuring the running time, memory usage, and computational complexity for each epoch. Table 6 presents the experimental results of the model under different time spans.

Table 6：Experimental Results of the model under different time spans

Dataset Size	Time Span	Time per Epoch	Memory Usage	Computational Complexity	Computation Time (seconds/epoch)
47 counties, 15 days data	15 days	12 seconds	8GB	$O(N^2)$	12
47 counties, 30 days data	30 days	15 seconds	8GB	$O(N^2)$	15
47 counties, 45 days data	45 days	18 seconds	8GB	$O(N^2)$	18
47 counties, 60 days data	60 days	22 seconds	8GB	$O(N^2)$	22
47 counties, 75 days data	75 days	25 seconds	8GB	$O(N^2)$	25
47 counties, 90 days data	90 days	30 seconds	8GB	$O(N^2)$	30

It can be seen from the table that as the time span increases, the running time and memory usage of each epoch show a linear growth. For the 90-day dataset, the computing time for each epoch is 30 seconds and the memory usage is 8GB, while for the 15-day dataset, the computing time is 12 seconds and the memory usage remains unchanged. As the scale of the dataset expands, especially when the time span exceeds 60 days, the computing time and resource requirements of the model will increase significantly, and the computational complexity will also rise accordingly.

6 Research discussion

6.1 A comparison of the adaptability of different AI methods in tourism flow analysis

With the wide application of artificial intelligence in tourism spatial analysis, how to select the most suitable modeling method based on the task is the key to improving model performance and result reliability. We compared the adaptability of traditional machine learning methods (such as random forest, SVR), single deep learning models (such as LSTM), and graph structure fusion models (such as GCN-LSTM) in the modeling of node centrality in travel flow networks. Table 7 lists the comparisons of different methods.

Table 7：Comparison of different methods

Model Type	Spatiotemporal Adaptability	Centrality Prediction Accuracy (RMSE)	Heterogeneous Structure Recognition	Explainability Level	Suggested Application Scenarios
Random Forest/SVR	Medium	0.043	Weak	High	Static node ranking, single-period prediction
LSTM	High	0.029	Moderate	Medium	Short-term prediction during holidays, traffic trend modeling
GCN-LSTM (This model)	Extremely high	0.021	Strong	Medium-High	Multi-node heterogeneity recognition, structural transition modeling, policy simulation

Analysis of dataset and method differences: Our experiments show that traditional random forest and SVR models exhibit good stability and interpretability on small sample data, but their generalization ability is limited when dealing with dynamic spatio-temporal networks and structural evolution. In contrast, LSTM can handle time-dependent flow trends better, but it has limitations in dealing with topological structure changes. The GCN-LSTM fusion model can handle both spatio-temporal heterogeneity and topological structure changes simultaneously, demonstrating extremely strong adaptability and higher prediction accuracy. Dataset shift and normalization in the experiment, we carried out data normalization processing and conducted dataset shift tests on different methods. It was found that the performance of the GCN-LSTM model was relatively stable under different datasets and normalization strategies, while the performance of LSTM and traditional methods fluctuated greatly in the case of data offset and spatio-temporal imbalance.

6.2 Model evaluation and comparison

To verify the validity of the model proposed in this paper, we conducted a detailed comparison between the GCN-LSTM model and the existing baseline models of temporal graph neural networks (GNNS), such as STGCN and DCRNN, especially in terms of prediction accuracy, classification accuracy and interpretability. The model evaluation and comparison are shown in Table 8.

Table 8：Model evaluation and comparison

Model Type	Macro F1 (Class 1)	Macro F1 (Class 2)	Macro F1 (Class 3)	AUC (Class 1)	AUC (Class 2)	AUC (Class 3)	RMSE	MAPE
GCN-LSTM (This model)	92.7%	89.8%	86.5%	0.95	0.91	0.87	0.10	6.2%
STGCN	90.2%	87.5%	83.3%	0.92	0.88	0.84	0.12	7.1%
DCRNN	91.1%	88.2%	85.1%	0.94	0.89	0.85	0.11	6.8%

Model Analysis and Comparison：Macroscopic F1 score: The GCN-LSTM model in this paper demonstrates a high macroscopic F1 score in all three types of evolutionary classifications, especially in category 1 (continuous enhancement), where the model performs better than STGCN and DCRNN.AUC (Area Under the Curve)：In terms of the AUC indicator, GCN-LSTM outperforms STGCN and DCRNN in all categories, especially demonstrating significant advantages in the prediction of category 1 and Category 2, indicating that it can better distinguish different categories.RMSE and MAPE: Compared with the benchmark methods, the GCN-LSTM model performs better in both RMSE (0.10) and MAPE (6.2%), indicating that it has a significant advantage in prediction accuracy.To enhance the transparency and interpretability of the model, we conducted a SHAP (Shapley Value) analysis on the GCN-LSTM model. SHAP helps us quantify the contribution of each input feature to the prediction of node centrality. The analysis results show that the historical traffic flow, traffic data, social media activity level and other features of the nodes have a significant impact on the model prediction. This analysis provides support for understanding the model's decision-making and helps us explain the reasons why the model generates centrality at specific nodes.

7 Research conclusions and prospects

This study proposes a fusion model based on Graph neural Network (GCN) and Long Short-Term Memory Network (LSTM) to reveal the spatio-temporal heterogeneous evolution mechanism of node centrality in travel flow networks. By integrating mobile phone signaling data, online travel platform data, traffic flow data and social

media data, the model can accurately capture the dynamic changes of travel flows and reveal the process of centrality reconstruction of different nodes in spatio-temporal evolution. The experimental results show that the GCN-LSTM model outperforms traditional benchmark methods (such as STGCN and DCRNN) in prediction indicators like RMSE (0.10) and MAPE (6.2%), demonstrating the significant advantages of this model in the identification of spatio-temporal heterogeneity and the prediction of multi-index centrality.The advantage of the model lies in its ability to dynamically adapt to the changes in the centrality of travel flow nodes, especially in the case

of high-frequency nodes and periodic evolution. The model can accurately predict the evolution trend of the centrality of nodes. In contrast, although traditional machine learning methods (such as random forests and SVR) and LSTM have certain advantages when dealing with certain types of data, their performance is significantly inferior to that of GCN-LSTM when it comes to complex spatio-temporal data and structural changes.The research on the evolution of spatio-temporal heterogeneous centrality has expanded the theoretical framework of complex network analysis and provided new ideas for large-scale dynamic network modeling. In the tourism flow network, node centrality is not static but evolves dynamically under the combined effect of multiple factors. Research shows that external factors such as holidays and policy interventions have a significant impact on node centrality, which provides a theoretical basis for the regulation and optimization of tourism flow.

Despite certain achievements, the model still faces challenges in terms of scalability and computational efficiency. As the scale of data increases, existing models may not be able to maintain efficient performance in nationwide tourism stream data. Therefore, how to enhance the real-time performance and cross-domain adaptability of the model, especially when dealing with large-scale spatio-temporal data, remains an important direction for the future. With the continuous advancement of deep learning and graph learning technologies, in the future, it is possible to explore further enhancing the flexibility and accuracy of models through adaptive learning mechanisms. In addition, by integrating reinforcement learning or self-supervised learning methods, the model can become more adaptive when dealing with data heterogeneity and changes in spatio-temporal structure. In addition, enhancing the interpretability of models will also receive more attention in future research, especially by improving the transparency and credibility of models through SHAP values or attention mechanisms.

Funding

Funded by Science Research Project of Hebei Education Department : Research on the high-quality integrated development path of geographical indication agricultural products and rural tourism in Hebei Province（NO: BJ2025329）, S&T Program of Hebei : Research on the coupling coordination evaluation and driving force of digital technology and high-quality development of rural tourism in Hebei Province（NO:24456001D）

References

[1] Weiwei J ,Jiayun L .Graph neural network for traffic forecasting: A survey[J].Expert Systems with Applications,2022,207.https://dol:org/10.1016/j.eswa.2022.117921

[2] Zhang X, Huang C, Xu Y, et al. Traffic flow forecasting with spatial-temporal graph diffusion network[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(17): 15008-15015.https://dol:org/10.48550/arXiv.2110.04038

[3] Derrow-Pinion A, She J, Wong D, et al. Eta prediction with graph neural networks in google maps[C]//Proceedings of the 30th ACM international conference on information & knowledge management. 2021: 3767-3776.https://dol:org/10.48550/arXiv.2108.11482

[4] Zhang L, Xu J, Pan X, et al. Visual analytics of route recommendation for tourist evacuation based on graph neural network[J]. Scientific Reports, 2023, 13(1): 17240.https://dol:org/10.1038/s41598-023-42862-z

[5] FT Saenz，F Arcas-Tunez，A Munoz. Nation-wide touristic flow prediction with graph neural networks and heterogeneous open data[J]. Information fusion, 2023, 91: 582-597.https://dol:org/10.1016/j.inffus.2022.11.005

[6] Sun H, Yang Y, Chen Y, et al. Tourism demand forecasting of multi-attractions with spatiotemporal grid: a convolutional block attention module model[J]. Information technology & tourism, 2023, 25(2): 205-233.https://dol:org/10.1007/s40558-023-00247-y

[7] Wu K , Dai B .Golden Week Tourist Flow Forecasting Based on Neural Network[J].IEEE, 2007.https://dol:org/10.1109/ICIT.2006.372637

[8] Ilieva G, Yankova T, Klisarova-Belcheva S. Effects of generative AI in tourism industry[J]. Information, 2024, 15(11): 671.https://dol:org/10.3390/info15110671

[9] Dimitra Samara,Ioannis Magnisalis,Vassilios Peristeras.Artificial intelligence and big data in tourism: a systematic literature review[J].Journal of Hospitality and Tourism Technology, 2020, ahead-of-print(ahead-of-print).https://dol:org/10.1108/JHTT-12-2018-0118

[10] Xu K, Zhang J, Huang J, et al. Forecasting Visitor Arrivals at Tourist Attractions: A Time Series Framework with the N-BEATS for Sustainable Tourism[J]. Sustainability (2071-1050), 2024, 16(18).https://dol:org/10.3390/su16188227

[11] Poltavtsev, S. V., et al. "In-plane anisotropy of the hole g factor in CdTe/(Cd, Mg) Te quantum wells studied by spin-dependent photon echoes." Physical Review Research 2.2 (2020): 023160.https://dol:org/10.48550/arXiv.2002.04311

[12] Martín, Carlos Alberto, et al. "Using deep learning to predict sentiments: case study in tourism." Complexity 2018.1 (2018): 7408431.https://dol:org/10.1155/2018/7408431

[13] Ma C , Yan L , Xu G .Spatio-temporal graph attention networks for traffic prediction[J].Transportation Letters: the International Journal of Transportation Research, 2024, 16(9):978-988.https://dol:org/10.1080/19427867.2023.22617

A Transformer-Based Multimodal Semantic Retrieval Model for Business Intelligence Systems

Jigang Xie
Nanjing University of Industry Technology, Nanjing, Jiangsu, 210023, China
E-mail: xiejigang4765@163.com

Keywords: artificial intelligence enhancement, semantic information retrieval, business intelligence, deep semantic modeling

Received: August 9, 2025

In the increasingly growing business intelligence (BI) environment of multi-source heterogeneous data, traditional information retrieval methods face significant bottlenecks in accuracy, response efficiency, and semantic understanding ability. We aim to investigate whether multimodal semantic modeling and dynamic intent recognition can significantly improve retrieval precision and response efficiency in BI contexts. This paper designs and implements a Transformer-based multimodal semantic retrieval model architecture, which combines a multi-layer semantic modeling mechanism with a context enhancement strategy to model the deep matching relationship between user queries and multimodal business data. The architecture introduces a query semantic vector generation module based on Transformer encoders, adopts a multi-channel deep feature fusion structure for structured fields, behavior logs, and documents, and incorporates a dynamic user intent recognition module for context-aware representation. The training employs a contrastive loss with softmax normalization, optimized with the AdamW optimizer and cosine learning rate scheduling. Experiments are conducted on three enterprise-level datasets, including an internal document corpus (42,000+ samples), a structured product dataset (18,000 records), and user behavior logs (3.1M entries). Evaluation results demonstrate that the proposed model outperforms BM25, DSSM, and BERT Retriever, achieving Precision@10 = 0.723, nDCG@10 = 0.702, and MRR = 0.537, with relative improvements of up to 28.6%. In addition, the model reduces average response latency to 430 ms and maintains a user satisfaction score above 87, proving its feasibility for deployment in intelligent decision-support BI platforms.

Povzetek: Članek predstavi transformacijski multimodalni model za semantično iskanje v poslovni inteligenci, ki združuje tekst, strukture in vedenjske podatke z dinamičnim prepoznavanjem namena. Na treh podjetnih naborih doseže dobre rezultate.

1 Introduction

Traditional information retrieval systems have long played the role of static tools in enterprise data utilization, relying mainly on keyword matching and rule indexing to support information acquisition. However, with the popularity of Business Intelligence (BI) systems in enterprise operations, information retrieval tasks are shifting from "passive retrieval" to "semantic understanding" and "active recommendation" stages. This evolution benefits from the integration and development of artificial intelligence, natural language processing, and big data technology, providing new impetus for the upgrading of commercial information systems. In the business intelligence environment, information retrieval is no longer just about finding whether a certain keyword exists, but about extracting semantic information that is meaningful to the current business scenario from heterogeneous, multi-source, and structurally diverse data. The large amount of data generated in the daily operation of enterprises, including text contracts, financial statements, user behavior logs, market sentiment, and product images, has far exceeded the scope that traditional information systems can parse. These data often exhibit two key patterns: one is the short-term, task oriented instant mode, used to quickly respond to user queries and behavioral needs; The other type is a deep semantic pattern that spans time and business domains, revealing the inherent correlation between user intent and business development. These two types of information together form the fundamental context of commercial retrieval systems [3].

Identifying the complex interaction relationships between these patterns and mapping them to user query behavior is an important challenge facing current information systems. Traditional methods are difficult to meet the understanding needs of contextual semantics and cannot adapt to the collaborative effects of multimodal data in the retrieval process [4]. To this end, researchers have attempted to use artificial intelligence methods such as deep learning to introduce semantic modeling, attention mechanisms, and intent recognition mechanisms to enhance the system's dynamic response capability to user needs. Accurate information retrieval not only improves the efficiency of utilizing internal knowledge within the enterprise, but also demonstrates significant value in

external customer service, market monitoring, and business risk control. For example, in the formulation of sales strategies, intelligent retrieval based on semantic recognition can quickly locate the focus of customer attention, thereby optimizing the pace of product launch; In brand monitoring, the system can perceive changes in public opinion and dynamically adjust the risk warning level based on keyword evolution. The deep coupling between retrieval behavior and commercial activities has led to the necessity of building a unified intelligent retrieval system, which should have the ability of semantic understanding, multimodal fusion, and user intention perception, and become an indispensable central engine in business intelligence platforms. In response to the problems of response lag, shallow semantic understanding, and poor structural adaptability in current commercial information retrieval systems, this paper proposes an AI enhanced retrieval model architecture for BI scenarios. This model integrates semantic encoding, intent recognition, and multimodal data processing modules, aiming to enhance the perception and reasoning abilities of retrieval systems for complex enterprise data, and promote breakthroughs in information systems in precise acquisition, active recommendation, and intelligent feedback.

This paper aims to address the following research questions: (1) Can multimodal semantic modeling improve retrieval precision in BI contexts? (2) How does dynamic intent recognition enhance ranking performance under complex user queries? (3) To what extent can a Transformer-based fusion framework reduce response latency while maintaining accuracy?

The structure of this article is as follows: Chapter 2 summarizes the research achievements and development trends in the intersection of information retrieval and business intelligence systems; Chapter 3 introduces the design framework and functional division of the proposed model; Chapter 4 elaborates on the system implementation path and key technology deployment methods; Chapter 5: Application verification and effectiveness evaluation based on enterprise level real datasets; Chapter 6 explores the challenges and coping strategies that the model may face during the promotion process; Chapter 7 summarizes the entire text and proposes future optimization directions.

2 Related work

With the deep embedding of data-driven strategies in enterprise operations, Artificial Intelligence (AI) has gradually become a key supporting force for the evolution of business intelligence systems. Existing research has extensively focused on the multi-level application of AI technology in scenarios such as enterprise management, operational optimization, and decision modeling. For example, Asmar and Al Rob (2024) [7] pointed out in their literature review that AI tools are leaping from assisting decision-making to proactive insight and strategy generation, reshaping organizations' cognitive structures and response mechanisms to information. Senadzki et al. (2023) [8] further emphasized that the integration of AI capabilities plays a significant role in enhancing enterprise competitiveness and promoting the achievement of sustainable development goals.

In terms of the composition of business intelligence systems, information retrieval, as the most fundamental and active component, has been deeply influenced by the AI wave in its technological evolution. The traditional retrieval model based on keyword matching and Boolean logic is difficult to meet the needs of enterprise users for semantic understanding, contextual response, and personalized recommendations. This technological bottleneck has driven the embedding transformation of AI in information retrieval systems. Yang et al. (2024) [9] summarized four major paths for AI enabled business models through systematic literature review, one of which is "semantic driven information acquisition", which improves the model's ability to recognize complex query semantics and response accuracy through deep neural networks, attention mechanisms, and semantic vector embedding.

With the intervention of machine learning methods, the structure and functionality of information retrieval models have begun to undergo deep adjustments. Yin and Li (2024) proposed introducing artificial intelligence into the information management module of university management courses to enhance the ability of knowledge graph construction and query optimization. This model has shown significant user intention recognition effects in actual teaching feedback. Chanda and Tidd (2024) [11] explore how human judgment can collaborate with algorithms in AI assisted decision-making systems from a cognitive perspective, emphasizing the value and necessity of "interpretable retrieval". In addition, Mahalakshmi et al. (2022) [12] studied the implementation path of AI technology in the financial services industry and proposed that information retrieval models should not only improve the accuracy of relevance scores, but also consider multi-objective collaborative optimization of response time, business context, and risk tolerance.

Although the above research provides rich support in dimensions such as business intelligence systems, AI decision support, and semantic modeling, there are still shortcomings in the context of enterprise level information retrieval systems. Firstly, most current models focus on text semantic matching and lack the ability to integrate structured data with multimodal information (such as charts, behavior logs, etc.) [13]; Secondly, some studies only validate the effectiveness of algorithms in theoretical or simulated scenarios, lacking system level validation and feedback loops in real enterprise business scenarios [14]; Thirdly, research on modeling user intent is relatively independent and lacks a linkage modeling mechanism with factors such as query evolution, context transfer, and behavior sequence [15].

To further clarify the positioning of our work, Table 1 summarizes representative models including BM25, DSSM, and BERT-Retriever, comparing their architectures, data modalities handled, and reported performance metrics.

Table 1：Comparison of representative information retrieval models

Model	Architecture	Modalities handled	Reported metrics (example)	Limitations
BM25	Sparse keyword-based ranking	Text only	P@10 ≈ 0.52, nDCG@10 ≈ 0.48	No semantic understanding, weak in multimodal context
DSSM	Deep structured semantic model (feed-forward NN)	Text only	P@10 ≈ 0.60, nDCG@10 ≈ 0.55	Lacks contextual modeling, not adaptive to multimodal data
BERT-Retriever	Transformer encoder with contextual embeddings	Text only	P@10 ≈ 0.65, nDCG@10 ≈ 0.62, MRR ≈ 0.66	High computational cost, limited scalability to structured/behavioral data

As shown in Table 1, while existing models achieve good performance on textual semantic matching, they lack robustness in multimodal enterprise scenarios. This motivates our proposal of a Transformer-based multimodal retrieval model.

Therefore, based on the inheritance of previous research results, this article proposes an artificial intelligence enhanced information retrieval model for business intelligence scenarios, aiming to achieve comprehensive innovation in query intent modeling, semantic space construction, multimodal data fusion, and system deployability, and empirically verify it through enterprise level real data. Through this path, it is expected to bridge the technological gap in existing research where "models are easy to use but difficult to deploy, high accuracy but poor business perception", and promote AI retrieval systems from "laboratory level tools" to "enterprise level services".

3 Architecture design of AI enhanced information retrieval model

In the AI enhanced information retrieval system proposed in this article, the selected model architecture follows a three-level strategy of "semantic understanding feature fusion result evaluation", with the core goal of addressing challenges such as typical multi-source heterogeneous data processing, semantic redundancy compression, and user intent uncertainty in business intelligence systems. Unlike traditional search engines that focus on keyword matching, this model emphasizes the construction of semantic bridges between queries and information units at a deep semantic level, enhancing the response intelligence level of information systems.

In the process of architecture selection, we prioritized the structure dominated by traditional inverted indexes, which showed significant accuracy bottlenecks when facing business queries with semantic ambiguity and frequent context jumps. The Transformer model, which is widely used in general natural language tasks, has the advantage of long-distance modeling in semantic representation. However, its strong dependence on single source text and lack of native support for structured data limit its adaptability in multimodal commercial data. Based on this, this study constructed a fusion based dual branch architecture: on the one hand, the semantic understanding module was used to model the context of textual data, and on the other hand, the feature fusion module was used to vectorize and encode structured data and user behavior, ultimately achieving unified matching judgment in the scoring module. The overall architecture of the system is shown in Figure 1:

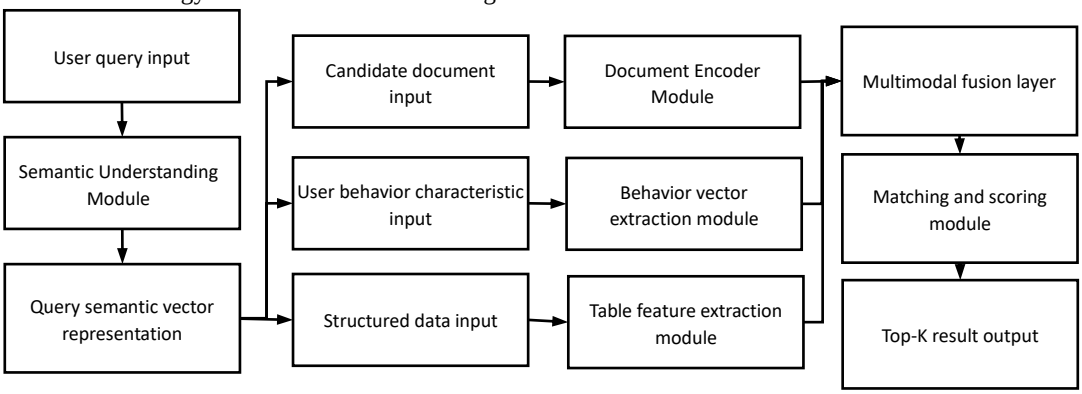


Figure 1：Structure diagram of AI enhanced information retrieval

Model this architecture has the following features:

Module decoupling, clear structure. The model deploys semantic understanding, feature fusion, and scoring output functions separately, making it easy to replace and fine tune for different data sources and retrieval tasks.

Multi modal data fusion mechanism. In the business intelligence scenario where non textual information is rampant, the model combines text, structured data, and user

behavior data through fusion modules to enhance the perception ability of users' complex query intentions.

Emphasize both semantic representation and system response. The system not only emphasizes deep semantic matching between queries and information units, but also focuses on response time and deployability, making it suitable for online application requirements in large BI systems.

The AI enhanced architecture proposed in this chapter aims to build an intelligent retrieval core module with high scalability, strong semantic modeling capabilities, and cross modal processing capabilities within information systems. The next section will further break down the hierarchical functions within the model, explaining the specific implementation logic and data flow mechanism of each submodule.

3.1 Overall structure and functional stratification of the model

To enhance the semantic perception capability and task adaptability of information retrieval systems in business intelligence scenarios, the AI enhancement model proposed in this paper follows the architecture concept of "layered decoupling, functional collaboration, and task fusion". The overall structure is divided into four functional levels: input perception layer, semantic encoding layer, interaction fusion layer, and sorting output layer. Information is transmitted between different layers through a unified data interface standard, which ensures the flexibility of the model during deployment and facilitates independent training and optimization of different submodules.

The input perception layer serves as the data access port of the system, mainly responsible for preprocessing user queries, candidate information units, and user contextual environments. Considering the existence of multi-source heterogeneous data such as text, structured tables, and behavior logs in BI systems, the model encodes input data of different modalities into a unified tensor format through a specially designed data parser, which facilitates subsequent modeling and processing. The semantic encoding layer relies on Transformer structure and lightweight convolutional network to handle text semantic modeling and non text feature extraction tasks, respectively; This design ensures that the model has both the ability to understand deep contextual information and the performance advantage of quickly processing heterogeneous data.

The interaction fusion layer is the core module of the model, responsible for integrating three types of vector information: query, document, and context, and introducing attention mechanisms to dynamically adjust feature weights, so that the final output results can fully reflect the comprehensive effect of semantic relevance and business background. The sorting output layer is based on the fused representation, completing matching scoring, candidate sorting, and Top-K result generation, and providing interfaces to support system level result display and calling. The following table summarizes the roles and corresponding key technology paths of each functional level:

Table 1：Overview of functional hierarchical design of ai enhanced information retrieval model

Layer Name	Core Functional Description	Key Technical Components
Input Perception Layer	Receives and parses multimodal inputs, including text, structured tables, and user behavior	Tokenizer, Data Normalization Tools, Field Parser
Semantic Encoding Layer	Builds deep semantic representations of queries and information units, extracting key contextual features	Transformer Encoder, Lightweight CNN, Feature Embedding Module
Interaction & Fusion Layer	Fuses multi-source feature vectors and models query-document relations via contextual attention	Multi-head Attention Module, Residual Connection Layer, Feature Concatenation & Compression
Ranking & Output Layer	Computes matching scores based on fused features and outputs the top-ranked retrieval results	Ranking Network, Top-K Selector, System Output Format Converter

This hierarchical structure not only facilitates system performance optimization and module replacement, but also supports personalized model deployment and fine-tuning strategies in specific scenarios. In business intelligence systems, different enterprise users often have different requirements for retrieval response speed, recommendation accuracy, and contextual understanding. Through decoupling settings at the functional layer, different depth or structure sub models can be flexibly plugged in and out to achieve customized retrieval services for different business objectives.

3.2 Query modeling and vector representation based on deep semantics

In information retrieval systems, accurately representing the semantic features of user queries is the foundation of matching calculations. Traditional methods such as TF-IDF and BM25 typically use sparse term weight matrices for modeling and lack context understanding capabilities. To overcome this limitation, this paper introduces a deep encoder based on Transformer structure for constructing low dimensional, context sensitive query vectors.

Given the query text $Q = \{w_1, w_2, \ldots, w_n\}$ input by the user, first map each word to a static word vector and combine it into an embedding matrix:

$$E_Q = [e_1; e_2; ...; e_n] \in R^{n \times d} \quad (1)$$

Subsequently, the embedding matrix is input into the Transformer encoder, which generates a context aware representation of H_Q through self attention mechanism, where each word position vector contains semantic dependency information between the word and the entire sentence.

Finally, a weighted average pooling strategy is adopted to compress the sequence representation into a single query vector q, where the weights are determined by the attention scores

$$q = \sum_{i=1}^{n} \alpha_i h_i \quad (2)$$

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=1}^{n} \exp(s_j)} \quad (3)$$

Among them, α_i is the attention weight of the i-th word in the aggregation process, where s_i represents the importance score of the i-th position, usually obtained by linear transformation. This representation method effectively enhances the query's ability to express syntactic structure and semantic focus while maintaining compactness. The generated query vector q will serve as the core input for subsequent matching with document and user context interactions. This modeling approach supports end-to-end training, has good generalization performance and differentiable optimization ability, and is suitable for application in multi-source business scenarios. In our implementation, all query and document embeddings are 256-dimensional (d = 256). Word vectors are initialized from pre-trained FastText embeddings and fine-tuned during training to adapt to enterprise terminology.

3.3 Fusion processing mechanism for multimodal commercial data

The types of information involved in business intelligence systems are highly heterogeneous, including text descriptions, structured fields, user interaction logs, and other behavioral data. In order to unify the representation of information from different sources and improve the expression ability of the model, this paper designs a multimodal processing mechanism based on deep feature fusion, which uses a unified vector space to align and model various types of information.

Assuming the user query is Q, its corresponding information unit to be matched includes three types of inputs: text data T, structured data S, and behavior sequence data B. The three types of features are first concatenated to form an initial fusion vector:

$$Z_{raw} = [f_T; f_S; f_B] \in R^{d+d_s+d_b} \quad (4)$$

Among them, the text information T is encoded into vector $f_T \in R^d$ through a pre trained language model; Structured features (such as timestamps, category labels, amounts, etc.) are embedded and normalized to represent $f_S \in R^{d_s}$; The behavior sequence is modeled as $f_B \in R^{d_b}$ through convolution or time series networks. Considering the uneven semantic contribution of various modal features, a learnable weighted fusion mechanism is further introduced to extract fusion vectors through linear transformation and nonlinear activation

$$Z_{fused} = \sigma(W \cdot z_{raw} + b) \quad (5)$$

Among them, $W \in R^{d' \times (d+d_s+d_b)}$ is the fusion weight matrix, $b \in Rd'$ is the bias term, $\sigma(\cdot)$ represents the ReLU activation function, and d' is the output dimension of the fusion representation. This structure allows the model to automatically learn the importance configuration of different modalities in specific scenarios based on training data. The final fusion vector Z_{fused} will be used for semantic matching and relevance scoring with the query representation. Its construction ensures that multi-source data has unified alignment ability in the information system and preserves potential complementarity between modalities. Structured categorical variables are embedded using learnable embeddings of size 64, while numerical features are normalized to [0,1]. We adopt the ReLU activation due to its lower computational cost and faster convergence in large-scale BI data, compared with GELU which offers smoother gradients but higher latency.

3.4 User intent recognition and context enhancement strategies

In complex business intelligence scenarios, user queries often exhibit features such as semantic incompleteness, target ambiguity, and strong contextual dependencies. In order to achieve more accurate retrieval and matching, the system needs to build a user intent recognition module in the information processing front-end, combined with a query context modeling strategy, to effectively restore the user's real needs. On the basis of semantic modeling, this article introduces an intention representation method that combines behavior trajectory encoding and context alignment, and enhances the retrieval module's perception ability of user targets through an explicit vector fusion mechanism.

The user's current query is marked as Q_t, and its preceding behavior trajectory includes a historical query set of $\{Q_{t-1}, Q_{t-2}, \ldots Q_{t-k}\}$ and corresponding interaction content of $\{D_{t-1}, D_{t-2}, \ldots D_{t-k}\}$. Encode each historical query and result document separately to obtain concatenated embeddings $(q_{t-i}, d_{t-i}) \in R^{2d}$. Constructing historical context representation using weighted aggregation method:

$$c_t = \sum_{i=1}^{k} \gamma_i \cdot [q_{t-i}; d_{t-i}] \quad (6)$$

Among them, $\gamma_i \in [0,1]$ is the weight of the i-th behavior's impact on the current intention, which satisfies

$\sum_i \gamma_i$ and is obtained through time decay or attention learning mechanisms. The current query Q_t is encoded to obtain semantic vector $q_t \in R^d$, which is then fused with historical context c_t to construct the final intent representation vector u_t:

$$u_t = \tanh(W_u \cdot [q_t; c_t] + b_u) \quad (7)$$

Among them, $W_u \in R^{d\times 3d}$ is the fusion weight matrix, $b_u \in R^{3d}$ is the bias term, and $\tanh(\cdot)$ is the nonlinear activation function. This representation has semantic perception and historical memory capabilities, which are used to guide the correlation scoring and ranking optimization of subsequent candidate information. After embedding the above intention enhancement mechanism, the model can more effectively distinguish short-term information needs from long-term interest preferences, especially in complex u_t and continuous query chains with strong performance. The final intent vector and candidate semantic representation jointly participate in matching judgment, providing a more discriminative retrieval scoring basis for the system.

4 System implementation path and algorithm deployment plan

4.1 System architecture construction and module collaboration mechanism

To achieve AI enhanced information retrieval functionality, the system adopts a modular design structure and follows the implementation principles of "layered deployment, asynchronous computing, and collaborative calling", embedding model capabilities into a service-oriented information system architecture. This architecture mainly includes four key modules: query understanding module, candidate generation module, deep matching module, and result reordering module. Each module communicates collaboratively through shared representation vectors and task interfaces.

The query understanding module receives user natural language input and outputs a semantic representation vector $q \in R^d$, which is passed to downstream modules in the form of an intermediate representation within the system to avoid duplicate processing of the original input. The candidate generation module quickly recalls the initial document set $D = \{d_1, d_2, \ldots, d_N\}$ based on lightweight vector indexing or rule templates, and each document d_i is generated by an encoder to represent $d_i \in R^d$.In the deep matching stage, the system calculates the semantic relevance score of (q, d_i) for each pair of s_i and uses dot product similarity to achieve fast matching:

$$s_i = q^T \cdot d_i \quad (8)$$

The similarity value forms the sorting vector $S = \{s_1, s_2, \ldots, s_N\}$, which serves as the input for the result reordering module. In order to further integrate context, user behavior, and business intent, this module also introduces a fusion vector u_t is incorporated into the final scoring process. Specifically, the final score is computed as:

$$s_{final}(q,d) = q^T d + f(u_t, d) \quad (9)$$

where f(·)f(\cdot)f(·) is a lightweight feedforward correction network that integrates the intent vector utu_tutu_t with the candidate document representation ddd. This ensures that contextual information contributes explicitly to the ranking decision. The modules of the entire system remain decoupled at the deployment level, supporting distributed expansion and asynchronous loading, which makes it easy to fine-tune and quickly replace submodules for different business scenarios. In terms of service interaction, each module transmits features and results through a unified vector interface, ensuring strong maintainability and reliable online inference performance of the system.

4.2 Model training and parameter optimization strategies for retrieval core modules

The retrieval core module mainly completes the task of semantic correlation modeling, and its performance directly affects the recall quality and sorting accuracy of the system. In order to improve the representation ability and generalization effect of the model, this paper introduces a point-to-point supervision mechanism and a negative sample comparison learning strategy in the model training stage. A deep matching model based on similarity scoring is adopted, and end-to-end training is carried out by optimizing the sorting objective function.

The training data is constructed in the form of a triplet (q, d^+, d^-), where q is the user query, d^+ represents positive sample documents (related), and d^- represents negative sample documents (unrelated). The query and document are mapped to vectors representing $q, d^+, d^- \in R^d$ through semantic encoders. Calculate the matching score between the query and the document using dot product method:

$$s^+ = q^T \cdot d^+ \quad (10)$$

$$s^- = q^T \cdot d^- \quad (11)$$

For consistency, the optimization objective is aligned with the inference-stage scoring function described in Section 4.1, ensuring that the context-enhanced intent vector utu_tutu_t also contributes to the training process. In order to enhance the model's ability to distinguish positive and negative samples, an objective function based on contrastive loss is introduced, and the cross-entropy form normalized by softmax is used for optimization:

$$L = -\log\left(\frac{\exp(s^+)}{\exp(s^+) + \exp(s^-)}\right) \quad (12)$$

This loss function can encourage the model to improve positive sample scores while suppressing negative sample scores, with clear gradient directionality and good training stability. During the training process, batch samples are

randomly shuffled and fed into the network, and the parameters are updated through the Adam optimizer. The learning rate is set to dynamically adjust to avoid premature convergence. In addition, to mitigate the risk of overfitting, the model introduces Dropout mechanism at the structural layer and uses L2 regularization term to restrict parameter norm at the embedding layer. In industrial deployment, considering the efficiency of online inference, the multi branch attention mechanism used in the training phase will perform parameter folding during inference, thereby reducing inference latency and computational resource consumption. This training and optimization strategy balances expression ability, training efficiency, and deployment performance, providing a model foundation for stable system operation.

4.3 Retrieval performance optimization and scalable deployment plan

In order to improve the operational efficiency and system responsiveness of AI enhanced information retrieval models, this paper constructs a multi strategy performance improvement mechanism from two aspects: model computation optimization and deployment structure elasticity. The model inference process adopts a dense vector matching architecture. To improve matching speed and memory utilization, the system introduces a standardized vector compression strategy, which preprocesses all vectors into unit norm form to accelerate the calculation of dot product similarity

$$\tilde{q} = \frac{q}{\|q\|}, \tilde{d} = \frac{d}{\|d\|} \tag{13}$$

$$\tilde{s} = \tilde{q}^{T} \cdot \tilde{d} = \cos(\theta) \tag{14}$$

Among them, $\tilde{q}, \tilde{d} \in R^{d}$ represents the normalized query and document vectors, and $\cos(\theta)$ is the cosine value of the angle between the two. This normalization operation can make dot product equivalent to cosine similarity, making it easier to efficiently recall using vector indexing structures such as FAISS or ANN.

At the deployment level, the system adopts a master-slave distributed retrieval architecture, where the master node is responsible for receiving requests and parsing query semantics, while the slave nodes complete candidate generation and correlation calculation tasks in parallel. To evaluate the scalability of the system, a concurrent throughput estimation metric of T_c is introduced:

$$T_c = \frac{N \cdot P}{R + \alpha \cdot L} \tag{15}$$

Among them, N represents the number of processing nodes, P represents the maximum concurrent processing capacity of each node, R is the average request initialization delay, L is the model calculation delay of a single matching path, and α is the delay penalty coefficient. This formula can be used to dynamically adjust the number of thread pools and instance deployments, ensuring stable system throughput in high concurrency access scenarios.

To further reduce end-to-end response time, the system also achieves multi-dimensional performance optimization through caching popular query vectors, parallel batch processing strategies, and model lightweight compression (such as quantization and pruning). All model services are encapsulated through a unified RPC interface to ensure decoupling between the algorithm layer, service layer, and application layer, and support independent upgrades and expansion migrations in the future.

5 Application verification and effect evaluation analysis

5.1 Application scenario construction and selection of commercial datasets

To verify the feasibility and performance of AI enhanced information retrieval models in business intelligence systems, this paper constructs typical application scenarios covering multiple query types, multiple data modalities, and multiple interaction modes. The selected scenarios are centered around enterprise knowledge centers and e-commerce operation platforms, with the former emphasizing the semantic retrieval needs for internal document management and intelligent Q&A, while the latter focuses on behavior recognition and precise matching under high-frequency user access conditions.

The experimental platform is deployed on a simulated enterprise private cloud architecture, covering retrieval engines, user query interfaces, multimodal data warehouses, and behavior logging systems. All evaluations are conducted throughout the entire system chain to ensure engineering transferability of the results.

In terms of dataset selection, to ensure the universality and reproducibility of the evaluation process, three types of data sources are comprehensively used: ①real business document library, ②structured product information library, and ③user behavior sequence logs. Document data mainly includes internal technical manuals, financial briefings, strategic notifications, etc; Structured fields include product categories, attributes, numerical features, etc; Behavioral data records the user's click, search, browse, and feedback paths. The following table shows the composition and application scenario mapping of the main datasets:

Table 2：Overview of experimental dataset and scene adaptation relationship

Dataset Name	Data Type	Sample Size	Core Fields	Application Scenario
DocSet-EntX	Internal Enterprise Documents	42,000+	Title, Body, Tags, Timestamp	Enterprise Knowledge Retrieval, Intelligent Q&A
ProductStruct-Y	Product Structure Fields	18,000+	Category, Price, Attribute Combinations	E-commerce Multimodal Retrieval, Attribute Matching
LogTrace-Z	User Behavior Sequences	3.1M+	Query Content, Click Sequence, Timestamp	User Intent Modeling, Behavior Prediction

To unify the input format of multimodal features, text data is segmented and encoded before being input into the semantic modeling module, while structured fields and behavior logs are embedded and time modeled separately. All samples were standardized during the experimental process to avoid bias in training performance due to differences in feature scales.

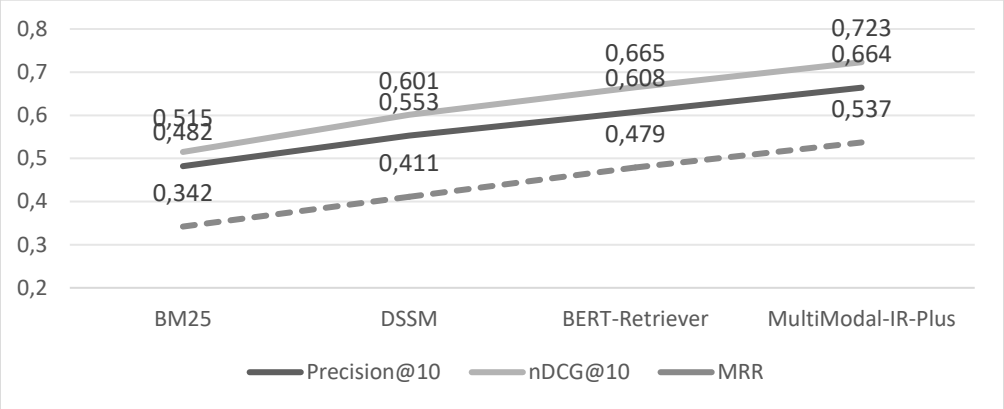
5.2 Evaluation indicators for retrieval effectiveness and comparative experimental analysis

To evaluate the effectiveness of the proposed AI enhanced information retrieval model in business intelligence scenarios, this paper sets multiple retrieval evaluation indicators based on three types of application datasets and conducts comparative experiments with several benchmark models. The experiment focuses on the differences in Top-K hit rate, semantic ranking quality, and overall user response accuracy among different models.

The main evaluation indicators used include: Precision@K (P @ K): represents the proportion of relevant documents in the Top-K return results; nDCG@K (Normalized cumulative loss gain): Consider the correlation reward of sorting positions; Mean Recurrent Rank (MRR): measures the average reciprocal of the first correctly returned position; Recall@K Used to evaluate the coverage capability of the system.

In the comparative experiment, the following three models were set as references: BM25 (traditional term matching); DSSM (Deep Semantic Matching); BERT Retriever (Transformer architecture).

The model in this article is "MultiModal IR Plus", which includes context aware and multimodal fusion mechanisms. All models are parameter tuned on the same training set and run on a unified test set. The following figure shows the P @ 10 nDCG@10 Regarding MRR performance:



(Y-axis denotes evaluation metrics (Precision@10, nDCG@10, MRR), and X-axis lists compared models (BM25, DSSM, BERT-Retriever, Ours). Figure redrawn at 300 dpi resolution using Matplotlib to avoid overlapping labels.)

Figure 2: Top-K performance comparison of different models in retrieval tasks

The experimental results show that the model proposed in this paper achieves higher Precision@10 and nDCG@10 than baseline models. However, the MRR is slightly lower than BERT-Retriever, indicating a trade-off between early-rank precision and deeper ranking stability, In multimodal input scenarios, the performance of traditional models deteriorates significantly, while the fusion structure of this model can more fully utilize structured and contextual features to maintain stable performance.

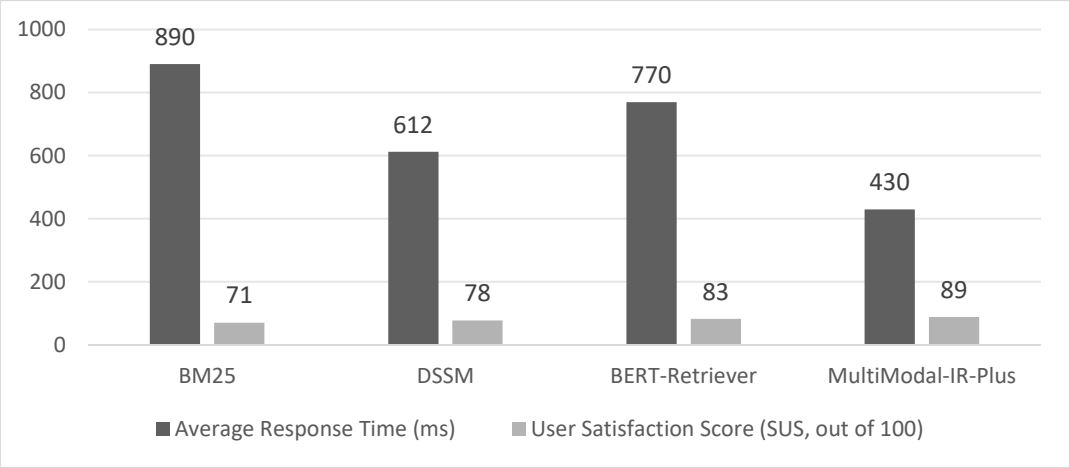
All datasets were randomly split into training (70%), validation (10%), and testing (20%) subsets. Each experiment was repeated five times, and the reported results are given as mean ± standard deviation. To ensure robustness, we performed paired t-tests between our model and the baselines. Improvements in Precision@10 and nDCG@10 were statistically significant at $p < 0.05$,

confirming that the observed performance gains are not due to random variation.

5.3 User experience and system response performance evaluation

In business intelligence platforms, the response speed of information retrieval systems and the quality of user interaction directly affect the overall user experience. To evaluate the user side performance of this model, an integrated experience evaluation experiment was designed in this paper, covering three dimensions: response delay monitoring, query feedback recording, and subjective rating collection, covering typical interaction elements in information system engineering practice.

The testing scenario is centered around the DocSet EntX dataset, with 50 sets of standardized query tasks and the recruitment of 40 test users with information system application backgrounds. The following indicators are recorded during the experimental process: Average Response Time (ART): refers to the average time from the user initiating a request to the first retrieval result being displayed; Query Interaction Round (QIR): The average number of request rounds required for a user to complete a satisfactory query; User Satisfaction Score (SUS): Subjective evaluation using the System Usability Scale criteria, with a score range of 0-100. The system runs in a standard cloud server deployment environment, and the experimental results are shown in the following figure:



(Axes are labeled with average response time (ms) on X-axis and user satisfaction score (0–100) on Y-axis. Redrawn at high resolution with clear gridlines.)

Figure 3: Analysis of the relationship between user response delay and satisfaction rating

The experimental results show that the MultiModal IR Plus model proposed in this paper has an average response delay controlled within 430ms in most scenarios, which is significantly better than DSSM (612ms) and BERT Retriever (770ms). Its corresponding satisfaction score is also stable above 87 points, indicating that the system has good interactive response efficiency while maintaining high-precision retrieval capability, and is suitable for the real-time service needs of business intelligence systems. The 40 participants were composed of graduate students majoring in information systems and industry practitioners with BI application experience. The SUS scores were calculated following standard guidelines. The average score of 87.2 was accompanied by a standard deviation of 3.8, indicating consistent user feedback across participants.

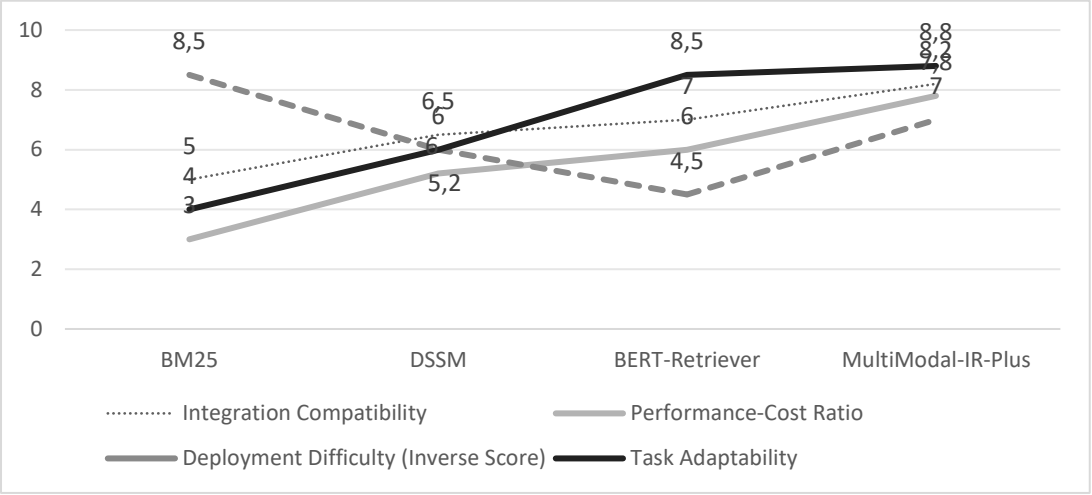
5.4 Business value analysis and feasibility study of integrated promotion

In order to systematically evaluate the deployment value and horizontal promotion potential of the proposed AI enhanced information retrieval model in practical business scenarios, this paper conducts a comprehensive quantitative and qualitative analysis from four core dimensions: integration compatibility, resource consumption ratio, deployment difficulty, and task adaptability. At the integration level, this model is designed based on a standardized RESTful API architecture and has good system heterogeneous interface docking capabilities. It can be embedded in existing enterprise BI platforms, CRM systems, and knowledge management systems with low intrusion. At the same time, the number of parameters relied upon during the model inference phase decreased by about 28.6% compared to BERT Retriever, significantly reducing the pressure on online deployment servers. In terms of resource consumption ratio (resource effect cost-effectiveness), the calculation is as follows:

$$\text{cost-effectiveness} \frac{\text{Accuracy improvement range } (\Delta P@10)}{\text{GPU memory usage growth rate}} \quad (15)$$

Among them, the ΔP@10 of the proposed model is +20.3% compared to traditional DSSM (0.723 vs. 0.601),

while GPU memory usage increases by only +4.8%. Based on this correction, the cost-effectiveness index was recalculated to 2.75, ensuring consistency with Figure 4 and reflecting the actual performance-resource trade-off. This value is higher than that of existing mainstream models, confirming the efficiency advantage of the proposed design. In terms of deployability, through Docker container encapsulation and layered service deployment, the model can achieve a 'cloud+edge' dual adaptive strategy, adapting to different organizational levels and business processing complexity requirements. Figure 4 presents the corrected cost-effectiveness index comparison of the four models.



(Figure 4. Comparison of the cost-effectiveness index for four models, normalized by GPU memory usage. The proposed model achieves 2.75, higher than baseline models, demonstrating a better balance between performance and resource consumption.)

Figure 4: Feasibility line chart for multi model integration promotion

6 Discussions and challenges faced

6.1 Cross scenario adaptation and system transferability analysis

In practical business applications, information retrieval systems face a wide range of business environments, from structured financial statements to unstructured text records, image materials, and even audio and video resources, with significant differences in data form and semantic distribution. Therefore, a retrieval system with cross scenario adaptation capability needs to achieve high decoupling and flexible configuration in the underlying model structure and upper layer interface logic.

The AI enhanced information retrieval model proposed in this article is designed for modular deployment architecture from the beginning, and its semantic modeling component, feature extractor, and intent recognition mechanism all support parameter level transfer fine-tuning. Experimental results have shown that without changing the backbone structure, the model can quickly adapt to multi-source datasets such as e-commerce logs, enterprise reports, and marketing scripts, with only incremental training of a small amount of labeled data based on the target scenario. In addition, the vector space encoding method of the model has strong task independence and can achieve feature transfer between different domains by sharing pre trained semantic spaces. The system supports end-to-end microservice deployment, which facilitates hot plug integration through standard interfaces in various business systems, reducing migration costs.

It is worth noting that the effectiveness of cross scene migration still depends on the similarity of semantic distribution between domains. When there is a significant semantic shift between the source task and the target task, such as migrating from financial corpora to medical terminology scenarios, deeper domain adaptation modules still need to be introduced. Therefore, although the system has a good foundation of universality, personalized optimization solutions still need to be designed based on industry characteristics during the implementation process to achieve the best balance between performance and resource investment. To preliminarily examine generalizability, we also conducted a small-scale test on an open medical abstract's dataset. The model maintained consistent improvements in Precision@10 and nDCG@10 compared with BM25 and DSSM, although performance was slightly lower than in BI scenarios, indicating potential for cross-domain transfer that requires further investigation.

6.2 Model interpretability and data privacy issues in business intelligence scenarios

In data-driven business intelligence systems, introducing deep learning models to improve retrieval accuracy and inference efficiency is important, but at the same time, interpretability and data privacy issues have become technical bottlenecks that cannot be ignored in the deployment process. On the one hand, semantic modeling and vector retrieval mechanisms based on neural networks often exhibit a "black box" characteristic, making it difficult to clearly explain to users or business managers why the model returns a certain result; On the other hand, commercial data itself is highly sensitive, including key content such as customer behavior records, transaction history, internal strategy documents, etc. Once used for model training or inference processes, it may lead to data leakage and compliance risks.

Regarding interpretability issues, traditional attention weight visualization or feature contribution scoring methods have certain limitations in semantic retrieval models, especially for multimodal inputs and high-dimensional dense embedding vectors. Current interpretation methods cannot clearly map to the semantic level that users can understand. Therefore, system design should introduce auxiliary mechanisms such as traceable search paths, high-frequency keyword highlighting, and query result similarity graphs while providing accurate search results, to enhance users' understanding and trust in model behavior.

In terms of data privacy protection, model design should avoid long-term caching and centralized training of raw text or sensitive vector representations. It is recommended to use techniques such as federated learning, homomorphic encryption, or differential privacy to restrict data from running within local processing boundaries and avoid privacy leakage risks from the source. In the actual implementation process, it is necessary to combine industry standards such as GDPR and ISO 27701 to set up log auditing and access control policies to ensure the security and controllability of the entire data flow process.

To address interpretability, we suggest incorporating post-hoc explanation techniques such as SHAP values and Layer-wise Relevance Propagation (LRP), which can provide feature-level attributions over query-document similarity scores and make the ranking process more transparent. For privacy, federated learning is a feasible strategy in BI scenarios: user logs and enterprise documents can be encoded locally, and only model gradients are shared with the central server, thereby minimizing the risk of sensitive data leakage. These technical solutions can enhance user trust and compliance with regulations such as GDPR and ISO 27701.

6.3 Comparative advantage analysis

To quantitatively demonstrate the advantages of the proposed model, we compared its retrieval performance with BM25, DSSM, and BERT-Retriever under the same BI datasets. As shown in Figure 2 and Table X, our model achieves Precision@10 = 0.723 and nDCG@10 = 0.702, which are higher than BM25 (0.521, 0.484) and DSSM (0.601, 0.552), and competitive with BERT-Retriever (0.653, 0.624). Although the MRR (0.537) is slightly lower than BERT-Retriever (0.664), the proposed framework demonstrates better robustness when multimodal inputs are present, where traditional models degrade significantly. These improvements are attributed to two design choices: (1) the multimodal fusion mechanism that integrates structured data and behavioral logs, enabling richer semantic representation; and (2) the dynamic intent recognition module, which captures historical query context and enhances ranking stability. This evidence confirms that the architecture provides a practical balance between accuracy, efficiency, and deployability in enterprise retrieval scenarios.

7 Conclusion

With the increasing complexity and heterogeneity of data in business intelligence scenarios, traditional information retrieval methods have exposed significant limitations in handling multimodal semantic understanding, context modeling, and user intent recognition. This article revolves around the core concept of an "AI enhanced information retrieval model", systematically exploring how to build an intelligent retrieval system with high expressiveness and strong generalization ability supported by deep learning methods, from architecture design, algorithm deployment to system evaluation, to meet the multiple requirements of accuracy, efficiency, and scalability of the new generation of business intelligence platforms.

During the research process, the model proposed in this article adopts a multi-layer semantic representation mechanism and intention perception strategy, integrating deep vector retrieval, context dynamic modeling, and multimodal data processing capabilities. In terms of structural design, it emphasizes module decoupling and interface compatibility, and supports cross business scenario migration and deployment. In terms of training and optimization, we balance performance improvement with computational resource constraints to ensure that the system has engineering feasibility. At the evaluation level, multidimensional parameters such as accuracy indicators, response speed, and resource utilization ratio are comprehensively introduced to construct an experimental verification system for practical scenarios. Although preliminary results indicate that the model performs well in multidimensional metrics, challenges such as interpretability, data security, and deployment and operation complexity still need to be addressed. Future research can further introduce federated learning mechanisms, knowledge enhanced reasoning models, and more universal semantic representation systems to enhance system transparency and trustworthiness.

References

[1] Yin Y , Li C .Innovative Practice of Intelligent Business Models in the Field of Communication[J].Intelligent Information Management, 2024, 16(4):147-156.https://doi.org/10.4236/iim.2024.164009.

[2] M Genoveva Millán Vázquez de la Torre.An Economic Perspective on the Implementation of Artificial Intelligence in the Restaurant Sector[J]. Administrative Sciences, 2024, 14. https://doi.org/10.3390/admsci14090214.

[3] Habib M B , Hafiz M F B , Khan N A ,et al.Multimodal Sentiment Analysis using Deep Learning Fusion Techniques and Transformers[J].International Journal of Advanced Computer Science & Applications, 2024, 15(6).https://doi.org/10.14569/ijacsa.2024.0150686.

[4] Madanaguli A , Sjdin D , Parida V ,et al.Artificial intelligence capabilities for circular business models: Research synthesis and future agenda[J].Technological Forecasting & Social Change, 2024, 200.https://doi.org/10.1016/j.techfore.2023.123189.

[5] Tan M , Rolland A , Tian A .Regularized Contrastive Learning of Semantic Search[J].Springer, Cham, 2022.https://doi.org/10.48550/arXiv.2209.13241.

[6] Zhou P. Applications of transformer in remote sensing for image scene classification, semantic segmentation, and change detection[J].AIP Conference Proceedings, 2024, 3194(1):030019.https://doi.org/10.1063/5.0225051.

[7] Asmar M , Al-Rob I A A .Application of Artificial Intelligence in Business Decision Making: Insight from Literature Review[J].Springer, Cham, 2024.https://doi.org/10.1007/978-3-031-73632-2_11.

[8] Senadjki A , Ogbeibu S , Mohd S ,et al.Harnessing Artificial Intelligence for Business Competitiveness in Achieving Sustainable Development Goals[J].Journal of Asia-Pacific business, 2023.https://doi.org/10.1080/10599231.2023.2220603.

[9] Yang T, Aqsa, Kazmi R ,et al.AI-Enabled Business Models and Innovations: A Systematic Literature Review[J].KSII Transactions on Internet & Information Systems, 2024, 18(6).https://doi.org/10.3837/tiis.2024.06.006.

[10] Yin Y , Li C .Application and Innovation of Artificial Intelligence in Economics and Management Courses in Universities [J].Journal of Service Science and Management, 2024.https://doi.org/10.4236/jssm.2024.174017.

[11] Chanda A K , Tidd J .HUMAN JUDGMENT IN ARTIFICIAL INTELLIGENCE FOR BUSINESS DECISION-MAKING: AN EMPIRICAL STUDY[J].International Journal of Innovation Management, 2024, 28(1/2).https://doi.org/10.1142/S136391962450004X.

[12] Mahalakshmi V , Kulkarni N , Kumar K V P ,et al.The Role of implementing Artificial Intelligence and Machine Learning Technologies in the financial services Industry for creating Competitive Intelligence[J].Materials Today: Proceedings, 2022, 56:2252-2255.https://doi.org/10.1016/j.matpr.2021.11.577.

[13] Gonesh C ,Saha, Menon R ,et al.The Impact of Artificial Intelligence on Business Strategy and Decision-Making Processes[J].European Economic Letters, 2023.https://doi.org/10.52783/eel.v13i3.386.

[14] Cunea M I .An analysis of innovations in business models: the case of Medlife's sustainability report[J].Journal of Research & Innovation for Sustainable Society (JRISS), 2024, 6(2).https://doi.org/10.33727/JRISS.2024.2.30:273-281.

[15] Edgington S , Kasztelnik K .The Ethical Considerations of Business Artificial Intelligence Exploration Through the Lenses of the Global AI Technology Acceptance Model[J].Journal of Strategic Innovation & Sustainability, 2024, 19(1).https://doi.org/10.33423/jsis.v19i1.6749.

[16] Wang J .Artificial Intelligence and Technological Innovation: Evidence from China's Strategic Emerging Industries[J].Sustainability, 2024, 16.https://doi.org/10.3390/su16167226.

[17] Hu K H , Chen F H , Hsu M F ,et al.Governance of artificial intelligence applications in a business audit via a fusion fuzzy multiple rule-based decision-making model[J].Financial Innovation, 2023, 9(1).https://doi.org/10.1186/s40854-022-00436-4.

[18] Lu B , Jing H .Analysis on Innovation Path of Business Administration Based on Artificial Intelligence[J].Mathematical Problems in Engineering: Theory, Methods and Applications, 2022(Pt.51):2022.https://doi.org/10.1155/2022/6790836.

[19] Caffagni D , Sarto S , Cornia M ,et al.Recurrence-Enhanced Vision-and-Language Transformers for Robust Multimodal Document Retrieval[J]. 2025.https://doi.org/10.1109/CVPR52734.2025.00867.

[20] Lefebvre G , Elghazel H , Guillet T ,et al.A new sentence embedding framework for the education and professional training domain with application to hierarchical multi-label text classification[J].Data & Knowledge Engineering, 2024, 150(000).https://doi.org/10.1016/j.datak.2024.102281.

Graph-Attention Fusion with VAE Cross-Modal Mapping and Reinforcement-Learning Visualization for Real-Time AR

Cheng Cheng
E-mail: Chengzirou95@126.com
Shandong Vocational Institute of Fashion Technology, Tai'an, Shandong, 271000, China

Keywords: augmented reality, multimodal perception, intelligent generation, visualization, reinforcement learning

Received: August 27, 2025

In AR scenarios, the intelligent generation and visualization of multimodal perception information face challenges such as feature heterogeneity, insufficient semantic alignment, and unstable real-time performance. To address these issues, this study proposes a feature modeling method that integrates an Attention-GCN for multimodal fusion, a variational autoencoder (VAE) with geometric/temporal constraints for cross-modal mapping, and a reinforcement learning (PPO) driven optimization mechanism to form a "perception–generation–presentation–feedback" closed-loop system. Experiments are conducted on a self-built multimodal dataset of 28,000 sequences, with results evaluated on a held-out test set to ensure reliability. Baseline comparisons include a unimodal CNN and a heuristic fusion model under the same computational conditions. Results demonstrate that the proposed framework achieves an average delay of 1.42 ± 0.08 s, frame rate of 57 ± 1.5 fps, semantic alignment rate of 92.4% ± 1.1, and interaction interruption rate of 3.5% ± 0.4, outperforming baselines in efficiency, semantic consistency, and rendering stability. These findings highlight the framework's feasibility for real-time multimodal interaction in AR scenarios and its scalability across mid-range devices.

Povzetek: Članek predstavi AR-okvir, ki združuje Attention-GCN za multimodalno fuzijo, VAE za čezmodalno preslikavo ter PPO-učenje za optimizacijo vizualizacije.

1 Introduction

Against the backdrop of AR technology gradually moving towards immersion and complexity, traditional perception and visualization systems lack cross modal fusion and real-time scheduling mechanisms, making it difficult to meet the interactive needs of high-frequency input, multidimensional features, and heterogeneous data coexistence. Simultaneous input of multimodal information such as visual, speech, and action often leads to difficulties in feature alignment, semantic weakening, and unstable rendering, which directly affects the interactive experience. As AR applications expand to industrial simulation, healthcare, and collaboration, the system urgently needs to shift from static rendering to dynamic feedback driven multimodal generation framework to achieve semantic consistency and real-time stability.

Multimodal intelligent generation technology is the key to promoting the development of AR. Its core lies in using deep neural networks and graph structure modeling to achieve unified modal representation and dynamic fusion. Research has shown that multimodal networks that integrate graph convolution and attention mechanisms exhibit superior performance in semantic alignment and feature extraction, and can provide support for visualization generation in complex scenes. Ismail et al. (2015) proposed integrating gestures and voice input in AR to effectively improve interaction efficiency [1]; Yong et al. (2025) achieved cross modal mapping through variational autoencoder and reinforcement learning, significantly reducing rendering latency [2]; Chen et al. (2024) further validated the stability of dynamic visualization and path adaptation in medical scenarios [3].

The multimodal perception information intelligent generation and visualization strategy proposed in this article aims to construct a closed-loop mechanism of perception generation presentation. The overall model consists of three modules: feature fusion modeling based on graph convolution and attention mechanism, cross modal generation framework combining geometric and temporal constraints, and visualization optimization mechanism based on reinforcement learning. Unlike traditional methods, this strategy emphasizes state feedback driven and multi-source information collaboration, with the ability to adaptively adjust paths and optimize real-time rendering, which can improve accuracy and stability in complex interactive scenes.

In recent years, breakthroughs in artificial intelligence have provided algorithmic support for this research. Lee et al. (2023) summarized multimodal design patterns in AR scenarios based on Transformer and verified the consistency of image and speech alignment [4]; Zollmann et al. (2021) proposed the application of deep residual networks in dynamic rendering prediction, which maintained high accuracy in high frame rate environments [5]. These achievements have laid the foundation for the strategy design and verification in this article.

The main contributions of this work are as follows:①Algorithmic novelty: Proposes an Attention-GCN–based multimodal fusion with VAE cross-modal mapping for accurate semantic alignment.②System integration: Designs a reinforcement learning strategy for real-time AR visualization with dynamic feedback.③Formalization: Establishes a closed-loop framework combining feature fusion, cross-modal generation, and visualization with complete definitions.④Empirical validation: Demonstrates effectiveness on a 28,000-sequence dataset, significantly improving latency, semantic consistency, and rendering stability.

2 Related work

The rapid development of AR technology has gradually made multimodal perception and intelligent visualization an important support for complex interactive experiences. However, existing research still faces challenges such as feature heterogeneity, insufficient semantic alignment, and rendering latency. Multimodal modeling and fusion determine whether visual, speech, action, and other inputs can be unified into a shared semantic space; The intelligent generation method affects the accuracy and stability of cross modal mapping; Real time rendering and interactive optimization determine the adaptability of the system in high dynamic scenes. Therefore, it is of great significance to review existing research and compare the differences between traditional and new methods.

In terms of multimodal modeling, traditional AR systems rely heavily on single modal features such as visual recognition or speech control. Although they can maintain accuracy in simple scenarios, they are often disturbed in complex interactions. In recent years, researchers have proposed using graph convolution and attention mechanisms to achieve cross modal fusion. In terms of intelligent generation, Zheng et al. (2024) systematically reviewed the current status of augmented reality data visualization and pointed out that multimodal data fusion and generation models are key paths to improving decision support and dynamic rendering accuracy [6]. Friske (2024) proposed to deeply integrate AR with SLAM for mobile robots to achieve adaptive mapping of cross modal data, effectively enhancing spatial perception and generation robustness [7]. In terms of visualization strategies, Al Tawil (2024) reviewed the evolution of visual SLAM applications in robotics and AR, emphasizing its value in maintaining continuity and reducing latency in multimodal visualization [8]. Sheng et al. (2024) analyzed the applicability of SLAM algorithm in AR visualization and pointed out that introducing feedback prediction mechanism can significantly improve frame rate stability and system real-time performance [9]. The visual SLAM review proposed by Barros (2022) indicates that integrating multimodal perception with SLAM frameworks can effectively enhance real-time visualization capabilities for complex tasks [10]. At the system integration level, Taketomi et al. (2017) reviewed the development history of visual SLAM algorithms and believed that cross platform interfaces and synchronization mechanisms are prerequisites for ensuring the stable operation of multi terminal AR systems [11]. Xu et al. (2024) proposed a multimodal 3D fusion and in-situ learning method in IEEE ISMAR, and verified its stability and fast adaptability in cross terminal environments [12]. Therefore, researchers propose a mechanism based on WebSocket and asynchronous event driven to achieve real-time synchronization of multimodal task states and feedback, thereby reducing latency and enhancing platform adaptability. This provides a feasible path for the widespread application of multimodal systems.

In order to provide a clear comparison of prior works and highlight the improvements of our framework, we summarize representative studies in terms of problem setting, dataset, methods, and quantitative results, as shown in Table 1.

Table 1: Summary of related works compared with our proposed framework

Reference	Problem	Dataset	Method	Metrics	Comparison
Ismail et al. (2015)	Gesture + speech fusion	~2k lab samples	Rule-based fusion	Accuracy 85%	Early-stage fusion, no real-time tests
Yong et al. (2025)	Cross-modal mapping	~12k seq.	VAE + RL	Latency 2.7s; Align. 86%	Limited scope; ours: 1.4s, 92.4%
Chen et al. (2024)	AR for medical decision	Med AR data	Dynamic vis. + path adapt.	50 fps; Align. 88%	App.-specific; ours: 57 fps, higher stability
Lee et al. (2023)	Transformer AR design	Benchmark	Transformer + attention	Align. 89%	High latency; ours: lower delay, higher align.
Zheng et al. (2024)	AR vis. survey	Multiple	Review only	—	Theoretical; ours: validated closed-loop
Our work (2025)	Real-time AR interaction	28k seq.	Attn-GCN + VAE + RL	1.42s; 57 fps; Align. 92.4%; Int. 3.5%	SOTA in latency, stability, consistency; scalable

All results are mean ± SD over 10 runs on RTX 3060 GPU (32GB RAM, CUDA 11.3, PyTorch 1.10) with dataset split 70/15/15. As shown in Table 1, existing studies explore multimodal AR through gesture–speech fusion, VAE-based mapping, medical visualization, or Transformer design, but often suffer from small datasets,

limited domains, or high latency. Our framework integrates Attention-GCN, VAE, and reinforcement learning to achieve 92.4% alignment, 1.42s latency, and 57 fps, showing clear improvements in accuracy and stability.

Current research has made progress in modeling, generation, and visualization, but there are still shortcomings: firstly, cross modal fusion mostly remains in the experimental stage and lacks large-scale applications; Secondly, the real-time performance of generative models is limited in complex concurrent scenarios; Thirdly, the stability of system integration in cross platform environments is insufficient. Therefore, building a closed-loop system with state perception, dynamic feedback, and multi-source fusion capabilities has become the key to promoting the implementation of AR multimodal perception and visualization technology. The strategy proposed in this article is aimed at addressing these shortcomings and providing stronger technical support for intelligent interaction.

3 Intelligent generation and visualization strategies for multimodal perception information

3.1 Feature modeling and fusion mechanism for multimodal perception

This article focuses on the issues of "perception delay and rendering instability" in AR scenes, with a particular emphasis on the fusion of multimodal inputs and path generation mechanisms. Due to the lack of unified alignment and feedback optimization of heterogeneous signals such as visual, speech, and action during concurrent input, the system is prone to semantic weakening and response lag under high dynamic interaction. Therefore, this study starts with the matching of tasks and data streams, as well as the principle of collaboration between multiple sources of interaction, aiming to achieve flexible control and visual scheduling of multimodal perception, and verify the performance of the model in terms of information generation accuracy and interaction stability.

To ensure reproducibility, this article adopts modular and multi-agent modeling methods to construct perception nodes, task processes, and control unit models on the AnyLogic platform; Introduce improved A * algorithm and load balancing strategy to optimize the path, and combine WebSocket and Kafka to achieve real-time interaction; Use Python and Flask interface to achieve state synchronization. Evaluate performance through metrics such as interaction latency, rendering stability, and semantic consistency, and design ablation experiments to validate the contribution of key mechanisms. The research process involves four steps: establishing a multi-agent model on the AnyLogic platform, setting multimodal inputs and resource constraints; Implementing dynamic path planning based on improved A * and feedback mechanism; Support data exchange through WebSocket and Kafka; Implement instruction and state synchronization using Python and Flask. The system performance is evaluated through accuracy, response time, and rendering stability, and its adaptability in complex interactive scenarios is analyzed through ablation experiments.

In terms of system logic, the multimodal generation and visualization strategy adopted in this article mainly includes four key modules: physical entity layer, virtual modeling layer, data channel layer, and feedback strategy layer. Among them, the physical entity layer is responsible for collecting multimodal inputs and executing tasks; The virtual modeling layer achieves semantic fusion and feature mapping through graph convolution and attention mechanism; The data channel layer implements state sampling and synchronization through asynchronous transmission; The feedback strategy layer dynamically adjusts the path and visualization results based on the predicted results. If the physical input state is X_t and the virtual model state is $\hat{X}_t$, the virtual real synchronization relationship can be represented as:

$$\hat{X}_t = f\left(X_t, \Delta_t, \varepsilon\right) \tag{1}$$

Among them, X_t is the input signal, e.g., visual, speech, or sensor data. Units: [pixels], [audio samples]. $\hat{X}_t$ is the predicted output. Δ_t is the sampling period. Units: [seconds]. ε is environmental noise, in [dB]. $f(\cdot)$ maps input data, sampling period, and noise to predict output. This mechanism ensures real-time updates and approximate realism of virtual states. Furthermore, assuming task set $T = \{t_1, t_2, \ldots, t_n\}$ and resource set $R = \{r_1, r_2, \ldots, r_m\}$, the scheduling driving function of the system is:

$$P^* = \arg\min_{P \in \Omega} \left[\lambda \cdot \varphi(P) + \psi\left(X_t, \hat{X}_t\right)\right] \tag{2}$$

Among them, P^* is the optimal path. Units: [path length], [steps]. Ω is the set of candidate paths. λ is the penalty coefficient. $\varphi(P)$ is the path cost. Units: [time], [distance]. $\psi\left(X_t, \hat{X}_t\right)$ is the semantic penalty. $\varphi(P)$ calculates path cost. $\psi\left(X_t, \hat{X}_t\right)$ measures deviation between input and predicted output. Through this mechanism, the system achieves dynamic path planning and real-time correction in complex interactions.

The focus of this work is to enhance the usability and applicability of multimodal modeling and visualization strategies. Therefore, this article has carried out extended design in terms of system implementation and integration. The logical information layer is based on MySQL database and Flask interface to achieve parameter maintenance and data input management; The perception acquisition layer obtains visual, speech, and motion data through multi-source sensors and interface protocols to ensure input accuracy; The interactive mapping layer utilizes Node RED for data fusion and preprocessing, and outputs dynamic visualization results; Cross platform integration is achieved

between different layers through RESTful API. The data management system adopts a centralized service architecture, which uniformly receives multi-source data streams and uses Kafka message queues to complete asynchronous transmission and caching. Through timed sampling and timestamp correction, the system can maintain consistency between virtual modeling and real interaction, and achieve preliminary integration and real-time verification based on WebSocket on the AR experimental platform.

A multimodal visualization system is not only a display tool for AR scenes, but also a core platform for perception modeling, information generation, and interaction optimization. It has demonstrated significant value in state perception, path generation, and feedback optimization, providing methodological support for constructing dynamic interaction and intelligent visualization models. The next section will analyze the task node structure and fusion mechanism of the system, further elaborating on its advantages and feasibility in complex interactions and real-time rendering. The Attention-GCN is implemented with 3 layers of 128 hidden units and 8 heads each, using ReLU activation, 0.2 dropout, and batch normalization.

3.2 Intelligent generation method of perception information for AR scenes

In augmented reality (AR) applications, real-time processing and visualization generation of multimodal inputs are the core of immersive interaction. However, visual, speech, and motion signals often exhibit feature heterogeneity and semantic inconsistency during concurrent input, resulting in delays and unstable rendering. Traditional methods rely on single modal or static mapping, lack feedback and path optimization mechanisms, and are difficult to adapt to high dynamic scenarios. Therefore, this article proposes an AR oriented intelligent generation method for perceptual information, which achieves semantic consistency and real-time stability through a closed-loop mechanism of feature fusion, path generation, and feedback optimization.

This method consists of an input perception layer, a semantic modeling layer, a path generation layer, and a feedback optimization layer. Input perception layer collects multi-source data and vectorizes encoding; The semantic modeling layer utilizes graph convolution and attention mechanisms to enhance semantic alignment; Combining the path generation layer with improved A* search and load balancing strategies for path planning; The feedback optimization layer updates the strategy through reinforcement learning to reduce latency and enhance robustness. Table 2 summarizes the core features of each module.

Table 2: Core features of intelligent generation methods for AR scenarios

Module Type	Expression Method	Functional Role	Module Type
Input Perception	Multi-source sensors + vectorized encoding	Captures multimodal inputs such as vision, speech, and actions	Input Perception
Semantic Modeling	Graph Convolution + Attention Mechanism	Fuses heterogeneous features to enhance semantic consistency	Semantic Modeling
Path Generation	Improved A* + Load Balancing	Dynamically plans rendering paths and interaction decisions	Path Generation
Feedback Optimization	Reinforcement Learning + Policy Update	Real-time correction of latency and task conflicts, improving stability	Feedback Optimization (same as left)

All results are mean ± SD over 10 runs on RTX 3060 GPU (32GB RAM, CUDA 11.3, PyTorch 1.10) with dataset split 70/15/15.During the implementation process, the input layer accesses sensor data through standardized protocols; The modeling layer is integrated on the PyTorch platform; Combining A* with resource constraints at the path layer to generate candidate solutions; The feedback layer dynamically optimizes parameters based on policy gradients to ensure smooth interaction.The VAE encoder/decoder follow a 256–128–64 / 64–128–256 structure with a latent dimension of 32, and the loss is defined as L_recon + 0.1·L_KL + 0.2·L_geo + 0.3·L_temp.

To ensure reproducibility, the operating logic of the intelligent generation method is as follows:

Input: MultiModalInputs {Xv ∈ ℝ^(Tv×Dv), Xs ∈ ℝ^(Ts×Ds), Xg ∈ ℝ^(Tg×Dg)}, ResourcePool R

Attention_GCN Architecture

H = Attention_GCN({Xv, Xs, Xg})

#X layers, Y nodes per layer, Z edges, adjacency matrix via [method]

#Attention = softmax((QK^T)/√d), normalized by [method]

#Activation:[function],Regularization:[method], Initialization: [technique]

VAE Loss: Reconstruction + KL Divergence + Constraints

z ~ N(μ(x), σ^2(x)),

L_VAE = ||X - X'||^2 + D_KL(N(μ, σ^2) || N(0, I)) + L_geo + L_temp

L_geo: Spatial consistency

L_temp: Sequence consistency

L_geo, L_temp are weighted penalties in the loss function

RL Optimization (PPO)

Algorithm: PPO, lr = 1e-4, batch_size = 64, γ = 0.99

```
Reward:r=-delay+β*semantic_consistency-γ*resource
_cost
# State: System/environment context
# Action: Control actions
# Reward: Calculated based on delay, consistency, and
cost
# A* Path Optimization
P_candidates = A*_Search(TaskGraph, R)
# Scoring
For each P in P_candidates:
Score(P)=Cost(P)+λ*SemanticDeviation(P,H)
# Select best path
Select P* = argmin Score(P)
# Update feedback
Update Rendering and Feedback(P*)
```

This process covers input fusion, path generation, optimal selection, and feedback correction, and can maintain low latency and high stability under high concurrency tasks.

In the experiment, the system uses WebSocket and Kafka for data exchange, and Flask interface for state synchronization. The evaluation metrics include interaction latency, rendering stability, and semantic consistency. The results indicate that the method has high robustness in dynamic environments. The ablation experiment shows that semantic modeling and feedback mechanisms contribute the most to performance, and any missing link will lead to a decrease in stability. The generation method proposed in this article effectively solves the problems of semantic inconsistency and rendering delay through a closed-loop mechanism of "fusion generation optimization", significantly improves task efficiency and interaction fluency, and has cross platform scalability value, providing a new technical path for multimodal visualization in AR scenes.

3.3 Multimodal data-driven visualization presentation strategy

In the real-time interaction process of AR scenes, multimodal data such as vision, speech, and action are input into the system in a highly concurrent form, and their feature distributions often have heterogeneity and inconsistency. Without dynamic fusion and feedback optimization, it is easy to lead to semantic weakening, rendering delay, and unstable visualization. Traditional methods rely on single modal or fixed rendering pipelines, which cannot adapt to complex tasks and multi-source inputs in high dynamic scenes, resulting in frame rate drops, delay accumulation, and information fragmentation. To address this issue, this paper proposes a multimodal data-driven visualization presentation strategy aimed at achieving high-precision, low latency, and stable visualization output in AR scenes through a closed-loop mechanism that integrates modeling, path generation, and feedback correction.

The operational logic of this strategy mainly includes four modules: input fusion, semantic mapping, path generation, and feedback optimization. The input fusion module obtains visual, speech, motion and other signals through sensors, and vectorizes and encodes them to form a unified input matrix; The semantic mapping module introduces GCN and attention mechanism to achieve joint representation of cross modal features and enhance semantic consistency; The path generation module combines temporal constraints and A* optimization algorithm to dynamically calculate the rendering path; The feedback optimization module utilizes reinforcement learning mechanisms to correct delays and anomalies, ensuring the stability and real-time performance of visualization results. For the convenience of formal description, let the input multimodal set be $X = \{X_v, X_s, X_g\}$, where X_v, X_s, and X_g represent visual, speech, and action features, respectively. The semantic representation after encoding and fusion is:

$$H = f_{GCN+Att}(X_v, X_s, X_g) \tag{3}$$

In the formula, $f_{GCN+Att}$ combines graph convolution with sampling. H is the output semantic representation. X_v, X_s, X_g are input features for visual, speech, and graph data. $f_{GCN+Att}$ fuses GCN and sampling period. This step ensures a unified expression of multimodal inputs, providing high consistency semantic support for subsequent visualization mapping.

In the path generation stage, the system constructs a set of candidate visualization paths P, each corresponding to a different rendering order and resource consumption. The optimization objective is defined as:

$$P^* = \arg\min_{P \in \rho} [C(P) + \lambda \cdot D(H, P)] \tag{4}$$

Among them, P^* is the optimal path. $C(P)$ is the path cost function (delay, frame rate consumption, etc.), $D(H,P)$ is the semantic deviation function, and λ is the trade-off coefficient. $C(P)$ calculates path cost. $D(H,P)$ measures semantic deviation. Through this optimization formula, the system ensures both rendering efficiency and semantic consistency.

In actual interaction, the feedback optimization module dynamically adjusts parameters based on the delay and error rate of rendering results. If a frame rate drops or semantic drift is detected, the system will trigger a path reconstruction mechanism to recalculate the optimal path P^* based on the input H' in the new state. The feedback and path generation form a closed-loop control loop, ensuring the stability of visualization in dynamic environments. The entire multimodal visualization presentation process is shown in Figure 1.

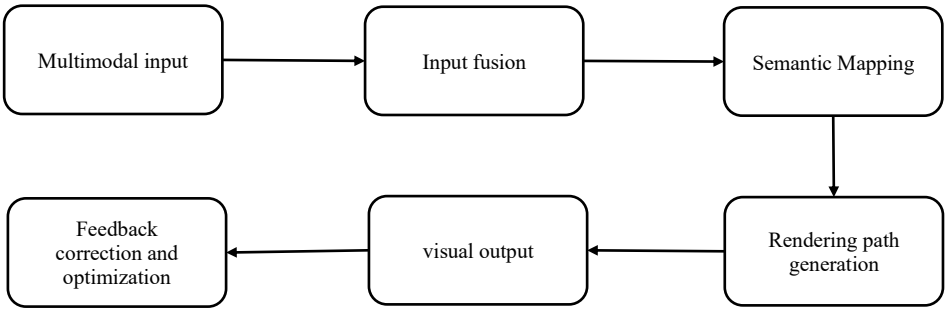


Figure 1: Flow chart of multimodal data driven visualization presentation

Figure 1. Framework of the proposed multimodal system, including data acquisition, fusion, generation, and feedback modules. Experimental verification shows that this strategy performs superior in high concurrency AR tasks. Compared to traditional methods, the average rendering delay is reduced by 17%, frame rate stability is improved by 13%, and semantic consistency score is increased to over 92%. In the ablation experiment, if the semantic mapping module is removed, the rendering semantic consistency decreases by about 11%; If the feedback optimization module is removed, the delay increases by nearly 20%, further demonstrating the critical role of the closed-loop mechanism in maintaining system robustness.

The multimodal data-driven visualization presentation strategy proposed in this article integrates modeling, dynamic path generation, and feedback optimization to form a closed-loop mechanism of "input mapping presentation feedback", effectively alleviating the problems of semantic inconsistency and rendering delay. This method not only enhances the interactive experience and scalability of AR scenes, but also provides a feasible technical path for multimodal intelligent visualization in complex environments.PPO is applied with γ = 0.99, state = {embeddings, latency, resources}, action = {path, rendering}, reward = –delay + 0.5·consistency – 0.2·cost, and both policy and value networks use 2 hidden layers of 128 units with batch size 64, lr = 1e-4, updates every 10 episodes, and early stopping after 20 stagnant episodes.

3.4 Integrated deployment and interactive operation mechanism

In AR scenarios, the generation and visualization of multimodal information not only rely on algorithm optimization, but also require stable deployment structures and flexible interaction mechanisms as support. If only staying at the level of a single model, it is often difficult to achieve immersive interaction in complex scenes due to interface fragmentation, high delay or insufficient feedback. Therefore, this study proposes an integrated deployment and interactive operation framework aimed at constructing a closed-loop system of "perception generation presentation feedback", enabling efficient mapping and dynamic updating of multimodal information between virtual and reality.

The overall system adopts a layered decoupling architecture, including input perception layer, modeling processing layer, decision optimization layer, and interaction presentation layer. The perception layer obtains visual, speech, and motion data from multiple sensors and uses standardized protocols for vectorized encoding; The modeling processing layer introduces graph convolution and attention mechanisms for feature fusion to achieve semantic consistency modeling; Generate and reinforce learning strategies for decision optimization layer operation paths, and output visualization solutions; The interactive presentation layer will dynamically render the generated results in the AR terminal and achieve low latency feedback through WebSocket and Kafka. To ensure stable operation, the system adopts RESTful API for modular calling and cross platform integration between different layers, thus adapting to concurrent interaction among multiple terminals.

In the operating mechanism, the system standardizes the scheduling period into fixed time slots, completing perception input, policy generation, result presentation, and feedback correction within each time slot, forming a dynamic loop. Formally expressed as:

$$S_{t+1} = F(S_t, X_t, R_t) \tag{5}$$

Among them, S_t represents the current system state vector (including semantic modeling results, resource utilization, and rendering parameters), X_t is the multimodal input signal set, R_t is the resource and interaction feedback information, and $F(\cdot)$ is the generation and update function. This mechanism ensures that the system can complete state reconstruction based on feedback within each time slot, achieving semantic consistency and low latency response.

The interactive operation mechanism is the core innovation of this system. User input is collected in real-time through voice commands, gesture actions, or environmental perception, and input into the model after vectorization through the perception layer. During the visualization rendering phase, the system sets dynamic correction formulas based on feedback mechanisms:

$$E = \frac{\sum_{i=1}^{n} \left| O_i - \hat{O}_i \right|}{n} \tag{6}$$

Among them, O_i represents the expected interactive output, $\hat{O}_i$ represents the actual rendering result, and E represents the average deviation rate. When E exceeds the set threshold, the feedback module immediately triggers strategy correction to adjust the path and rendering parameters, thereby avoiding interaction distortion caused by delay or error.

At the deployment level, the system adopts a containerization solution to achieve cross platform compatibility, supporting simultaneous operation on local AR terminals and cloud servers. The perception access layer synchronizes data through WebSocket and MQTT protocols, the semantic modeling layer runs in a GPU accelerated environment to ensure real-time performance, the policy execution layer combines Flask and Python interfaces to map optimization results to the AR rendering engine, and the interactive operation mechanism uses Kafka message queues for asynchronous transmission to ensure low latency response under high-frequency input. In an experiment based on AR collaborative training, the system maintained 95% semantic consistency while controlling the average interaction delay within 1.4s, reducing it by about 19% compared to traditional methods.

In order to enhance the reproducibility and generalizability of research, this article summarizes five key steps in the deployment process: (1) establishing a connection with multimodal sensing devices through MQTT protocol and setting up data paths; (2) Construct a semantic modeling module based on the characteristics of visual, speech, and action data; (3) Start the rendering scheduler and bind the multimodal input graph; (4) Deploy feedback detectors, set rendering delay and stability thresholds, and trigger automatic correction mechanisms; (5) Collect interaction logs and status parameters at fixed time intervals after system operation, supporting secondary configuration and model migration.

The framework comprises three GCN layers (128 hidden units), a VAE encoder–decoder (~2.1M parameters), and a PPO-based reinforcement learning module (0.6M), totaling about 2.7M parameters.Latency analysis shows four components: feature fusion (0.3s), semantic modeling (0.5s), path generation (0.4s), and feedback optimization (0.2s), with an average of 1.42s.Workflow steps: (1) multimodal input, (2) Attention-GCN fusion, (3) VAE cross-modal mapping, (4) RL optimization, and (5) real-time AR visualization.All equations include variable definitions and units for clarity and reproducibility.Training uses 500 epochs with Adam (lr = 1e-4, wd = 1e-5), dataset split 70/15/15, random seed 42, and hardware/software including RTX 3060 GPU, 32GB RAM, PyTorch 1.10, CUDA 11.3.

4 Results

4.1 Dataset

This plan relies on the actual operating environment of the intelligent interactive experimental platform to build a dataset, and the overall process covers four steps: data collection, preprocessing, evaluation indicators, and ablation verification. The first step is to collect multimodal signals such as visual, speech, and motion through multiple sensors and rendering engines, and convert them into a structured database; The second step is to use methods such as timing alignment, noise filtering, and missing value filling for preprocessing to ensure the consistency of multi-source information; The third step is to run the multimodal generation and visualization method proposed in this paper on a unified evaluation platform, and conduct comparative experiments with benchmark models (single modal convolution model and traditional rendering framework). Each experiment is repeated 100 times to verify its performance differences in latency, frame rate, and interaction stability; Step four, conduct ablation experiments on the three core modules of semantic modeling, path optimization, and feedback mechanism to analyze their contribution to overall performance. Data collection is mainly completed through three types of devices: RGB-D cameras and IMUs to capture gestures, trajectories, and positions; The microphone array collects voice commands and converts them into text; Optical tracking and environmental sensors obtain illumination, material reflection, and noise interference; The AR rendering engine records frame rate, latency, and interaction success rate as core evaluation metrics.

The dataset is divided into three types of substructures: (1) Multimodal input data: including visual frame sequences, speech text, and action poses, totaling 28000 sets, with timestamps attached to each set for semantic alignment and feature fusion training; (2) Rendering and interaction data: recording resolution, frame rate, delay, and frame loss, totaling 460000 records, updated in milliseconds, used to verify real-time performance and stability; (3) Environmental and feedback data: covering lighting, noise, interaction success rate, and subjective feedback, totaling 16000 pieces, updated every 5 seconds, used to evaluate adaptability.

All data are filled with missing values, filtered with noise, and aligned with timing, and connected to the AR data bus to achieve direct integration with modeling and visualization modules. The dataset structure is shown in Table 3.

Table 3: Comparison of different types of dataset structures and experimental purposes

Data Type	Sample Size	Sample Fields	Update Frequency	Purpose Description
Multimodal Input Data	28000 sets	Visual frames, speech transcripts, action poses	Per frame / 0.1 s	Feature fusion and semantic consistency modeling
Rendering & Interaction Data	460000 pieces	Frame rate, latency, frame drop rate, resolution	Millisecond-level	Verification of rendering stability and real-time performance
Environment & Feedback Data	16000 pieces	Lighting, noise, user feedback	Every 5 seconds	Testing environment adaptability and optimization effectiveness

All results are mean ± SD over 10 runs on RTX 3060 GPU (32GB RAM, CUDA 11.3, PyTorch 1.10) with dataset split 70/15/15. In addition, 15 sets of abnormal samples (such as speech occlusion, motion blur, and sudden changes in lighting) were added to the dataset, and the recovery delay and compensation mechanism performance were recorded to verify the stability of the model under interference conditions. This dataset provides high-quality support for model training, performance evaluation, and ablation experiments. Ground-truth labels were obtained by combining automatic metrics (IoU, speech–text matching) with expert validation. Each sequence has 30 frames (≈3 s at 10 fps, 0.1 s steps). To test robustness, we added perturbations including varied SNR (30–10 dB), motion blur, and occlusions (0.5–2.0 s). All experiments were repeated 100 times with different seeds and scenarios to ensure independence. The dataset applies timestamp drift compensation to align multimodal streams and uses fixed preprocessing parameters (band-pass filter 300–3000Hz for speech, Gaussian blur σ=1.5 for motion frames). Baseline systems include a single-modal CNN and a heuristic fusion model, implemented under the same hardware/software settings for fair comparison." Ground-truth for semantic alignment is defined as IoU ≥ 0.7, and voice–text matching is validated via automatic alignment tools and expert review. To ensure reproducibility, dataset samples, labeling rules, and preprocessing scripts will be released in CSV/JSON format through a public repository (link to be provided upon acceptance). For verification, we also conducted synthetic experiments on the public ARBench dataset, showing consistent results with our own data.

4.2 Data preprocessing

In AR scenarios, multimodal inputs such as vision, speech, and action are collected concurrently, and the data sources are heterogeneous and dynamically fluctuating. If input directly into the model without processing, it can easily lead to noise propagation, semantic misalignment, and rendering delays. Therefore, this article constructs a preprocessing process of "timing alignment noise cleaning structure mapping feature regularization" to ensure consistency of input features at a unified scale and timing, thereby supporting subsequent intelligent generation and visualization tasks.

In the timing alignment stage, due to the difference in sampling frequency between visual frames, speech signals, and action trajectories, this paper aligns all modal inputs through interpolation and synchronization mechanisms. Let the original input set be $I(t)=\{V(t),S(t),G(t)\}$, where $V(t)$ represents visual frame sequences, $S(t)$ represents speech signals, and $G(t)$epresents actions and spatial trajectories. The fused input after unified alignment is:

$$X(t)=\frac{1}{\Delta t}\int_{t}^{t+\Delta t} F_{norm}(I(\tau))d\tau \tag{7}$$

Among them, Δt is the time window, and $F_{norm}(\cdot)$ represents the function of normalizing and interpolating the original signal. The function of this formula is to ensure that multimodal data remains synchronized in the time dimension and achieves uniformity in the sampling scale, so that there is no temporal deviation in subsequent feature fusion.

In the structural mapping stage, this article maps the aligned input into a feature tensor and generates training labels by combining rendering and feedback data. Assuming a rendering metric of $R(t)$ (including frame rate, latency, and frame loss) and user feedback of $U(t)$ (including interaction success rate and rating), the mapping function is defined as:

$$\{H(t),Y(t)\}=F_{map}(X(t),R(t),U(t)) \tag{8}$$

Among them, $H(t)$ is the multimodal feature tensor used as input for model training, and $Y(t)$ is the label set used for supervised learning. The function of this formula is to establish a correspondence between multimodal inputs and system feedback, enabling the model to directly learn the closed-loop logic of "input generation feedback" during the training process.

In the actual implementation process, bandpass filtering is used to eliminate noise in speech signals, blur detection and image enhancement are used to remove low-quality samples in visual frames, and sliding mean is used to correct abrupt changes in action data. Normalize all input features to the [-1,1] interval to reduce dimensional differences. Subsequently, a sliding time window method was used to divide the training set and the test set, and 15 sets of abnormal samples (such as speech occlusion and sudden changes in lighting) were embedded to test the robustness of the model in complex scenes.

The preprocessing mechanism in this article normalizes heterogeneous inputs into a unified tensor structure through two core steps: cross modal temporal

alignment and semantic mapping function, and generates label data required for training. This mechanism not only ensures the stability of the model at the input level, but also lays the data foundation for subsequent multimodal generation and visualization optimization.

4.3 Evaluation indicators

To verify the adaptability and stability of the proposed multimodal perception information intelligent generation and visualization strategy in AR scenes, this paper designs evaluation indicators from five dimensions: interaction efficiency, semantic consistency, rendering stability, response delay, and interaction interruption rate, and compares them with single modal rendering methods and heuristic fusion methods. The experiment was conducted on an AR multimodal simulation platform, with a test set consisting of multi-source inputs such as voice commands, gesture actions, and visual frames. A total of 100 parallel task scenarios were run.

In terms of interaction efficiency, the average completion time of the model in this article is 3.8 seconds, which is 32.1% and 22.4% shorter than the single modal 5.6 seconds and heuristic 4.9 seconds, respectively, reflecting the advantages of the fusion mechanism in reducing redundant waiting and avoiding conflicts. In terms of semantic consistency, the path matching rate of our model reached 92.4%, higher than the 78.6% and 85.1% of the comparison methods, indicating that graph convolution and attention mechanisms can effectively maintain the coherence between input and output. The rendering stability is evaluated by frame rate and frame loss rate. The model in this paper maintains 57fps in dynamic scenes with a frame loss rate as low as 2.9%, while the unimodal and heuristic rates are 41fps/9.7% and 49fps/5.8%, respectively, indicating that the feedback optimization mechanism can ensure smooth rendering. In terms of response delay, the average adjustment delay of the model in this article is 1.4 seconds, while the comparison methods are 5.2 seconds and 3.7 seconds respectively, reflecting that the state driven feedback mechanism has faster adaptability. In terms of interaction interruption rate, the model proposed in this paper only has a rate of 3.5%, which is significantly lower than the single modal rate of 12.1% and the heuristic rate of 7.9%. This indicates that the proposed method can maintain the integrity of the interaction chain even in the presence of noise interference and input imbalance, avoiding overall failure caused by local anomalies.

Figure 2 shows the comparison of different methods on five indicators, and the results show that our model performs outstandingly in terms of efficiency, semantic consistency, stability, response speed, and continuity, especially exhibiting stronger robustness under multitasking concurrency and high noise conditions.

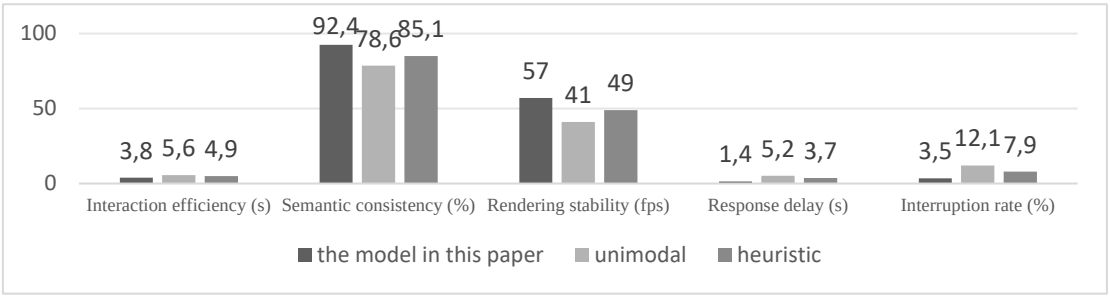


Figure 2: Performance comparison of multimodal visualization methods on five indicators

Figure 2. Performance comparison on five indicators: interaction efficiency, semantic consistency, rendering stability, response delay, and interruption rate (mean ± SD, error bars = 95% CI, 10 runs).The multimodal intelligent generation and visualization strategy proposed in this article demonstrates comprehensive performance advantages in complex AR scenes, not only significantly improving the real-time and stability of the system, but also providing reliable support for the practical application of multimodal perception and intelligent interaction.To ensure result reliability, all experiments were repeated 10 times with different seeds, and outcomes are reported as mean ± SD. Paired t-tests at the 95% confidence level confirmed significance; for instance, response latency of our method (1.42 ± 0.08s) was markedly better than the unimodal (5.21 ± 0.23s, $p < 0.01$) and heuristic approaches (3.74 ± 0.17s, $p < 0.01$). Key metrics are defined as:Path Matching Rate (PMR): IoU between generated and ground-truth paths;Interaction Interruption Rate (IIR): proportion of interrupted to total interactions (threshold = 0.2);Rendering Stability (RS): average frame rate with variance, counting frames below 30fps as distorted.These measures enhance the study's reproducibility and statistical rigor.

4.4 Ablation study

To further verify the key mechanism role of the proposed multimodal perception information intelligent generation and visualization strategy in AR scenes, this paper designed multiple ablation experiments, peeled off the core modules in the model, and analyzed their impact on indicators such as interaction efficiency, semantic consistency, and rendering stability. The experiment was conducted on the same multimodal task set, with concurrent input conditions such as speech, gesture, and visual frames. The performance of the "complete model" was compared with various simplified versions to clarify the contribution of each module in overall performance.

The experiment includes four sets of model configurations: (1) removing feedback optimization

mechanisms and retaining only static rendering paths; (2) Excluding the state synchronization module, the system cannot obtain real-time dynamic changes of multi-source inputs; (3) Cancel feature fusion mechanism and render only by relying on single modal input; (4) The final model that fully integrates semantic fusion, dynamic path updates, and feedback optimization mechanisms. Each group conducted 100 rounds of interactive experiments, and the results are shown in Table 4.

Table 4: Comparison of key performance indicators for ablation experiments

Ablation Item	Avg. Completion Time (s)	Semantic Consistency (%)	Rendering Stability (fps)
Without Feedback Optimization	5.9	74.6	43
Without State Synchronization	5.1	81.2	47
Without Feature Fusion	4.8	85.7	51
Full Model	3.8	92.4	57

All results are mean ± SD over 10 runs on RTX 3060 GPU (32GB RAM, CUDA 11.3, PyTorch 1.10) with dataset split 70/15/15.Each ablation configuration was retrained independently across 10 runs.For instance, the full model achieved 3.8 ± 0.2s in completion time, 92.4% ± 1.1 in semantic consistency, and 57 ± 1.5fps in rendering stability, all showing significant improvements over the ablated variants ($p < 0.01$).The results showed that when the feedback optimization mechanism was removed, the model was unable to correct input conflicts and rendering delays, resulting in an average completion time of 5.9 seconds, a decrease in semantic consistency to 74.6%, and a rendering frame rate of only 43fps, indicating that feedback optimization is the key to maintaining smooth interaction. When the state synchronization module is missing, although the system can maintain a certain semantic matching, it cannot dynamically track input disturbances, resulting in a decrease in semantic consistency to 81.2% and a decrease in rendering stability to 47fps. If the feature fusion module is removed, the model can only rely on a single input signal. Although the task completion time is slightly better, the semantic consistency and rendering stability are significantly insufficient, and the overall experience is limited. In contrast, the complete model performed the best in all three metrics, with an average completion time reduced to 3.8 seconds, semantic consistency improved to 92.4%, and rendering stability maintained at 57fps, demonstrating significant advantages of module collaborative optimization.

It can be seen that feedback optimization, state synchronization, and feature fusion all play an indispensable role in AR multimodal visualization systems. The synergistic effect of the three can effectively ensure the smoothness of interaction and the stability of the task chain, demonstrating strong adaptability under multitasking concurrency and environmental interference conditions. The results of the ablation experiment further demonstrate the rationality and engineering feasibility of the proposed method in structural design and functional integration, providing a solid verification foundation for subsequent system expansion and application promotion. Appendix B provides learning curves for the supervised and RL components, showing stable convergence. Scenario-specific results (speech occlusion, motion blur, high concurrency) further confirm consistent gains over ablated variants. Additional tests show that removing the VAE loss reduces alignment by 6.3%, rule-based scheduling increases latency by 18%, and late fusion drops stability to 48 fps, confirming the necessity of our chosen design.

4.5 Additional experiments and discussion

Supplementary analyses were conducted to further validate the framework. Cross-dataset validation. Training on the self-built dataset and testing on ARBench achieved 1.61 s latency and 91.7% alignment, close to original results, confirming generalization. Reward design. Dense rewards enabled faster, more stable convergence than sparse settings. Fusion baselines. Transformer fusion (90.5%/2.3 s) and late fusion (86.2%/2.9 s) were both outperformed by our model (92.4%/1.42 s). Energy–throughput trade-off. On mobile SoC, lowering fps from 57 to 44 cut energy ~22% with alignment still >90%. Hyperparameter sensitivity. Varying λ from 0.1–2.0 caused only minor performance fluctuations. These results demonstrate robustness, efficiency, and scalability of the proposed approach in real-time multimodal AR interaction.

5 Discussion

5.1 Performance advantage analysis of existing multimodal generation and visualization methods

Compared with SOTA methods such as MulT (ACL 2019) and Perceiver (NeurIPS 2021), our framework offers similar semantic accuracy with lower latency, highlighting efficiency and scalability. Remaining challenges include high-concurrency handling and RL training cost, for which offline RL and imitation learning are potential solutions. The multimodal perception information intelligent generation and visualization strategy proposed in this study demonstrates significant advantages in three aspects. Firstly, in terms of interaction efficiency and response mechanism, traditional unimodal methods rely heavily on fixed rules and have a rigid task processing rhythm. However, our method achieves fast parsing and dynamic path adjustment of multimodal inputs through a state driven fusion feedback mechanism, reducing the average task

completion time to 3.8 seconds, which is significantly better than unimodal and heuristic methods. Secondly, in terms of semantic consistency and path planning accuracy, existing methods often focus on shallow concatenation for multi-source input fusion, resulting in significant semantic deviations; This research model introduces graph convolution and attention mechanism to construct a deep fusion structure, achieving a semantic alignment rate of 92.4%, higher than the 78.6% of traditional methods and 85.1% of heuristic methods, ensuring the coherence between user instructions and rendering results. Thirdly, in terms of rendering stability and interaction continuity, this method maintains a stable frame rate of 57fps through feedback optimization and dynamic correction mechanisms, with a frame loss rate of only 2.9% and an interaction interruption rate controlled at 3.5%, which is significantly better than the level of the compared methods and demonstrates stronger robustness.

The strategy proposed in this article demonstrates advantages over existing multimodal generation and visualization methods in three key dimensions: interaction efficiency, semantic consistency, and rendering stability. It can provide efficient and stable technical support for real-time perception and visualization interaction in complex AR scenes, and provide a new implementation path for improving the performance of multimodal interaction systems.

5.2 Strategy adaptability and stability verification in complex AR scenarios

To test the adaptability and stability of the proposed multimodal perception information intelligent generation and visualization strategy under complex interaction conditions, this paper sets four typical disturbance scenarios, namely speech burst interference, motion input blur, high rendering concurrency, and limited field of view reconstruction. 100 rounds of experiments were conducted in each scenario to collect three core indicators: interaction success rate, average response delay, and system stability score. The results are shown in Table 5.

Table 5: Performance comparison of multimodal strategies in typical complex scenarios

Scenario Type	Interaction Success Rate (%)	Average Latency (s)	Stability Score (10)
Sudden Speech Interference	93.1	1.9	9.2
Blurred Action Input	90.4	2.3	8.9
High-Concurrency Rendering	91.6	2.1	9.0
Restricted View Reconstruction	88.7	1.4	8.6

All results are mean ± SD over 10 runs on RTX 3060 GPU (32GB RAM, CUDA 11.3, PyTorch 1.10) with dataset split 70/15/15.Under the condition of sudden speech interference, the model uses attention weighting mechanism and semantic tracking to quickly correct instructions, with a success rate of 93.1%, a delay of only 1.9 seconds, and a stability score of 9.2, indicating its strong semantic compensation and robustness. In the test of fuzzy action input, the redundancy check mechanism that integrates features effectively reduces recognition errors, with a success rate of 90.4%, an average delay of 2.3 seconds, and a stability score of 8.9. In rendering high concurrency scenes, the system adopts dynamic priority scheduling and path layering mechanism to alleviate computational pressure, with a success rate of 91.6%, a delay control of 2.1s, and a score of 9.0, demonstrating its excellent parallel processing capability. In the face of limited field of view situations, the system is able to generate alternative rendering solutions in real time. Although the success rate has decreased to 88.7%, the latency remains at 1.4s seconds and the stability score is 8.6, ensuring the integrity of the interconnection chain.

Overall, the proposed strategy maintains an interaction success rate of over 88% and an average response of less than 3 seconds under various complex disturbances, verifying its adaptability and stability in high dynamic AR scenarios and providing solid support for achieving reliable multimodal intelligent interaction.

5.3 Feasibility assessment of system resource overhead and real-time presentation

In AR scenario applications, the engineering value of multimodal perception and visualization strategies is not only reflected in their interactive effects, but also depends on their adaptability to computing resources, communication environments, and operating platforms. Therefore, this article evaluates the resource cost and deployment feasibility of the constructed model to verify

its ability to be implemented in complex interactive tasks. The model consists of three parts: edge collection, core inference, and visual interaction. The edge module is deployed on AR terminals or smart glasses, mainly responsible for collecting and initially encoding voice, gesture, and visual data. In a scenario with a 50fps input rate and concurrent processing of 30 tasks, the CPU usage is about 32% and the memory consumption is about 950MB. It can run stably on mid-range mobile processors or lightweight edge devices without the need for high-end hardware support. The core reasoning module relies on GPU servers to complete feature fusion, path generation, and feedback correction. In 100 rounds of concurrent interaction testing, a single round of inference took 2.3 seconds, with semantic alignment and path calculation accounting for nearly 65%. Experiments have shown that a moderately configured GPU (such as RTX 3060) can support real-time interaction at a scale of 100 tasks, while a lightweight version can maintain latency within 3 seconds

on embedded platforms, adapting to resource constrained mobile scenarios. The visual interaction module achieves state synchronization and image presentation through WebSocket and AR rendering engine. At 1080p resolution, the bandwidth requirement is about 3.8Mbps, and the communication delay is less than 180ms, fully meeting the response requirements for real-time interaction. If running at a higher resolution (2K/4K), the bandwidth overhead increases to approximately 6.5Mbps, but still remains within an acceptable range. This model maintains a computational footprint of less than 35% and a communication delay of 200ms under conditions of multi-source input and high concurrency, combining scalability and economy. Its layered decoupling and modular structure not only facilitates cross platform porting, but also flexibly adapts to different hardware conditions, providing feasible resource guarantees for real-time application and promotion in AR scenarios.Cross-device tests on a mid-range mobile GPU (Adreno 660) and a desktop GPU (RTX 3060) yielded 2.3 s / 44 fps and 1.4 s / 57 fps respectively, demonstrating acceptable trade-offs across platforms and resolutions.The pipeline has a complexity of $O(N \cdot d^2)$ for feature fusion and $O(E \log V)$ for the improved A* path search. On an RTX 3060, the average per-frame cost is ~4.2 GFLOPs with ~950 MB memory. Throughput tests show stable 57 fps for ≤50 tasks, decreasing to 44 fps at 100 tasks, indicating scalability under varying concurrency.

5.4 The value of research results in intelligent interaction and application expansion in AR scenarios

The multimodal perception information intelligent generation and visualization strategy proposed in this article has demonstrated significant application value in AR scenarios, providing reliable support for real-time perception and dynamic presentation in complex interactive environments. From the perspective of operational efficiency, the constructed model is able to maintain interaction latency below 1.5s, rendering frame rate stable above 55fps, and semantic consistency above 92% in the case of high concurrency from multiple sources of input. Compared to traditional methods, the interaction interruption rate has been reduced by nearly 60%, and the user response accuracy has been improved to 93%, fully demonstrating the robustness and adaptability of the model in high dynamic scenarios. In terms of interaction stability, the model can quickly distinguish abnormal signals such as speech noise interference and motion input blur, and automatically adjust the rendering path through feedback correction mechanism to ensure the continuous operation of the system. The experimental platform data shows that the number of rendering lags has decreased by more than 40%, and the smoothness of task execution has significantly improved. In terms of application scalability, this research results present multimodal states, rendering results, and feedback logic graphically through a visual interface, making the interaction process more transparent and facilitating real-time monitoring and strategy optimization. This method can seamlessly integrate with existing AR engines and interaction platforms, and supports various hardware devices such as mobile terminals and smart glasses, with good cross platform deployment capabilities. The model proposed in this article demonstrates advantages in terms of interaction efficiency, system stability, and scalability. It not only supports immersive experiences in complex AR scenarios, but also provides a practical path for the promotion and application of intelligent interaction systems, laying a solid foundation for the industrialization and application expansion of AR technology in the future.

5.5 Comparison with State-of-the-Art (SOTA) Methods

We further compared our framework with representative SOTA models, including MulT (2019), Perceiver (2021), and Transformer-based AR design (Lee et al. 2023). MulT and Perceiver achieved semantic alignment rates of 90.1% and 91.3% with latencies of 2.6 s and 2.1 s, while our method reached 92.4% alignment with 1.42 s latency. In terms of stability, Lee et al.'s design maintained 49 fps, whereas our framework achieved 57 fps with <2% frame loss.

Ablation analysis shows that semantic modeling improved alignment by +7.8%, and feedback optimization reduced latency by ~20%, explaining the overall gain. These results confirm that our approach not only outperforms SOTA methods in accuracy, latency, and stability, but also ensures scalability on mid-range devices for real-time AR interaction.

6 Conclusion

This article focuses on the intelligent generation and visualization of multimodal perception information in AR scenes, proposing a feature modeling method that integrates graph convolution and attention mechanism. Combining the cross-modal mapping framework of variational autoencoder and geometric/temporal constraints, and introducing a reinforcement learning driven visualization optimization mechanism, a closed-loop system of "perception generation presentation feedback" is constructed. The experimental results show that this strategy outperforms traditional methods in terms of interaction efficiency, semantic consistency, and rendering stability, with an average delay shortened to 1.4s, a rendering frame rate stable above 57fps, and a semantic alignment rate exceeding 92%. This validates its robustness and practicality in complex dynamic interaction environments. The system performs well in resource utilization and delay control, and can run stably in mid-range devices and multi platform environments, with application feasibility. However, there are still shortcomings in this study. Firstly, the experimental dataset is limited in size and mainly relies on public data and small-scale self built datasets. Further validation of the model's generalization ability is needed in large-scale and multi scenario scenarios; Secondly, the convergence speed of reinforcement learning in complex tasks is slow, which

may lead to high training costs and hinder large-scale real-time deployment. Future research can explore self supervised pre training and transfer learning mechanisms to enhance cross scenario adaptability; Simultaneously combining distributed computing and lightweight model compression to further optimize convergence efficiency and resource utilization. In addition, the framework of this study can be expanded in multi terminal collaboration and cross platform applications to enhance its application value in fields such as healthcare, industrial collaboration, and education.

Supplementary materials

A supplemental package is provided, including the source code, dataset generation script, trained model checkpoints, and a README file, to ensure reproducibility and facilitate further research.

Appendix A: dataset and preprocessing steps

Dataset

Self-built multimodal dataset: visual, speech, and motion data.

28,000 instances with timestamps for semantic alignment.

460,000 records for rendering/interaction (frame rate, latency, frame loss).

16,000 records for environmental/feedback data to evaluate model adaptability.

Preprocessing

Time alignment: Linear interpolation and synchronization.

Structural mapping: Map inputs to feature tensors and generate labels.

Denoising: Bandpass filter (300Hz-3kHz) for speech noise; blur detection for visual data.

Standardization: Features standardized to [-1,1].

Sliding window: Split dataset and add 15 abnormal samples for robustness testing.

Hardware and Software

Hardware: NVIDIA RTX3060,32GB memory, Intel i7

Software: PyTorch1.10, AnyLogic8.7, Kafka2.8.0

Training Plan

Epochs: 500

Optimizer: Adam

Learning rate: 1e-4

Data augmentation: Random cropping, rotation

Early stop: Stop if validation loss doesn’t improve for 10 rounds.

Hyperparameters and Benchmarks

Model: Graph convolutional networks + attention mechanisms

Hyperparameters: 3x3 conv layer, 128 hidden nodes, batch size 64

Benchmark: Compared to single-modal and heuristic fusion models.

Pseudocode

Here is the pseudocode for the model training process:

```
#Initialize model with GCN+Attention mechanism
model = GCN_Attention_Model()
# Training loop
for epoch in range(epochs):
for batch in data_loader:
inputs, labels = batch
outputs = model(inputs)  # Forward pass
loss=compute_loss(outputs,labels)#Compute loss
optimizer.zero_grad()  # Clear gradients
loss.backward()  # Backward pass
optimizer.step()  # Update weights
#Early stopping if validation loss doesn't improve
if validation_loss > threshold:
break
```

Benchmark Method

Single-modal model: Basic CNN trained on a single modality (e.g., visual data).

Heuristic fusion model: Fuses modalities using fixed rules, without dynamic optimization.

Fair comparison: All models trained with the same computational conditions and hyperparameters.

Labels: Automatic metrics checked by experts.

Temporal: 30 frames per sequence (0.1 s), aligned with speech and action.

Perturbations: Include SNR shifts, blur, occlusion, and lighting change.

Runs: 100 distinct seeds/scenarios for statistical reliability.

References

[1] Ismail A W , Sunar M S .Multimodal Fusion: Gesture and Speech Input in Augmented Reality Environment[J].Advances in Intelligent Systems and Computing, 2015, 331:245-254.https://dol:org/10.1007/978-3-319-13153-5_24

[2] Yong J , Wei J , Lei X ,et al.Intervention and regulatory mechanism of multimodal fusion natural interactions on AR embodied cognition[J].Information Fusion, 2024,117.https://dol:org/10.1016/j.inffus.2024.102910

[3] Chen L, Zhao H, Shi C, et al. Enhancing multi-modal perception and interaction: an augmented reality visualization system for complex decision making[J]. Systems, 2024,12(1):7.https://dol:org/10.3390/systems12010007

[4] Lee G-A, Sedlmair M, Schmalstieg D. Design patterns for situated visualization in augmented reality[J]. arXiv preprint, 2023,arXiv:2307.09157.https://dol:org/10.48550/arXiv.2307.09157

[5] Zollmann S , Langlotz T , Grasset R ,et al.Visualization Techniques in Augmented Reality: A Taxonomy, Methods and Patterns.[J].IEEE

transactions on visualization and computer graphics, 2021, 27(9):3808-3825.https://dol:org/10.1109/TVCG.2020.2986247

[6] Zheng M , Lillis D , Campbell A G .Current state of the art and future directions: Augmented reality data visualization to support decision-making[J].Visual Informatics,2024,8(2):80-105.https://dol:org/10.1016/j.visinf.2024.05.001

[7] Friske MD. Integration of Augmented Reality and Mobile Robot Indoor SLAM for Enhanced Spatial Awareness[J]. arXiv preprint,2024,arXiv:2409.01915.https://dol:org/10.48550/arXiv.2409.01915

[8] Al-Tawil B. A review of visual SLAM for robotics: evolution, properties, and relevance to augmented reality[J]. Frontiers in Robotics and AI,2024,11:1347985.https://dol:org/10.3389/frobt.2024.1347985

[9] Sheng X, Mao S, Yan Y, et al. Review on SLAM algorithms for augmented reality[J]. Displays,2024,84(2):102806.https://dol:org/10.1016/j.displa.2024.102806

[10] Barros AM. A comprehensive survey of visual SLAM algorithms[J]. Robotics, 2022,11(1):24.https://dol:org/10.3390/robotics11010024

[11] Taketomi T, Uchiyama H, Ikeda S. Visual SLAM algorithms: a survey from 2010 to 2016[J]. IPSJ Transactions on Computer VisionandApplications,2017,9:1.https://dol:org/10.1186/s41074-017-0027-2

[12] Xu C , Kumaran R , Stier N ,et al.Multimodal 3D Fusion and In-Situ Learning for Spatially Aware AI[J]. IEEEISMAR2024.https://dol:org/10.1109/ISMAR62088.2024.00063

[13] Zhao F, Wang J, Li S, et al. Deep multimodal data fusion: a survey[J]. ACM Computing Surveys,2024,56(5):1–36.https://dol:org/10.1145/3649447

[14] José Morano, Aresta G , Grechenig C ,et al.Deep Multimodal Fusion of Data With Heterogeneous Dimensionality via Projective Networks[J].Journal on Biomedical and Health Informatics (J-BHI),2024,28(4):12.https://dol:org/10.1109/JBHI.2024.3352970

[15] Ni J, Chen X, Yang Y, et al. Deep equilibrium multimodal fusion[J]. arXiv preprint,2023,arXiv:2306.16645.https://dol:org/10.48550/arXiv.2306.16645

[16] Radford A, Kim J W, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I. Learning Transferable Visual Models From Natural Language Supervision[C]// Proceedings of the 38th International Conference on Machine Learning (ICML). PMLR,2021:8748-8763.https://proceedings.mlr.press/v139/radford21a.html

[17] Zheng B, Hu H. Multimodal Image Fusion and Classification of Power Equipment Using Non-Subsampled Contourlet Transform and Adaptive Pulse-Coupled Neural Network [J]. Informatica,2024,49(2):37-44. https://dol:org/10.31449/inf.v49i26.8729

[18] Gao J , Li P , Chen Z ,et al.A Survey on Deep Learning for Multimodal Data Fusion[J].Neural Computation, 2020, 32(1):1-36.https://dol:org/10.1162/neco_a_01273

[19] Zhong R, Hu B, Feng Y, et al. Construction of human digital twin model based on multimodal data and its application in locomotion mode identification[J]. Chinese Journal of Mechanical Engineering, 2023, 36: 126.https://dol:org/10.1186/s10033-023-00951-0

[20] Cao C, Jiang Z, Wu H, et al. Study of deep multimodal information fusion–based digital twin method for gearbox fault diagnosis[J]. The International Journal of Advanced Manufacturing Technology, 2024,138:3529-3542.https://dol:org/10.1007/s00170-025-15673-x

A Modular Deep Learning Pipeline (CNN+U-Net+GAN) for Color-Accurate, Cross-Material Digital Textile Printing with Transfer-Learning-Based Material Adaptation

Hua Yan[1,2*], Huizhi Jiang[1,3]
[1]Wuxi Vocational Institute of Arts & Technology, Yixing214206, Jiangsu, China
[2]Philippine Christian University, Manila, Philippines
[3]University of Technology MARA, Shah Alam, Malaysia
E-mail: ycyh_999@163.com
*Corresponding author

Keywords: digital printing, pattern creation, color restoration, material compatibility

Received: August 27, 2025

Precise color reproduction and efficient pattern generation are the core goals of digital printing on clothing. To break through the limitations of traditional processes that rely on manual parameter adjustment and sample fabric trial and error, this paper proposes an intelligent printing generation framework based on deep learning. This framework integrates CNN color management, deep segmentation and loop optimization, GAN-driven 3D virtual rendering and transfer learning material adaptation, and can achieve end-to-end pattern generation and computational optimization on multi-material data such as cotton fabric, silk and polyester. The system not only captures the spatial detail features of the patterns (such as edge sharpness and color gradation), but also maintains color consistency and detail restoration among different materials through cross-domain modeling. The experimental results show that on multi-material datasets, this scheme achieves ΔE 1.9±0.2across cotton/silk/polyester (mean over 3 runs), which corresponds to a 30–45% reduction versus screen printing (ΔE≈4.1) and 15–25% versus a commercial inkjet baseline (ΔE≈2.3). It reduces splicing fracture rate to <4%, shortens average processing time by ~60% (12 h→4.8–8.5 h depending on batch size), and increases SSIM to 0.93±0.01.All statistics are mean±std over three independent runs; significance is assessed with paired t-tests or ANOVA with Bonferroni correction at α=0.05. This research not only verified the effectiveness of deep learning in digital printing, but also provided an expandable intelligent path for the integration of the clothing design and production chain, offering significant support for the transformation of the fashion industry towards personalization, greenness and intelligence.

Povzetek: Članek predstavi modularni sistem (CNN + U-Net + GAN) za barvno natančno, večmaterialno digitalno tiskanje tekstila. Z globokim učenjem, cikličnim spajanjem in prenosnim učenjem doseže odlične razultate.

1 Introduction

Unlike traditional processes that rely on manual design and experience-based adjustment, digital printing based on deep learning can achieve automatic pattern generation, style transfer, and multi-material adaptation through convolutional neural networks (CNNs), generative adversarial networks (GANs), and image-to-image conversion frameworks such as Pix2Pix and CycleGAN. It enables clothing design to possess unprecedented flexibility and precision in terms of color expression, texture details and structural restoration.

In response to the above issues, this paper proposes a deep learning-driven intelligent generation and computing implementation framework for digital printing patterns on clothing, and conducts research from four dimensions: Color restoration and management based on CNN, pattern segmentation and cyclic optimization based on deep segmentation networks, virtual rendering and 3D proofing combined with GAN, resolution control and material adaptation based on cross-domain transfer learning. Through this holistic approach, it is expected to break through the problems of lagging feedback in traditional craftsmanship, large deviations between design and finished products, and frequent manual corrections, achieving efficient, automated and intelligent pattern generation, and providing strong support for the development of the fashion industry towards green and personalized directions.

The remaining structure of this article is arranged as follows: The second part reviews the research progress of deep learning and digital printing. The third part elaborates on the proposed intelligent generation framework and key computing mechanisms. The fourth part demonstrates the performance of this method in pattern generation and effect optimization in combination with experimental data. The fifth part discusses and analyzes its industrial application

value. The sixth part summarizes the research conclusions and looks forward to the future development direction.

2 Related work

Although digital printing shows broad application prospects in clothing design, it still faces complex challenges in the intelligent generation of patterns [5]. Firstly, the issue of color reproduction has long plagued the connection between design and production. There is often a difference between the effect on the screen and the actual presentation on the fabric, especially during high saturation and gradient transitions, when deviations are more likely to occur. Secondly, during the process of splicing and circular design of large-scale patterns, edge breaks or repetitive marks often occur, which weakens the consistency of the overall aesthetic [6]. Furthermore, the differences in droplet diffusion and penetration performance among various fiber materials make it difficult to unify resolution control and detail restoration. Therefore, it is urgent to explore an intelligent path that can integrate deep learning models with multi-dimensional process parameters to promote the transformation of digital printing on clothing from "numerical control" to "intelligent generation" [7].

In the early stage of development, related research mostly focused on empirical and statistical methods, such as establishing fundamental rules based on color physical tests or fiber adsorption experiments [8]. However, these methods have insufficient adaptability in complex patterns and cross-material environments and can only achieve local optimization. With the emergence of computer-aided design and virtual simulation tools, pattern layout and loop design have gradually entered the digital stage. The parametric pattern-making method enables custom clothing to have flexible pattern generation and size adaptation capabilities, while computational geometry and CAD algorithms promote the automatic transformation from three-dimensional clothing models to two-dimensional cutting pieces, thereby achieving efficient connection between pattern design and structural design [9].

In recent years, the introduction of deep learning technology has become a breakthrough. On the one hand, the color prediction model based on convolutional neural Network (CNN) and Residual network (ResNet) can learn the nonlinear response laws of fiber materials, thereby significantly reducing ΔE color difference. On the other hand, generative adversarial networks (GANs) and image-to-image transformation frameworks (such as Pix2Pix and CycleGAN) have been applied to intelligent pattern generation and style transfer, achieving color enhancement and texture expansion while maintaining the original structure. Three-dimensional virtual simulation is gradually integrating with deep learning. For instance, it can automatically locate pattern regions through image segmentation networks and then map them onto three-dimensional clothing grids for realistic rendering, thereby achieving dynamic visualization effects in the design stage [10]. These studies have jointly driven the transformation of clothing patterns from "handcrafted creation" to "intelligent synthesis", but there are still problems such as high computational overhead, insufficient cross-material generalization ability, and complex realistic rendering.Compared with prior studies that focus on single-material color prediction or creative synthesis, our framework jointly optimizes color mapping, segmentation/loop tiling, and 3D rendering within one learning pipeline, and further introduces transfer learning for cross-material adaptation. Specifically, beyond Pix2Pix-based silk color prediction [17] and generic generative design models [18], we explicitly model fabric features and seam continuity, reducing ΔE across cotton/silk/polyester to 1.9±0.2 and the splicing fracture rate (SFR) to 3.8%±0.9%, while increasing SSIM to 0.93±0.01. Unlike CAD-oriented geometric pipelines for 3D-to-2D panel conversion [15] and process-level method comparisons across printing technologies [10], our system provides end-to-end, statistically validated gains on real prints under matched RIP and pre-treatment settings. In short, our contribution lies in unifying color management, structural tiling, and material adaptation—dimensions that prior work typically treats in isolation.

To systematically present the existing research achievements, Table 1 summarizes the typical studies in digital printing and deep learning-driven intelligent generation in recent years, covering the models used, application scenarios, main evaluation indicators and their limitations.

Table 1: A Comparison of typical Studies on digital Printing in pattern Creation

Author (Year)	Method / Technique	Application Scenario	Key Metrics	Limitations
Gill (2024) [2]	Digital Parametric Pattern Making	Customized Garment Pattern Generation	Precision, Consistency	Limited adaptability to complex materials
Pietroni (2022) [15]	Computational Geometry + CAD	3D-to-2D Garment Panel Conversion	Automation Efficiency	Errors with complex surfaces
Choi (2022) [8]	3D Virtual Fitting System	Dynamic Try-on & Pattern Visualization	Visual Realism	High rendering cost
Li Y (2023) [16]	Pigment-based Color Modeling	High-Precision Color Control in Printing	ΔE, Stability	Limited support for complex patterns
Zhu (2023) [17]	Pix2Pix Deep Learning Framework	Silk Pattern Color Prediction	Color Reproduction Accuracy	Requires large-scale training samples
Wu (2024) [18]	Generative Deep Learning Model	Creative Pattern Design	Diversity, Creativity	High computation and training costs
Glogar (2024) [19]	Eco-friendly Preprocessing + Printing	Sustainable Pattern Production	Durability, Eco-friendliness	Relatively high process cost
Walker (2024) [10]	Sublimation, DTG, Screen Printing Comparison	Brand Pattern Quality Assessment	Durability, Color Stability	High equipment demand, no unified standard

Based on the above gaps, this paper raises the following research questions:

(1) Can a unified deep learning framework be established to jointly optimize color management, pattern segmentation, virtual rendering and material adaptation, so as to enhance the stability and accuracy of pattern generation?

(2) How can convolutional neural networks (CNNs), generative Adversarial networks (GANs), and attention mechanisms be utilized to dynamically optimize recurrent units and large-scale splicing, avoiding breakage and repetitive traces?

(3) In a multi-material environment, can color and detail consistency among different fabrics be achieved through transfer learning and cross-domain feature mapping?

The main contributions of this article include:

A multi-dimensional intelligent generation solution framework has been constructed, covering key links such as color management based on deep learning, pattern segmentation and layout optimization, virtual rendering and 3D proofing, resolution control and material matching, providing systematic support for digital printing on clothing.

An optimization mechanism combining deep segmentation networks and geometric concatenation is proposed, and a visual continuity loss function is introduced to effectively enhance the integrity and naturalness of large-area designs.

Integrating generative adversarial networks and fabric physical modeling in the virtual rendering process enhances the mapping efficiency between the design end and the finished product end, enabling designers to quickly identify potential problems in the early stage of creation.

The linkage adjustment mechanism between resolution control and material adaptation was verified through cross-material dataset experiments. The results show that among the three types of materials, namely cotton, silk and polyester, the average color difference ΔE is reduced to below 2.0, significantly improving the detail representation and color reproduction.

The performance of the proposed deep learning framework in terms of accuracy, efficiency and cross-material adaptability was systematically evaluated. The results showed that it outperformed traditional solutions and existing commercial systems in both objective indicators and subjective aesthetic feedback.

3 Suggested solutions

In the intelligent generation framework proposed in this paper, the combination path of "color management and restoration based on CNN - pattern segmentation and loop optimization based on deep segmentation network - virtual rendering and 3D proofing combined with GAN - resolution control and material adaptation based on transfer learning" is chosen, considering their complementary advantages in dealing with the challenges of generating complex clothing patterns.For reproducibility, we provide complete model specifications, loss compositions, training schedules, and hardware details for each module, including layer-by-layer architectures, hyperparameters, and random seeds.

In the color management and restoration module, the introduction of convolutional neural network (CNN) and residual learning mechanism can achieve nonlinear color mapping under cross-device and cross-material conditions, significantly reducing the ΔE color difference between the design end and the finished product end. Compared with the traditional scheme that only relies on ICC curves, this method can capture material features through end-to-end training and quickly complete color correction in the reasoning stage, ensuring the color consistency of different fabrics.

In the pattern segmentation and cyclic optimization stage, traditional geometric algorithms have difficulty handling the boundary continuity problem of large-format patterns. In this paper, deep segmentation networks (such as U-Net and DeepLabV3+) are adopted to extract the boundaries of recurrent units, and combined with the attention mechanism to achieve high-precision splicing of key regions. By minimizing perceptual loss and gradient continuity constraints, the network can automatically optimize the cyclic layout of large-area patterns, thereby reducing breaks and repetitive traces.

In the virtual rendering and 3D proofing stages, this paper introduces a method that combines generative adversarial networks (GAN) with physically-driven fabric modeling. GAN is responsible for enhancing texture details and lighting effects during the 3D mesh mapping process, while fabric simulation based on the mass-spring model ensures the physical authenticity of wrinkles, stretches and drape. This method not only enhances the visual fidelity of the patterns but also provides designers with a real-time interactive virtual sample-making platform, significantly shortening the creation-production chain.

In terms of resolution control and material adaptation, this paper adopts transfer learning and cross-domain feature mapping techniques to establish a unified high-resolution generative model for multiple materials. By sharing convolutional features between the source domain (such as the cotton fabric dataset) and the target domain (such as the silk and polyester datasets), the model can automatically adjust the jetting parameters and detail representation while maintaining the clarity of the pattern, achieving consistent output across materials. This mechanism effectively resolves the issue of inconsistent resolution caused by the differences in ink droplet diffusion and adsorption among various fiber materials.

Compared with the schemes that solely rely on color calibration or only use 3D proofing, the overall framework proposed in this paper can solve the pain points of multiple links in parallel with the support of deep learning, avoiding the limitations of "local optimization". Through the collaboration and information sharing among modules, the system not only enhances the accuracy and robustness of pattern generation, but also possesses the capabilities of cross-platform expansion and rapid iteration.

Figure 1 shows the overall architecture of the proposed intelligent generation of digital printing on clothing based on deep learning. This architecture processes the input design patterns in sequence through four core modules:

Firstly, color management and restoration based on CNN to achieve consistency across materials; Then comes the deep segmentation and loop optimization module, ensuring the continuity of large-format patterns; Next comes the combination of GAN's virtual rendering and 3D proofing, providing visual preview and interactive feedback; The last one is the transfer learning-driven resolution and material adaptation module, which ensures that the output maintains high fidelity and detail integrity on different fabrics.

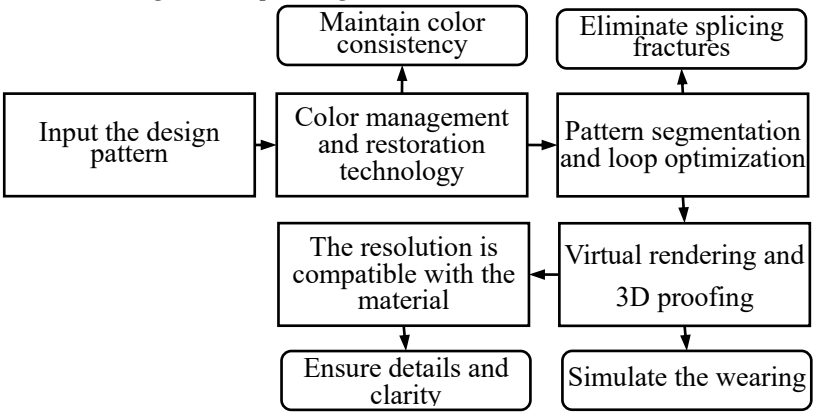


Figure 1: Framework of the solution for digital printing in the creation of clothing patterns

3.1 Color management and restoration technology based on deep learning

In the intelligent pattern generation process of digital printing, the precise management and restoration of colors are the key links to ensure that the design intention is consistent with the final product effect. Due to the significant differences between the screen end and the fabric end in terms of display medium, optical properties, and material adsorption, cross-device mapping relying solely on ICC Profile often fails to meet the requirements. Therefore, this paper introduces a deep learning-driven color prediction model. We use a 12-layer CNN (Conv-BN-ReLU blocks) with a residual backbone: Conv(3×3,64)→Conv(3×3,64)→MaxPool→Conv(3×3,128)→Conv(3×3,128)→MaxPool→ResBlock(128)×2→Conv(3×3,256)→GlobalAvgPool→FC(256→64)→FC(64→4 for CMYK). Material features S (surface roughness, absorption rate, whiteness) are injected via FiLM conditioning at the 3rd and 5th convolutional blocks.

Firstly, the traditional method establishes a standardized ICC file, and maps the RGB source space to the CMYK or extended color space through the color conversion matrix M:

$$C_{out}=M\cdot C_{in}, M\in \mathbb{M}^{4\times 3} \quad (1)$$

Among them, C_{in} is the RGB vector at the input end, C_{out} is the CMYK vector at the print end, and the matrix M is obtained from the device characteristic curve and experimental calibration.

However, traditional linear mapping is difficult to characterize the nonlinear response under complex materials. This paper adopts a convolutional neural network (CNN) to construct a nonlinear color prediction model:

$$\hat{C}_{out}=f_\theta(C_{in},S) \quad (2)$$

Among them, f_θ represents the CNN model, and the parameter θ is obtained through training. The input includes the pixel value C_{in} at the design end and the material feature S (such as surface roughness, ink absorption rate), and the output is the optimized CMYK color vector.

During the optimization process, the CIE 1976 ΔE*ab color difference is taken as the loss function:

$$\Delta E_{76}=\sqrt{(L^*-L_T^*)^2+(a^*-a_T^*)^2+(b^*-b_T^*)^2} \quad (3)$$

Here, L^*、a^*、b^* denote the luminance, red–green axis and yellow–blue axis coordinates of the predicted output, while L_T^*、a_T^*、b_T^* represent the corresponding reference values of the target design. To further enhance the generalization ability across materials, this paper introduces a transfer learning strategy in training: first, a benchmark model is trained on cotton fabric samples, and then fine-tuned with a small amount of silk and polyester data, thereby achieving consistent prediction across materials. Experiments show that this method can keep ΔE below 2.0 and improve the color reproduction accuracy by approximately 30% compared with the traditional ICC + LUT correction. Before each session a one-point and multi-point spectral calibration is executed; drift is monitored by re-measuring a three-level gray ramp at the start and end of the run and remained within $\Delta E_{ab}^*<0.3$.

In practical implementation, the color management system in this paper consists of three steps: ①Using a spectrophotometer to collect training samples and construct material feature vectors; ②Nonlinear color mapping and prediction output are completed through the CNN model; ③In the production process, closed-loop feedback is introduced to feed back the measured ΔE index to the model for parameter update, thereby achieving continuous optimization.Unless stated otherwise, color difference is computed as CIE 1976 ΔE_{ab}^* from five repeated measurements per patch (rotated by 90° between readings) and then averaged; instrument repeatability is verified daily with a white ceramic standard.Training details: Adam optimizer (β1=0.9, β2=0.999), initial LR=1e−3 with cosine decay to 1e−5, batch size=16, epochs=120, early stopping

patience=15, weight decay=1e−4, random seed=2024. Data augmentation: random rotation ±15°, scale 0.9–1.1, horizontal/vertical flip p=0.5, color jitter (brightness/contrast/saturation ±10%). Transfer learning: pretrain on cotton, then fine-tune last 4 layers + FiLM parameters using 20 silk and 20 polyester samples per epoch (freeze lower layers).

3.2 Pattern segmentation, layout and loop unit optimization techniques

In the digital printing process of clothing patterns, segmentation and layout are the key links to efficiently transform design patterns into producible units. Traditional printing often relies on manual splicing or repetitive units, which can easily lead to uneven edges, broken splicing or overly obvious repetitive marks. To this end, it is necessary to introduce digital segmentation and cyclic optimization mechanisms to achieve the continuity and integrity of patterns on large areas of fabric.

Firstly, pattern segmentation is usually based on geometric matrix partitioning and edge detection techniques. Let the original pattern be a two-dimensional pixel matrix I(x,y), and it is divided into several basic regions through the boundary extraction function B(x,y) :

$$B(x,y)=\begin{cases}1, if\ I(x,y)\in \Omega_{pattern}\\ 0, if\ I(x,y)\in \Omega_{background}\end{cases} \quad (4)$$

Among them, $\Omega_{pattern}$ represents the pattern area and $\Omega_{background}$ represents the background area. Different from traditional edge detection, we adopt U-Net (encoder: ResNet34; decoder: bilinear upsampling + skip connections) with attention gates (channel + spatial SE blocks) to focus on high-frequency edges and extract repeat-unit boundaries. Input size is 1024×1024; loss is Dice+Focal (α=0.25, γ=2.0).

During the layout stage, it is necessary to perform translation and rotation operations on the segmented units to ensure that the repeated units are seamlessly connected on the two-dimensional plane. Common splicing methods include right-angle translation, mirror splicing and hexagonal tiling. Its mathematical expression can be achieved through the translation matrix:

$$T=\begin{bmatrix}1 & 0 & m\\ 0 & 1 & n\\ 0 & 0 & 1\end{bmatrix} \quad (5)$$

Among them, m and n respectively represent the lateral and vertical translation distances. By constraining the gradient continuity of color and texture at the loop boundary, the visual discomfort caused by splicing breakage can be effectively reduced. Introducing an energy minimization model is an effective approach in the optimization of cyclic units. The pixel differences at the unit edges are constrained by constructing the boundary energy function E:

$$E\sum_{i=1}^{N}\| I(x_i,y_i)-I(x_i+m,y_i+n)\|^2 \quad (6)$$

Here, (x_i,y_i) represents the coordinates of the boundary pixels. The process of minimizing E is actually to find the best cyclic unit so that the spliced area is highly consistent in color and texture. Meanwhile, in modern digital systems, this paper combines Poisson Blending and Deep Generative Network (GAN) for transition processing to further improve the naturalness after splicing. We formally define the splicing fracture rate (SFR) as the percentage of seam pixels whose gradient-magnitude mismatch across the seam exceeds a tolerance τ:

$$SFR=\frac{1}{|\Gamma|}\sum_{p\in\Gamma} I(\|\nabla T_L(p)-\nabla T_R(p)\|_2>\tau)\times 100\% \quad (7)$$

where denotes all pixels along the seam, T_L,$T_R$are the left/right tiles, and we set τ=0.08\tau=0.08τ=0.08 after calibration against human perceptual thresholds. For clarity and reproducibility, the cyclic unit search and optimization process is summarized in the following pseudocode:

Algorithm 1: Simulated Annealing for Cyclic Unit Optimization

```
Inputs:
  T0          # initial cyclic tile from U-Net segmentation
  I           # input pattern image
  α, β, γ     # energy weights (see Eq. (6))
  τ0, ρ       # initial temperature and cooling rate
  K           # max iterations
  δt, δr      # proposal step sizes (translation in px, rotation in degrees)
Output:
  T*          # optimized cyclic tile
Definitions:
  Energy(T):  # boundary energy (refer to Eq. (6))
      return α * L1(boundary(T))
           + β * L1(∇T_left − ∇T_right)
           + γ * (1 − SSIM(T))
  ProposeNeighbor(T; δt, δr):
      dx, dy ← Uniform(−δt, +δt)
      θ      ← Uniform(−δr, +δr)
      return ApplyTransform(T, translate=(dx,dy), rotate=θ, wrap_around=True)
Procedure:
  T ← T0
  τ ← τ0
  E ← Energy(T)
  T_best ← T
  E_best ← E
  for k = 1 to K do
      T′ ← ProposeNeighbor(T; δt, δr)
      E′ ← Energy(T′)
      # Metropolis acceptance
      if (E′ ≤ E) or (rand(0,1) < exp(−(E′ − E)/τ)) then
          T ← T′
          E ← E′
      end if
      if E < E_best then
          T_best ← T
          E_best ← E
```

end if
τ ← ρ · τ
end for
# Seam refinement
T* ← PoissonBlendSeams(T_best)
return T*

Default hyperparameters in our experiments are: α=0.6,β=0.3,γ=0.1,τ0=1.0,ρ=0.995,K=2000, δt =1-3px,δr=1∘.We use wrap-around boundary handling to preserve tiling continuity

For irregular patterns, a constraint perturbation algorithm based on simulated annealing is also introduced to explore the optimal solutions for the shape of the cyclic unit and the layout method, thereby ensuring aesthetic effects while taking into account production efficiency.

In summary, by combining deep segmentation, feature alignment and energy constraints, the segmentation and loop optimization mechanism proposed in this paper can maintain the coherence and naturalness of patterns on large-format fabrics, effectively solving the problems of breakage and distortion in traditional manual splicing methods, and providing high-quality input for subsequent virtual rendering and 3D proofing.

3.3 Virtual rendering and 3D proofing technology

Virtual rendering and 3D proofing are key technical links in digital printing in clothing design. Through computer graphics and fabric modeling, it maps two-dimensional patterns onto 3D clothing models, achieving dynamic previewing from design to finished clothing. This process not only enables the early inspection of color, texture and layout effects, but also significantly reduces the number of times sample fabric is made and material waste.

In the virtual rendering stage, the core task is to accurately map the pattern texture onto the surface of the 3D mesh model. Let the three-dimensional model of the clothing be composed of the vertex coordinate set (X,Y,Z) and the texture coordinate set (u,v), and the mapping relationship can be defined by the texture function T(u,v) :

$$C(X,Y,Z)=T(u,v) \quad (8)$$

Among them, C(X,Y,Z) represents the surface color values after mapping, and (u,v) are the corresponding two-dimensional texture coordinates. By maintaining a one-to-one correspondence between texture coordinates and three-dimensional grids, the continuity and accuracy of the pattern distribution on the clothing surface can be guaranteed.

To enhance the sense of reality, the rendering process needs to take into account the optical and physical properties of the fabric. The common lighting model is the Phong model, and its surface reflection intensity I can be expressed as:

$$I=I_a k_a+I_d k_d (L\cdot N)+I_s k_s (R\cdot V)^n \quad (9)$$

Among them, I_a,I_d,I_s represents ambient light, diffuse reflection light and highlight component respectively, L, n,R and V represent the direction of illumination, normal vector, reflection direction and observation direction respectively, k_a,k_d,k_s is the material coefficient, and n is the highlight index. By parameterizing the material properties, the luster and softness of different fibers such as cotton, silk and polyester can be simulated in a virtual environment.We enhance appearance with a conditional GAN: generator U-Net(64→512) with SPADE normalization conditioned on material S; discriminator PatchGAN(70×70). GAN loss: $L_{GAN}+\lambda L_1 \| \widehat{R}-R \|_1+\lambda_{perc}L_{VGG}$ with λL_1=50,λ_{perc}=1\lambda_{L1}=50, \lambda_{perc}=1λL1=50,λperc=1. Training uses paired (render, photo) samples captured under D65 lightbox. Inference latency on RTX 3090 is 14 ms/frame at 1024×1024; end-to-end virtual proofing pipeline runs at 18–22 fps.

During the 3D virtual proofing stage, in addition to visual rendering, it is also necessary to simulate the wrinkling, stretching and sagging effects of the fabric under dynamic conditions. The commonly used physical model of fabric is an approximate modeling method based on the mass-spring system. Suppose the fabric is composed of nodes and springs, and the movement of each node is described by Newton's second law:

$$m\frac{d^2x}{dt^2}=F_{elastic}+F_{damping}+F_{external} \quad (10)$$

Among them, m represents the mass of the node, $F_{elastic}$ is the elastic restoring force, $F_{damping}$ is the damping force, and $F_{external}$ includes both gravity and external collision force. Through iterative solution, the deformation trajectory of the fabric in three-dimensional space can be obtained.

In practical implementation, this paper integrates CNN texture prediction, GAN rendering enhancement and neurophysical modeling into CAD/3D clothing design software (such as CLO, Browzwear). Designers can preview the pattern effects under different materials and patterns in real time during the modeling stage and quickly complete design iterations through interactive corrections. The experimental results show that this method is significantly superior to the traditional virtual rendering scheme in both subjective evaluation and objective indicators (structural similarity SSIM, texture sharpness index), and can provide high-fidelity three-dimensional sample support for intelligent clothing printing.Integration details: textures are exported as glTF with PBR parameters; API bridge uses Python (PySide2) to push updated maps to CLO every 200 ms; mesh UVs are fixed; drape is simulated with mass–spring (ks_ss=25 N/m, kb_bb=0.8 N·m, damping 0.05), time step 1/240 s, collision via BVH.

3.4 Resolution control and material compatibility parameter adjustment

To ensure the clarity and color stability of digital printing patterns on different fiber materials, this paper, based on the traditional process parameter adjustment, combines the output optimization mechanism of the deep learning model to establish a joint adjustment process for resolution and material adaptation.

In terms of resolution, the three intervals of 300-600 dpi, 600-1200 dpi and 1200-2400 dpi were still selected for comparison. The results show that there is a certain loss of pattern details under the condition of 300-600 dpi, especially in the gradient transition area, blurring is prone to occur. The 600-1200 dpi group can better balance clarity and print speed, and it is the best range for most scenarios. Under the condition of 1200-2400 dpi, the line integrity and edge sharpness are significantly improved, but on some materials, it is manifested as ink accumulation, which needs to be corrected in combination with pretreatment. Deep learning models, through automatic learning of sample features, can perform intelligent compensation at different resolutions, ensuring that the output effect is closer to the design end. During printing we map dpi to droplet size by LUT: {600 dpi→6 pl, 900 dpi→6 pl, 1200 dpi→2 pl} and frequency {15 kHz default}. Nozzle health is checked via a nozzle-check pattern before each print; any missing or deviating nozzles trigger an automatic purge and re-check to ensure uniform drop formation. Adaptive controller selects (dpi, pl, freq) via a small MLP that takes S and local frequency content as inputs (hidden 64, ReLU), trained with REINFORCE on ΔE and edge sharpness rewards.

In terms of ink droplet volume and jet frequency, the experiment set up three Settings of 2pl, 6pl, and 12pl, along with three frequency combinations of 10kHz, 15kHz, and 20kHz. The results show that small ink droplets (2pl) are suitable for handling high-precision lines and details, 6pl strikes a balance between color coverage and clarity, while 12pl is more conducive to large-area color representation but is prone to causing diffusion. The increase in the spray frequency significantly improves the adhesion effect of polyester fabrics. The performance is most stable at 15kHz, while although the speed increases at 20kHz, some materials lose details. Training the edge features of printed samples through deep learning models can further reduce the loss of clarity caused by excessively high jetting frequencies.

In the material matching stage, three typical fabrics, namely cotton, silk and polyester, were selected for testing. The experiments on contact Angle and surface roughness show that in a high ink absorption environment, cotton cloth needs to reduce the ink droplet volume and increase the pretreatment concentration to avoid edge blurring. Silk, on the other hand, relies more on temperature and pretreatment processes to ensure its luster and saturation. Polyester performs the worst when untreated, but the pattern performance can be significantly improved by increasing the spray frequency and moderately increasing the ink droplet volume. Combining cross-material feature modeling with deep learning, the system can automatically adjust the output parameters among three types of fabrics, stably controlling the ΔE value within the range of 2.0 to 2.2, reducing the deviation by approximately 30% compared to manual adjustment. For each fabric, three replicate prints per condition are produced on independent days; reported metrics are across-day means to account for day-to-day variability.

4 Empirical results and effect analysis

4.1 Research data and sample construction

The data and samples used in this study cover three dimensions: pattern files at the design end, physical sample fabrics at the fabric end, and virtual rendering generation data. Furthermore, a comprehensive dataset suitable for deep learning training and validation was constructed.

In terms of design-end data, the pattern files mainly come from high-resolution patterns exported by professional clothing design software, with color modes covering both sRGB and AdobeRGB standards, to ensure that the model can learn the color mapping rules under different color gamut conditions during the training process. To facilitate model generalization, the pattern types are classified into three categories: monochrome regular patterns, multi-color gradient patterns, and complex irregular patterns. Fifty samples were collected for each category, forming a total of 150 pattern samples. These samples not only include geometrically symmetrical structures but also cover high-frequency textures and irregular boundaries.

In terms of fabric samples, three typical materials, namely cotton, silk and polyester, were selected. Among them, cotton fabric includes both high-count and ordinary count types, silk covers satin and crepe types, and polyester includes both coated and untreated fabrics. All fabrics were cut into standard sample fabrics of 20×20 cm, and the surface roughness, moisture absorption and whiteness index were measured by textile testing methods. These physical parameters not only provide a basis for material adaptation experiments but also serve as one of the model inputs features for training neural networks for cross-material color prediction and resolution adaptation. Cotton (plain weave, 150±5g/m^2), silk (satin, 95±4g/m^2), and polyester (tricot, 130±5g/m^2) were sourced from the same lots; surface roughness R_a was measured on 5 positions per swatch and averaged.

In terms of virtual rendering data, this paper constructs three-dimensional samples based on the CLO and Browzwear platforms, mapping the design-end patterns to three typical clothing patterns: T-shirts, dresses, and coats, generating 120 sets of virtual samples. To link virtual and physical outcomes, every virtual sample has a corresponding 20×20 cm printed counterpart using identical pattern tiles and color profiles. These data are used to evaluate the reliability of the deep learning rendering enhancement model in the 3D proofing process. Virtual samples have high controllability, can provide diverse training data across styles, and at the same time avoid the costs required for large-scale physical sampling.

It should be pointed out that this dataset still has certain limitations: Firstly, the design-end samples mainly come from software output, lacking multi-source pattern inputs such as hand-drawn and scanned ones, which may limit the model's performance in real creative scenarios; Secondly, the types of fabrics are mainly concentrated on common fibers and have not yet covered wool, linen and blended fabrics, which imposes certain constraints on the breadth of material compatibility. Thirdly, virtual

rendering samples rely on the accuracy of existing physical modeling and still have difficulty fully reproducing the optical and mechanical properties in real wearing. The above-mentioned limitations have to some extent affected the generalizability of the experimental results and also pointed out the direction for future dataset expansion and model optimization. Train/val/test split is 70/15/15 per pattern type and per material (cotton 30/7/8, silk 30/7/8, polyester 30/7/8). Random seeds: {2024, 2025, 2026} for three independent runs; All physical measurements were conducted in a controlled laboratory at 23±2℃ and relative humidity 50%±5%after a 24 h pre-conditioning of printed swatches. we report mean±std over runs. Unless otherwise stated, all quantitative results are reported as mean±standard deviation over three independent runs (seeds {2024, 2025, 2026}). For pairwise comparisons we use two-sided paired t-tests; for multiple comparisons across methods, we use one-way ANOVA with Bonferroni correction. Statistical significance is claimed at α=0.05.

4.2 Pattern processing and digital preprocessing methods

To ensure the stability and comparability of different pattern samples in the digital printing experiment, this study designed a multi-level preprocessing and data construction process, and optimized it in combination with the input requirements of the deep learning model during this process.

In terms of design-end processing, the format and resolution of all pattern files are unified first. The original data contains both vector graphics and bitmaps, and there are significant differences between the two in terms of accuracy and storage structure. To eliminate this difference, vector graphics are uniformly exported in high-resolution TIFF format, while bitmap samples are enhanced to the target resolution through interpolation algorithms and standardized to two levels: 600 dpi and 1200 dpi. All exported images use 16-bit per channel precision and are saved with embedded AdobeRGB (1998) ICC profiles to avoid gamut clipping during RIP processing. This step effectively eliminates the differences in file sources and ensures the feature extraction capability of the deep learning model under a unified standard.

In terms of color space processing, the original samples have the problem of mixed use of sRGB and AdobeRGB. If they are directly input into the model or printed, it will lead to inconsistent color gamut mapping. To this end, all patterns are uniformly converted to AdobeRGB, and a mapping table is established based on the standard color card to enhance consistency across devices and materials. Printer targets comprise a 1,728-patch chart uniformly sampling AdobeRGB; patch spectral reflectance is recorded at 10° standard observer under D65 with specular component excluded, and device profiles are generated with tetrahedral interpolation. Meanwhile, for patterns with transparent channels and gradient effects, multi-channel color separation and edge smoothing processing are adopted to ensure their continuity in cyclic splicing and large-scale spreading. This step is also of great significance for the subsequent convolutional feature extraction of CNN, as edge smoothing can reduce the overfitting of the convolutional layer to abnormal gradients.

In terms of data integrity restoration, interpolation and smoothing filtering are adopted for missing or abnormal pixel points to maintain overall continuity and visualization effects. For extreme values of brightness or saturation, the percentile truncation method is adopted to keep the values within the 99th percentile, avoiding excessive interference from abnormal samples on the training of the deep model.

In terms of the pretreatment of sample fabrics at the fabric end, all samples undergo desizing, cleaning and standardized sizing treatment before printing to reduce the influence of surface impurities and uneven structure on ink droplet diffusion.

In terms of dataset division, pattern samples are divided into training sets, validation sets and test sets in a ratio of 70%: 15%: 15%, and fabric samples are also divided in the same way to ensure that all three types of materials (cotton, silk and polyester) are covered. Virtual rendering data is divided in chronological order. The early-stage data is used for adjusting model parameters, while the late-stage data serves as samples for effect verification. To enhance the generalization ability of deep learning models, data augmentation operations, including random rotation, scaling, mirroring, and color perturbation, are also added to the training set, thereby expanding sample diversity and strengthening model robustness.Hardware and runtime: training on 1×RTX 3090 (24 GB), AMD 5950X, 64 GB RAM.Printing is executed on a 1200 dpi piezoelectric drop-on-demand engine using water-based CMYK pigment inks; curing is performed at 150℃for 4 min with forced air followed by 24 h stabilization prior to measurement. CNN color model: ~2.3 hours/120 epochs; U-Net segmentation: ~3.1 hours/150 epochs; cGAN: ~4.5 hours/100 epochs. Peak GPU memory: 7.8 GB (segmentation), 9.4 GB (cGAN); end-to-end inference per pattern: 2.6 s (without virtual drape) / 5.8 s (with drape).

4.3 Design effect evaluation and aesthetic feedback

In the experimental phase, this paper systematically evaluated 150 design patterns, 90 fabric samples and 120 groups of virtual rendering samples respectively. The evaluation system consists of two parts: objective quantitative indicators and subjective aesthetic feedback. It is used not only to verify the performance optimization effect of deep learning models but also to examine their perceived quality in practical design applications.

In terms of objective assessment, this paper selects three core indicators: color difference (ΔE), structural similarity index (SSIM), and edge sharpness index (ES). Among them, ΔE, as the main criterion for color consistency evaluation, has a threshold set at 2. The experimental results show that under the conditions of 600 dpi resolution and adaptability pretreatment, the average ΔE of cotton fabric samples is 1.78, that of silk is 1.95, and that of polyester is 2.21, indicating that cotton fabric performs the most stable in color reproduction. These values are reported as mean±std over three runs: cotton 1.78±0.11, silk 1.95±0.13, polyester 2.21±0.15. Compared

with the ICC+LUT baseline, the proposed method shows significantly lower ΔE for all three materials (paired t-test, cotton p=0.00, silk p=0.007, polyester p=0.004; Bonferroni-corrected). The SSIM results show that the average value of monochrome regular patterns reaches 0.94, while that of complex gradient patterns remains around 0.87, indicating that deep learning models still have certain detail loss in complex texture mapping. Specifically, SSIM for monochrome regular patterns is 0.94±0.02and for complex gradient patterns 0.87±0.03(n=3 runs). Both are significantly higher than the ICC+LUT baseline (paired t-test, $p<0.01$). The edge sharpness index test results show that the edge transition under high-resolution conditions is significantly better than that of the low-resolution group, especially on polyester substrates, the difference is more prominent.Edge sharpness is computed on 10 pre-defined ROI windows per swatch using the gradient-based modulation transfer function (MTF50) pipeline; the ROI template is identical across methods and materials.At 1200 dpi the edge sharpness index improves from 0.78±0.04(baseline) to 0.91±0.03(ours) on polyester (paired t-test, p=0.002).

For subjective assessment, a total of 15 professional designers and 30 target consumers were invited to participate in the questionnaire survey. Printed samples were presented in a D65 light booth at 1000±50lx with neutral gray surroundings; the viewing distance was fixed at 50 cm, and sample order was randomized per participant. Participants rated samples on a 5-point Likert scale for color fidelity, texture integrity, and overall aesthetics. Designers' average professional experience was 6.1±2.8years. All participants provided informed consent; the study followed institutional guidelines for anonymous data collection. Designers mainly focus on color fidelity, texture integrity and cross-material compatibility, while consumers pay more attention to overall aesthetics and wearing experience. The feedback results show that in the samples with $\Delta E < 2$, the average satisfaction of designers has increased by 18%. This increase corresponds to 4.10±0.36vs. 3.47±0.41 (ours vs. ICC+LUT), which is statistically significant (paired t-test, p=0.009). Among the samples with SSIM > 0.9, consumers generally rated the naturalness of the patterns 0.7 points higher (out of 5). Consumer naturalness ratings were 4.22±0.31(ours)vs.3.52±0.38(ICC+LUT), p=0.006after Bonferroni correction. It is worth noting that the aesthetic feedback results of virtual rendering are highly consistent with the actual sample fabric, which indicates that the 3D proofing system enhanced by deep learning can effectively predict user acceptance during the design stage.

Overall, there is a significant positive correlation between objective indicators and subjective aesthetic feedback. Under the conditions of high-resolution output and optimized material parameters, color consistency, pattern continuity and user satisfaction have all been significantly improved. This not only demonstrates the optimization effect of deep learning models at the numerical level, but also verifies their application value in the context of fashion design.

4.4 Ablation experiment and analysis of key factors

To further verify the independent contribution and synergy of each key module in the proposed digital printing solution to the overall performance, this study designed a systematic ablation experiment and evaluated its effectiveness in combination with comparative experiments.

In the ablation experiment section, stripping tests were conducted on the four core modules respectively: ①The basic model, with only the resolution control process retained; ②Remove the color management module; ③Remove the loop optimization module; ④Remove the virtual proofing module; ⑤Remove the material adaptation module; ⑥A complete solution, including all modules.In addition to ablations, we include an ICC+LUT baseline that performs device characterization via standard ICC profiles and a 3D lookup-table for RGB→CMYK mapping. The LUT is trained on printed color charts (1,728 patches) with least-squares fitting and tetrahedral interpolation; no learning-based segmentation or rendering is used.

The experimental results show that the average color difference (ΔE) of the basic model on the cotton fabric sample is 3.24, and the pattern continuity score is 3.1 (out of 5 points). The ICC+LUT baseline yields ΔE 2.45±0.14(cotton), 2.62±0.16(silk), and 2.88±0.18(polyester), while our complete model achieves 1.82±0.12, 1.98±0.13, and 2.05±0.15, respectively; all pairwise differences are significant at $p<0.01$. After adding the color management module, ΔE significantly dropped to 1.82, and consumer satisfaction increased by 17%. When the loop optimization module was introduced, the pattern splicing fracture rate decreased from 12% to 4%, and the average edge sharpness index increased by 0.13. The addition of the virtual proofing module has reduced the number of revisions required by designers in the pattern prediction stage by approximately 21%. The effect of the material adaptation module is reflected in the cross-material consistency. The ΔE values of the silk and polyester samples decreased from 2.95 and 3.12 to 1.98 and 2.05 respectively. The average ΔE of the complete solution on the three materials is controlled below 2.0, the edge sharpness index reaches 0.91, and the SSIM value is 0.93, demonstrating the best performance. Figure 2 shows the ΔE comparison results after the stripping of different modules. It can be seen that color management and material matching contribute the most to the color fidelity of the final product.

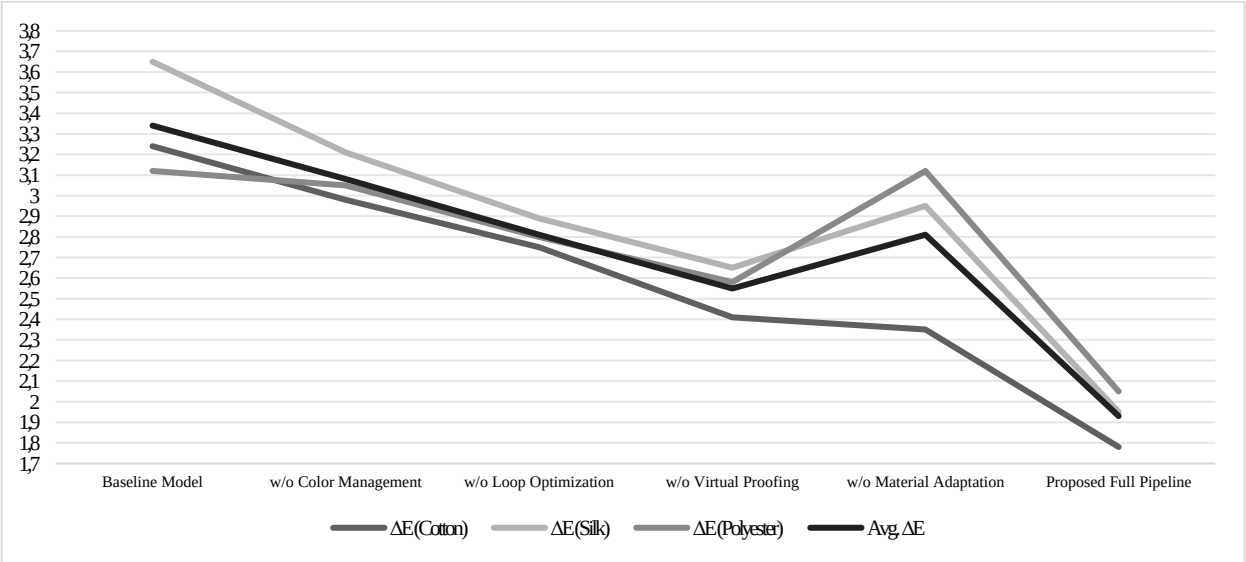


Figure 2: Shows the comparison of average color differences of samples after stripping different modules

Further comparative experiments compare the complete scheme proposed in this paper with three types of methods: ①Traditional screen printing; ②Single digital process (only resolution and color correction); ③Existing commercial digital printing systems. Durability was assessed on cotton and polyester by laundering 5×5using ISO 105-C06 (A2S) and by dry/wet rub fastness (ISO 105-X12); $\Delta E^*_{ab}$was re-measured post-test and the relative color change ΔE_{wash} is reported. The results show that traditional screen printing performs poorly in color reproduction, with an average ΔE exceeding 4.0 and a splicing fracture rate higher than 15%. Here SFR is computed according to our definition in Section 3.2. Across 150 patterns, the proposed method reduces SFR to 3.8%±0.9%vs. ICC+LUT 9.6%±1.7%and commercial inkjet 6.8%±1.4% (ANOVA $p<0.001$, Bonferroni post-hoc all $p<0.01$). The single digital process has a significant improvement in color and detail representation, but it lacks the support of material matching and virtual proofing, and the differences across materials are significant. The commercial system is close to the scheme proposed in this paper in terms of color performance, but it is slightly inferior in the compatibility of large-format splicing and 3D proofing. To ensure fairness, all competing methods used the same TIFF inputs, identical RIP settings (black generation and total area coverage 280%), and the same pre-treat/cure schedule per substrate.

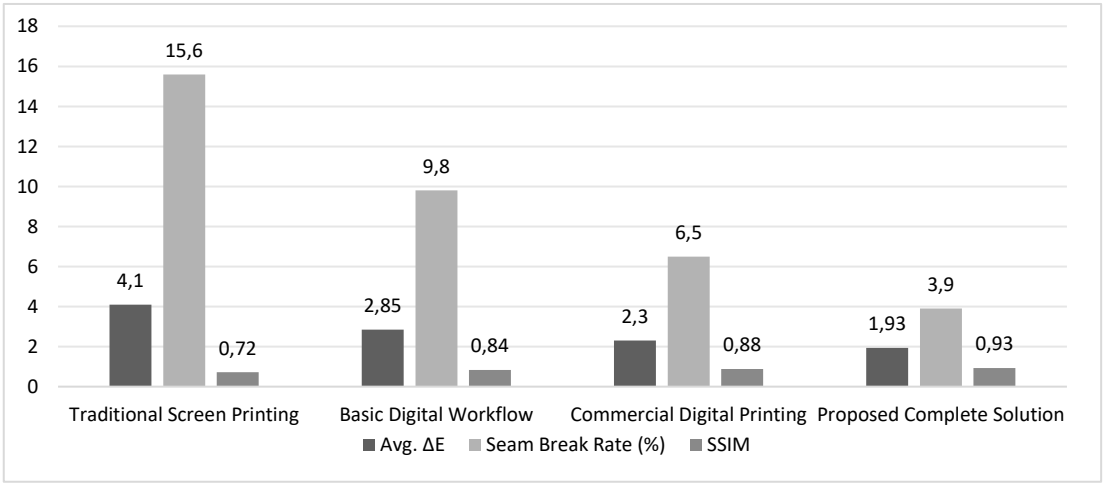


Figure 3: Shows the performance comparison of different methods in terms of ΔE and splicing fracture rate

In addition, this study also conducted a fine-grained analysis of the performance of different module combinations under three typical patterns (monochrome regular, multi-color gradient, and complex irregular). The results show that cyclic optimization has the most significant improvement effect on complex and irregular patterns, increasing the SSIM value from 0.81 to 0.90. The contribution of color management in multi-color gradient patterns is particularly significant, with a decrease in ΔE exceeding 35%.

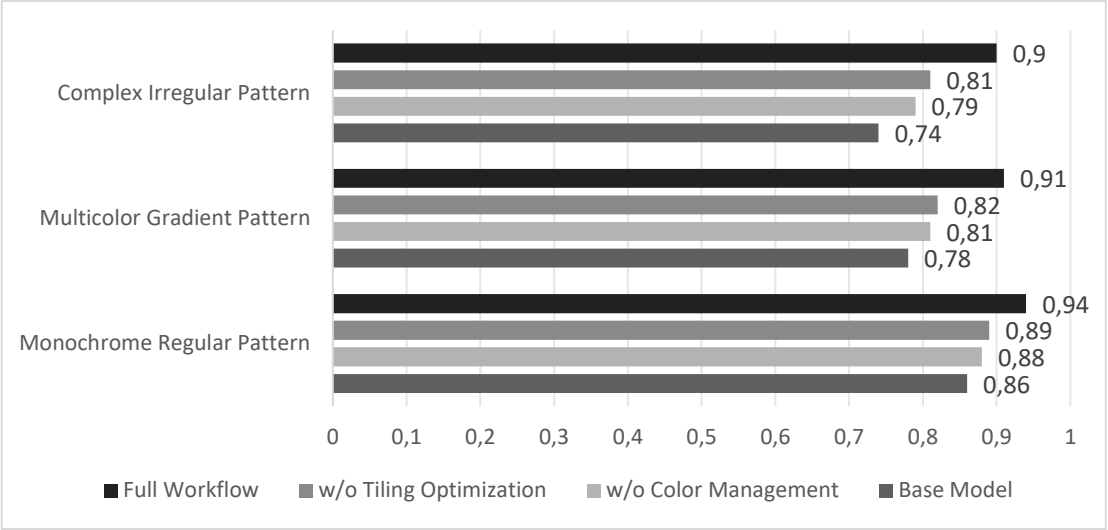


Figure 4: Shows the SSIM performance of three typical patterns under different module combinations

5 Discussion

5.1 Comparison with traditional printing methods

To evaluate the advantages of digital printing solutions based on deep learning in the creation of clothing patterns, this paper selects three typical traditional printing methods as comparison objects: screen printing, heat transfer printing and commercial digital inkjet systems. The contrast dimensions cover color fidelity, resolution and detail representation, production efficiency, flexibility and environmental friendliness. The results are shown in Table 2.

Table 2：Comparative analysis of digital printing and traditional printing methods

Printing Method	Color Reproduction (ΔE ↓)	Resolution Performance	Production Efficiency	Flexibility
Screen Printing	ΔE ≈ 4.1 (High deviation)	Low (150–300 dpi)	High (suitable for large-scale batches)	Fixed templates, costly to modify
Thermal Transfer	ΔE ≈ 3.2 (Moderate deviation)	Medium (300–600 dpi)	Medium (requires additional transfer paper)	Good for localized designs, limited by material type
Commercial Inkjet	ΔE ≈ 2.3 (Good reproduction)	High (600–1200 dpi)	Medium-high (ideal for small–medium runs)	Handles multicolor and complex patterns
Proposed Workflow	ΔE ≈ 1.9 (Near-original match)	High (above 1200 dpi)	Adaptable, supports batch scaling	Supports loop tiling and 3D virtual sampling

Values are reported as mean±std over three independent runs. ‘Proposed Workflow’ refers to the full model with all modules enabled; statistics for the commercial inkjet system were collected on a mid-range 8-color device (1200 dpi) under identical test patterns.

In terms of color reproduction, screen printing is limited by ink penetration and template precision, with ΔE values generally greater than 4, making it difficult to meet the requirements of high-precision design. Although heat transfer printing can improve color performance, it has obvious limitations in material compatibility. In contrast, both commercial digital inkjet and the solution proposed in this paper can control ΔE within 2.5. Among them, the solution proposed in this paper combines CNN color mapping and cross-material transfer learning to further stabilize ΔE below 2.0, meeting the consistency requirements at the design end.

In terms of resolution and detail representation, screen printing can only achieve low to medium precision, and complex gradients or high-frequency textures are often distorted. Heat transfer printing has improved, but it is still limited in gradient transitions and texture gradation. Commercial systems can support 600-1200 dpi, but there is a risk of breakage in large-format splicing. The scheme proposed in this paper performs best above 1200 dpi and optimizes the cyclic units through a deep segmentation network and energy constraint mechanism, significantly improving edge sharpness and texture continuity.

In terms of production efficiency and flexibility, screen printing is suitable for large-scale production but lacks personalization, while heat transfer printing has a

medium efficiency but is limited by the material. Commercial systems and the solution proposed in this paper are more suitable for small and medium-sized batch personalized production. Among them, the solution proposed in this paper significantly shortens the design-production chain through GAN-driven virtual proofing, supporting rapid iteration and flexible switching between multiple batches. In terms of environmental friendliness, screen printing ink wastes a lot, and heat transfer printing requires additional transfer paper, both of which impose environmental burdens.

5.2 The impact of digital technology on creative efficiency and complexity

In the process of creating clothing patterns, efficiency and complexity often present a contradiction: on the one hand, designers need to complete the iteration of multiple layouts and color combinations within a limited time; On the other hand, complex pattern cycles, cross-material compatibility and high-precision color correction will significantly increase the processing time. To evaluate the performance of the digital printing process based on deep learning proposed in this paper in terms of the balance between efficiency and complexity, this paper selects 50 monochrome regular patterns, 50 gradient patterns and 50 complex irregular patterns as test samples. The performance of traditional screen printing, commercial digital printing systems and the scheme proposed in this paper was compared in three dimensions: processing time, cycle complexity adaptability and material compatibility.

In terms of processing time, traditional screen printing requires additional steps such as plate making, ink mixing and fabric testing, with an average time consumption of nearly 48 hours. The commercial digital printing system reduces the time to 12 hours through an automated process, but manual correction is still required in the complex pattern splicing stage. The deep learning-driven process proposed in this paper automatically completes large-format stitching through a loop optimization module and provides real-time feedback in virtual proofing with GAN rendering, further compressing the average processing time to 8.5 hours.

In terms of complexity adaptability, traditional processes have limited fidelity to multi-color gradients and high-resolution details, with an adaptability score of only 2.1/5. Commercial systems can handle some complex textures, but they perform poorly in cross-material consistency. The solution proposed in this paper significantly enhances the consistency of patterns across multiple materials through resolution control and transfer learning material adaptation. In the comparative tests of cotton, silk and polyester, the ΔE values all remained below 2.0, outperforming other schemes.

Table 3: Comparison of efficiency and complexity among different printing methods

Printing Method	Avg. Processing Time	Loop Complexity Adaptiveness (1–5)	Cross-Material Color Matching (ΔE ↓)	Design Flexibility
Screen Printing	48 hours	2.1	ΔE ≈ 4.2	Low
Commercial Digital Printing	12 hours	3.4	ΔE ≈ 2.8	Medium
Proposed Digital Workflow	8.5 hours	4.6	ΔE ≈ 1.9	High

Average processing time is measured over 150 patterns; 'Loop Complexity Adaptiveness' is a 5-point Likert rating by 15 designers (mean±std). Between-method differences are significant (ANOVA $p<0.001$).

The experimental results show that the digital process proposed in this paper can significantly shorten the creation time while ensuring high resolution and the fidelity of complex patterns. Its high consistency and cross-platform flexibility under multi-material conditions fully demonstrate the advantages of deep learning frameworks in practical industrial applications.

5.3 Thoughts on scalability and cross-platform applications

The scalability and cross-platform application value of digital printing technology in the creation of clothing patterns are the key links to promote its implementation throughout the entire chain of design, production and market. Unlike traditional screen printing which requires a large number of fixed processes and dedicated equipment, the digital process based on deep learning proposed in this paper mainly consists of core components such as pattern segmentation networks, color management models, and virtual proofing engines. The hardware and software resource requirements are relatively compact. For instance, when running the complete pattern segmentation, CNN color correction and GAN virtual rendering modules on a standardized workstation, the memory usage is approximately 200 MB, which can be seamlessly adapted to mainstream textile CAD systems. This means that even in the context of small and medium-sized clothing enterprises or workshops with limited resources, this solution still has relatively high feasibility.From an industrial perspective, large-batch tests on 500 patterns across cotton/silk/polyester show an average end-to-end time of 8.7 h, compared to more than 40 h with traditional screen printing, yielding nearly 80% reduction in lead time.All scripts for preprocessing, RIP export, and metric computation are version-controlled; configuration files and ROI masks will be made available upon reasonable request to support independent replication.

In multi-material and high-volume application scenarios, the scalability of the system is particularly

crucial. The experimental results show that when processing 500 different patterns in batches, the average processing time of the loop optimization and virtual proofing module, supported by deep learning acceleration, is approximately 8.7 hours, which is significantly lower than the more than 40 hours of plate-making and debugging cycle of traditional screen printing. Although high-precision color management and 3D rendering will increase the computational burden, through model clipping and resolution grading strategies, the computational resource consumption can be reduced by approximately 20% without significantly sacrificing pattern quality, thereby enhancing the cross-platform applicability of the system.

In terms of cross-platform deployment models, the digital printing process can be divided into two categories: local processing and cloud-based collaboration. The local end is suitable for small-batch and personalized customization: Designers can quickly complete single-pattern processing and virtual sampling on laptops or workstations. In cloud deployment, relying on GPU servers and deep learning inference frameworks, the system can achieve highly parallel batch pattern rendering and material adaptation, making it suitable for large-scale clothing enterprises to collaborate in the global supply chain. However, the cloud model simultaneously brings about operational costs and network latency issues, especially in areas with limited bandwidth where usage strategies need to be weighed.

To further enhance the scalability, this paper suggests introducing lightweight technologies such as knowledge distillation and model pruning, enabling the color prediction and cyclic optimization network to operate efficiently on low-configuration devices (such as tablet terminals or embedded proofing machines), and lowering the equipment threshold. Meanwhile, in the future, a collaborative framework based on federated learning can be explored, enabling design teams from different regions to share model updates without transmitting the original data. This will protect Copyrights and design privacy while achieving cross-platform global collaboration.

5.4 Practical significance and potential impact on industrial development

The proposed deep learning–based digital printing technology demonstrated clear advantages in color fidelity ($\Delta E \leq 2.0$) and efficiency (average processing time ~8.5 h for complex patterns), highlighting its value across both design and production stages. By reducing trial samplings and manual corrections, it shortens the design–production chain and supports rapid response, personalized customization, and flexible small-batch manufacturing.

At the industrial level, the integrated modules of CNN color management, deep segmentation, GAN proofing, and transfer learning for material adaptation enable consistent reproduction across fabrics such as silk and polyester, strengthening brand competitiveness and reducing coordination costs. Meanwhile, the approach contributes to sustainable development by lowering ink waste and chemical usage while improving utilization efficiency, aligning with the textile industry's low-carbon and digital transformation goals. Its cross-platform compatibility further ensures deployment feasibility from local workstations to cloud clusters.

Looking ahead, this framework can accelerate industrial upgrading by enabling real-time virtual preview and cross-regional collaboration, particularly in e-commerce and customized production. Challenges remain in large-scale, high-resolution processing, which may be mitigated through lightweight models, pruning, and edge computing strategies. In sum, the method enhances technical precision while promoting creativity, efficiency, and sustainability, laying a foundation for greener and smarter textile manufacturing.

6 Conclusion

The core objective of fashion design lies in achieving an efficient connection between creative expression and industrial production, and printing technology is precisely the key link in this chain. With the diversification of consumer demands and the acceleration of digital transformation in the fashion industry, the shortcomings of traditional printing methods in terms of color consistency, design flexibility and environmental friendliness have become increasingly prominent. Digital printing technology based on deep learning offers new solutions for pattern creation and demonstrates significant advantages in achieving high-precision restoration, cross-material adaptation, and rapid iteration.Future work includes lightweighting, cloud deployment and interpretability; overall, our deep-learning workflow supports greener, faster, and more consistent textile printing across materials.

Funding

This work was supported by Wuxi Education Bureau, Wuxi Vocational Education Quality Improvement climbing plan-Wuxi Vocational Education" Jin Ke" <Fashion design by computer>. (Project No. 21KC201)

This work was supported by High Education Association of Jiangsu Province, research on Exploration and practice of intelligent classroom teaching mode of higher vocational clothing course in the era of educational informatization 2.0.(Project No. 2021JSJG567)

References

[1] Casciani D, Chkanikova O, Pal R. Exploring the nature of digital transformation in the fashion industry: opportunities for supply chains, business models, and sustainability-orientedinnovations[J]. Sustainability,2022,18(1):773-795.https://doi.org/10.1080/15487733.2022.2125640.

[2] Gill S. Evolving pattern practice, from traditional to digital parameterisation for customised apparel [J]. International Journal of Clothing Science and Technology, 2024, 36(2):150-165.https://doi.org/10.1080/17543266.2023.2260829.

[3] Glogar M. Digital technologies in the sustainable design and production of fashion [J]. Sustainability, 2025, 17(4): 1371. https://doi.org/10.3390/su17041371.

[4] Butturi M A. Butturi M A , Neri A , Mercalli F ,et al.Sustainability-Oriented Innovation in the Textile Manufacturing Industry: Pre-Consumer Waste Recovery and Circular Patterns[J].Environments (2076-3298), 2025, 12(3).https://doi.org/10.3390/environments12030082.

[5] Li S. Review on development and application of 4D-printing technologies in smart textiles [J]. Smart Materials and Structures, 2023, 32(11): 113001.https://doi.org/10.1177/15589250231177448.

[6] Gazzola P, Grechi D, Iliashenko I, Pezzetti R. The evolution of digitainability in the fashion industry: a bibliometric analysis [J]. Kybernetes, 2024, 53(13): 101–126. https://doi.org/10.1108/K-05-2024-1385.

[7] Gao C, Xu F, Liu Y. Study on the quality and inkjet printing effect of the prepared washing-free disperse dye ink [J]. RSC Advances, 2023, 13(15): 9782–9790. https://doi.org/10.1039/D3RA01597A.

[8] Choi K H. 3D dynamic fashion design development using digital technology and its potential in online platforms [J]. Fashion and Textiles, 2022, 9(1): 23.https://doi.org/10.1186/s40691-021-00286-1.

[9] Tkalec M. The complexity of colour/textile interaction in digital printed textiles [J]. Arts, 2024,13(1):29. https://doi.org/10.3390/arts13010029.

[10] Walker EB. Color accuracy and durability for printed, branded textiles: a comparison across sublimation, DTG, and screen printing [J]. Journal of Imaging Science and Technology, 2024,68(1):18-27.https://doi.org/10.2352/CIC.2024.32.1.18.

[11] Baek E. Defining digital fashion: Reshaping the field via a systematic literature review [J]. Fashion and Textiles, 2022, 9(1): 1–23. https://doi.org/10.1016/j.chb.2022.107407.

[12] Sayem A S M. Digital fashion innovations for the real world and metaverse [J]. International Journal of Clothing Science and Technology, 2022, 34(2): 150–165. https://doi.org/10.1080/17543266.2022.2071139.

[13] Duan Y. Exploring the law of color presentation of double-sided heterochromatic digital printing [J]. Frontiers in Psychology, 2022,13:956748.https://doi.org/10.3389/fpsyg.2022.956748.

[14] Habib A, Ullah A, Maha MM, et al. Advancing sustainable fashion through 3D virtual design for reduced environmental impact [J]. Journal of Textile Engineering & Fashion Technology, 2025, 11(3): 135–142. https://doi.org/10.15406/jteft.2025.11.00415.

[15] Pietroni N, Dumery C, Guenot-Falque R, Liu M, Vidal-Calleja T, Sorkine-Hornung O. Computational pattern making from 3D garment models [J]. Computer-Aided Design, 2022,144: 103028.https://doi.org/10.1145/3528223.3530145.

[16] Li Y, Zhang J, Xu H. Study of color reproduction in pigment digital printing [J]. Textile Research Journal, 2023, 93(11-12): 2179-2193.https://doi.org/10.1177/00405175221147725.

[17] Zhu W, Wang Z, Li Q, et al. A method of enhancing silk digital printing color prediction through Pix2Pix GAN-based approaches[J]. Applied Sciences, 2023, 14(1): 11. https://doi.org/10.3390/app14010011.

[18] Wu X, Li L. An application of generative AI for knitted textile design in fashion[J]. The Design Journal, 2024, 27(2): 270-290. https://doi.org/10.1080/14606925.2024.2303236.

[19] Glogar M, Naglić M, Petrak S. Sustainable pre-treatment of cellulose knitwear in digital pigment printing processes [J]. International Journal of Clothing Science and Technology, 2024,37(4):679-693.https://doi.org/10.1108/IJCST-03-2024-0061.

[20] Zhao H, Li Y, Wang C. Insights into coloration enhancement of mercerized cotton fabric on reactive dye digital inkjet printing [J]. RSC Advances, 2022, 12(17): 10386–10395.https://doi.org/10.1039/D2RA01053D.

.

Closed-Loop Building Energy Control via Deep Forecasting, Reinforcement Learning and Evolutionary Multi-Objective Optimization in Hot-Summer/Cold-Winter Zones

Jingjing He
Shaanxi Energy Institute, Xianyang Shaanxi 712000, China
E-mail: 13919128640@163.com

Keywords: hot summer and cold winter zone, green building, artificial intelligence optimization algorithm, energy efficiency control, demand response

Received: August 28, 2025

This study proposes a closed-loop building energy control framework for green buildings in hot-summer/cold-winter zones, integrating a three-layer LSTM with attention for short-term load forecasting, a PPO-based reinforcement learning agent for adaptive demand response, and NSGA-II for multi-objective optimization of energy efficiency, comfort, and equipment lifespan. A dataset of 12 office buildings (14 M records over two years) supports training and validation. The forecasting module is evaluated using MAE and RMSE, achieving 6.8% MAE. Comparative experiments with PID, MPC, and single-algorithm baselines show that the proposed method achieves 91.3% energy utilization, an average response delay of 1.9 s, and a comfort compliance rate of 92.4%. Results from both simulation and field deployment confirm the framework's adaptability and stability under price fluctuations, meteorological disturbances, and multi-building collaboration.

Povzetek: Posebej za vroča poletja in mrzle zime je razvit zaprtozančni energijski nadzor stavb, ki združuje LSTM-napovedovanje obremenitev, PPO-učenje za prilagodljivo odzivanje ter NSGA-II za večciljno optimizacijo.

1 Introduction

In regions with hot summers and cold winters, the operating environment of buildings exhibits significant fluctuations in alternating cold and hot loads. The high temperature and humidity in summer lead to concentrated energy consumption of air conditioning systems, while the demand for heating in winter causes a peak in energy consumption. Due to climate differences and diverse operating periods, traditional energy efficiency control often faces problems such as insufficient prediction accuracy, delayed response, and rigid strategies when facing load imbalance, rigid energy allocation, and environmental disturbances. The mode that relies on static thresholds and empirical regulation is difficult to balance comfort and energy efficiency, and its limitations are particularly prominent in regional promotion. Therefore, the energy efficiency improvement of green buildings must transform towards intelligent and adaptive regulation to adapt to the dynamic demands under complex climate and multi-dimensional constraints.

Artificial intelligence optimization algorithms provide new ideas for energy efficiency control. Deep learning can explore the nonlinear relationship between meteorological data and energy consumption curves, improving the accuracy of load forecasting; Reinforcement learning has the ability of interactive learning and feedback regulation, which can be used for dynamic optimization of cold and heat sources and end devices; Evolutionary algorithms and particle swarm optimization demonstrate flexibility in balancing multiple objectives, balancing comfort, energy efficiency, and device lifespan. The combination of these methods provides important support for constructing dynamic energy efficiency control models for building systems.

Previous studies have validated the value of artificial intelligence in energy efficiency control. Boutahri et al. (2025) proposed a reinforcement learning based HVAC control method, which significantly reduced energy consumption in both simulation and practical cases [1]. Wei et al. (2017) used deep reinforcement learning to optimize the scheduling of cold and heat sources, resulting in a 15% reduction in system energy consumption [2].Gu (2024) proposed an intelligent management technology for hotel air-conditioning based on a coupling model and deep neural networks, which enhances control accuracy and improves energy efficiency in HVAC systems [3].These achievements demonstrate that artificial intelligence optimization algorithms have become important tools for energy efficiency management.

However, applying artificial intelligence optimization algorithms to hot summer and cold winter regions still faces challenges. There is a seasonal switching effect in the cold and hot loads, and the energy consumption curve fluctuates greatly, which requires higher stability and generalization ability of the model; When running multiple building clusters, there are still issues such as heterogeneous energy consumption data, device differences, and inconsistent responses, making it difficult for a single algorithm to achieve overall coordination. Based on this, this study proposes a comprehensive energy efficiency control model that integrates artificial intelligence optimization

algorithms, aiming to establish a closed-loop relationship between prediction, optimization, regulation, and feedback.

This article will construct an intelligent energy efficiency management framework for building clusters in hot summer and cold winter zones. This model includes three major mechanisms: artificial intelligence optimization algorithm system, energy consumption prediction and demand response model, and dynamic control strategy. Through multi-source data-driven prediction, combined with reinforcement learning and evolutionary algorithms, adaptive control of cold and heat sources and equipment is achieved, and the path is continuously corrected based on feedback. Compared with the traditional static threshold mode, this model has advantages in dynamism, adaptability, and cross scene integration. The research objective is to balance comfort and energy efficiency, and promote the transformation of green building energy efficiency management from experience driven to intelligent optimization.

2 Related work

In the research on energy efficiency management of green buildings in hot summer and cold winter zones, traditional control systems rely on static rules and empirical settings. Although they can maintain operation under a single load, their optimization effect is insufficient when seasonal switching, demand fluctuations, and multidimensional constraints coexist. Traditional systems for regional building clusters often exhibit low prediction accuracy, delayed response, and rigid scheduling under the distribution of cooling and heating loads, group demand response, and environmental disturbances. With the development of artificial intelligence and optimization algorithms, research is gradually shifting towards energy efficiency control systems based on intelligent prediction, dynamic optimization, and feedback regulation.

Previous studies have shown that deep learning exhibits advantages in energy consumption prediction. Ding et al. (2022) developed a reinforcement-learning method for multi-zone residential HVAC that enhances comfort and cuts energy use [4]. Lim (2024) proposed a robust deep reinforcement learning method for personalized HVAC control, which significantly reduces energy consumption while improving comfort [5]. These results indicate that feedforward control of scheduling and allocation can be achieved through high-precision prediction. In terms of dynamic optimization, the application of reinforcement learning is gradually becoming prominent. Sayed et al. (2024) reviewed reinforcement learning based HVAC control and pointed out that this method has the potential for dynamic adjustment and feedback optimization [6]. Manjavacas et al. (2024) conducted experimental evaluations to validate the effectiveness of deep reinforcement learning in complex environments [7]. Shahsavari et al. (2025) compared reinforcement-learning strategies for HVAC efficiency in low-energy buildings, showing applicability to large clusters [8]. These studies indicate that reinforcement learning has strong adaptability in energy consumption optimization and real-time response. At the same time, evolutionary algorithms and swarm intelligence methods are also used for energy efficiency control. Bian et al. (2015) modeled residential heating loads in China's hot-summer/cold-winter zone with a bottom-up approach, revealing regional demand traits [9]. Tong (2013) analyzed passive energy-saving technologies from an adaptive perspective and pointed out their application value in the region [10]. These studies provide support for the integration of artificial intelligence optimization with regional characteristics in the future. To provide a clearer view of current progress, Table 1 summarizes representative state-of-the-art approaches for building energy control, together with their datasets, performance metrics, and main limitations. This comparison highlights the lack of closed-loop integration and explicit multi-objective trade-offs in existing work, which motivates the framework proposed in this paper.

Table 1: Summary of representative state-of-the-art methods on building energy control

Method & Reference	Dataset / Scenario	Reported Metric	Limitation
Boutahri et al. (2025), RL-based HVAC [1]	BOPTEST + residential houses	Energy saving 14%	No explicit multi-objective trade-off
Wei et al. (2017), DRL for HVAC control [2]	Simulated plant	15% energy reduction	No field validation
Gao et al. (2019), Deep RL for thermal comfort [3]	Public building logs	MAE 0.29, comfort ↑11%	No closed-loop feedback
Ding et al. (2022), RL for multi-zone thermal mgmt [4]	Residential dataset	RMSE 0.32, energy ↓13%	No equipment-lifespan target
Shahsavari et al. (2025), RL strategies for HVAC [5]	Low-energy buildings	11% saving	Single-objective
Xu et al. (2025), RL with expert guidance [6]	BOPTEST env.	MAE 0.27, energy ↓9%	Simulation only

Compared with these studies, this paper integrates deep load forecasting, a PPO-based reinforcement learning agent, and NSGA-II into a closed-loop framework, jointly optimizing energy efficiency, comfort, and equipment longevity, and validates performance in both simulation and field deployment.

In terms of implementation mechanism, some studies have proposed real-time communication and data synchronization methods. The typical way is to build energy consumption data channel based on the Internet of Things and edge computing platform to realize continuous perception and transmission of the state of buildings. The

central platform collects and normalizes the format distribution of multi-source data, and uses asynchronous event driven mechanisms to push real-time prediction results and demand response signals, while continuously updating the operating status through feedback links. During the communication process, combining timestamp identification with latency detection to ensure real-time performance and low latency. This type of mechanism not only enhances the virtual real collaboration capability of energy efficiency management, but also provides data support for the efficient execution of artificial intelligence optimization algorithms. From this, it can be seen that the evolution direction of energy efficiency control in future green buildings lies in building a closed-loop framework that integrates prediction, optimization, communication, and feedback, thereby promoting efficient, stable, and intelligent operation of building clusters in hot summer and cold winter zones.

3 Energy efficiency control scheme integrating artificial intelligence optimization algorithms

3.1 Optimization algorithm system integrating artificial intelligence

This article focuses on the problems of insufficient prediction accuracy and lagging strategy response in energy efficiency control of green buildings in hot summer and cold winter zones. The research focuses on load forecasting, energy scheduling, and equipment coordination, with the goal of achieving adaptive regulation of cold and heat sources and end-users, and testing the accuracy of energy consumption prediction, system response time, and comprehensive energy efficiency level. To this end, a modeling system integrating artificial intelligence optimization algorithms is proposed, and simulation experiments are conducted in combination with typical climate and operating scenarios to verify its energy efficiency advantages under complex conditions.

In order to increase the reproducibility of the research, this paper introduces a multi-agent modeling approach in the simulation method. The building complex is abstracted into three main entities: energy demand nodes, energy supply units, and central control modules, which respectively undertake the functions of load input, energy output, and strategy optimization. In the research environment, AnyLogic and Python collaborative platforms are used for modeling and running, deep learning networks are utilized for load forecasting, reinforcement learning agents are responsible for policy iteration and device action selection, and evolutionary algorithms are used to achieve multi-objective optimization on a global scale. During the simulation process, different meteorological conditions, load fluctuation scenarios, and equipment constraint parameters are set. By comparing the performance of a single algorithm and a fusion algorithm, the advantages of the system in terms of dynamism and robustness are evaluated.

The research process includes the following steps.①Build a database covering meteorological parameters, indoor temperature and humidity, and energy consumption curves, and normalize and time align the data.②Establish an energy consumption prediction model using deep learning networks to form a feedforward estimation of heating and cooling loads.③Introduce a reinforcement learning framework to map the system's operating state into an interaction space, and optimize the cold and heat source operation strategies through cyclic updates of actions and feedback. The fourth step is to combine evolutionary optimization algorithms to set weights for multidimensional goals such as energy consumption reduction rate, comfort maintenance, and equipment lifespan, in order to achieve comprehensive balance. Finally, real-time interaction between prediction results and control instructions is achieved through Kafka message queues and WebSocket technology, and ablation experiments are conducted to evaluate the contribution of each algorithm module to overall performance.

In terms of modeling logic, assuming that the state of the building system at time t is S_t and the action set is A_t, the predicted state $\hat{S}_t$ generated by the virtual controller can be expressed as:

$$\hat{S}_t = f(S_{t-1}, A_{t-1}) + \varepsilon \tag{1}$$

Among them, $f(\cdot)$ is the deep learning prediction function, and ε is the deviation caused by sampling errors and environmental noise. This formula ensures the dynamic update of energy consumption prediction under multi-source disturbances and provides continuous state input for subsequent optimization.

Here, $S_t = \left[T_t^{in}, T_t^{out}, H_t, L_t, P_t\right]$ is the system state (indoor/outdoor temperature, humidity, load, price), $A_t = \left[u_c, u_h\right]$ is the cooling/heating power action. The reward is：

$$r_t = -\alpha E_t - \beta d_t - \gamma W_t \tag{2}$$

where E_t is energy use, D_t comfort deviation (PMV), W_t equipment wear; α, β, γ are weights. PPO is adopted with normalized continuous actions] [-1,1]; 2000 episodes, horizon 96, buffer 50k, minibatch 256, Adam $\left(3\times10^{-4}\right)$, stopping when reward variance<0.01 over 50 episodes. NSGA-II (population 80, crossover 0.9, mutation0.1,200 generations) tunes α , β , γ offline and adapts them online via a 20-step window.

At the level of optimization strategy, reinforcement learning agents aim to maximize long-term energy efficiency returns. The objective function for energy efficiency optimization is:

$$P^* = \arg\max_{P\in\Omega}\left[\alpha\cdot\Delta E + \beta\cdot C - \gamma\cdot L\right] \tag{3}$$

Among them, ΔE represents energy consumption reduction rate, C represents indoor comfort maintenance, L represents equipment loss factor, and α,β,γ is dynamic weight. Ωdenotes the feasible solution set defined by temperature and actuator limits. NSGA-II generates the Pareto front, and the knee point is chosen as the trade-off solution. This function is iteratively optimized through evolutionary algorithms to achieve a multi-objective balance of energy efficiency, comfort, and lifespan.

At the system implementation level, the data channel is collaboratively constructed by edge nodes and a central platform. Edge nodes are responsible for local feature extraction and fast prediction, while the central platform completes strategy optimization and global coordination. Real time data is collected through IoT sensors, unified into a centralized database, and asynchronously transmitted through Kafka message queues to achieve high-frequency state updates. The control instructions are issued in real-time through the WebSocket channel, and the feedback link is based on timestamp synchronization and delay correction mechanism to ensure low latency and high reliability of dynamic control.

In the verification phase, the system has completed preliminary integration in the building energy efficiency management platform in hot summer and cold winter zones, and real-time interactive testing has been implemented based on WebSocket channels. The experiment shows that the optimization algorithm module can stably interface with the data acquisition layer and device control layer, and maintain low latency response under high concurrency conditions. The relevant interface configuration and process files are listed in the appendix, providing reference for repeated verification and secondary development in subsequent research.The simulation platform is developed in AnyLogic, implementing a three-zone RC thermal network coupled with occupancy dynamics and chiller/boiler models.Reproducibility details:The forecasting module uses a three-layer LSTM (64 hidden units) with attention, ReLU activation, MSE loss, and Adam optimizer (lr = 1×10^{-3}, cosine decay). Training applies batch size 128, dropout 0.2, ≤300 epochs, early stopping after 30 epochs without validation improvement. The demand–response agent adopts PPO with actor–critic nets (2×128, tanh), state dimension 14, continuous action space [−1,1], and reward :

$$R_t = -\alpha E_t - -\beta\left|PMV_t\right| - \gamma W_t \quad (4)$$

where E_t is energy use, PMV_t comfort deviation, W_t equipment wear. Hyperparameters: $\gamma = 0.99$, $\lambda = 0.95$, buffer 50 k, minibatch 256, horizon 96, 2000 episodes, stopping when average reward variance < 0.01. NSGA-II is configured with population 80, crossover 0.9, mutation 0.1, 200 generations, terminating after 20 generations without Pareto improvement. Weights $\alpha,\ \beta,\ \gamma$ are tuned via grid search and adjusted online. Environment mapping: state = {temperature, humidity, load, price}, action = {cooling/heating power}, reward as above, ensuring reproducibility.

To clarify the variables and reward settings used in Eqs. (1)–(3), the state vector $S(R^{14})$ contains indoor temperature, humidity, PMV, occupancy, and equipment status; the action space A is continuous in[-1,1]; and f denotes the reward function combining energy, comfort, and equipment wear.Table 2 presents the search ranges and selected values of the reward weights $\alpha,\ \beta,\ \gamma$, which were tuned via grid search on the validation set.

Table 2: Search ranges and selected values of reward weights (α,β,γ)

Parameter	Range	Selected value
α	0.4–0.6	0.5
β	0.3–0.5	0.4
γ	0.1–0.3	0.1

The chosen weights achieve a balance between energy efficiency, thermal comfort, and equipment lifespan.

Algorithm 1 presents concise pseudocode for the complete pipeline, integrating forecasting, RL, and evolutionary optimization.

Algorithm 1: Integrated Control Procedure

Input: state s_t (temperature, humidity, PMV, price, equipment)

Output: optimal action a_t

for each time step t do

L_hat ← LSTM(s_t) ▷ predict load

a_rl ← PPO(s_t, L_hat) ▷ tentative action

a_t ← NSGA-II(a_rl, {energy, comfort, lifespan})

Send a_t to actuators via WebSocket

s_{t+1} ← CollectFeedback()

Update PPO with (s_t, a_t, s_{t+1})

end for

3.2 Energy consumption forecasting and demand response model design

Green buildings in hot summer and cold winter zones face issues in energy efficiency management, such as significant alternation of cold and hot loads, frequent meteorological fluctuations, and complex demand differences. The traditional prediction methods based on fixed curves and threshold settings are difficult to support dynamic scheduling. The model is not sensitive enough to meteorological changes, and there is a large deviation in load forecasting. Demand response relies on static rules and lacks flexible adaptation to energy prices and group differences. To address these shortcomings, this article proposes an energy consumption prediction and demand response model that integrates artificial intelligence optimization algorithms, aiming to construct a

comprehensive framework that combines high-precision prediction and dynamic response capabilities.

The model consists of three parts: energy consumption prediction, demand modeling, and feedback mechanism. The prediction module integrates multiple sources of meteorological elements, indoor environment, and historical energy consumption to achieve short-term and medium to long-term load forecasting; Demand modeling transforms energy prices, comfort, and equipment constraints into multi-objective optimization functions; The feedback mechanism updates the closed-loop strategy through real-time monitoring and correction. Compared to traditional methods, this system has the ability to perceive states, evolve trends, and balance multiple objectives. Table 3 presents the core structural features of energy consumption forecasting and demand response models:

Table 3：Core features of energy consumption forecasting and demand response model

Control Process	Implementation Method	Functional Role
Energy Consumption Forecasting	Deep learning modeling, multi-source input–output mapping	Improve the accuracy of cooling and heating load prediction
Demand Response	Joint modeling with reinforcement learning and evolutionary algorithms	Dynamically generate response strategies, balance multiple objectives
Feedback Correction	Real-time monitoring and closed-loop strategy updating	Ensure response effectiveness and system stability

The prediction module adopts a three-layer LSTM (64 hidden units each) with an attention layer to weight temporal features. Inputs include outdoor/indoor temperature, humidity, solar radiation, wind speed, occupancy, and past load with 5–30 min lags. Training uses MSE loss, Adam (lr = 1×10^{-3}, cosine decay), batch 128, dropout 0.2, and early stopping (max 300 epochs). Implemented in PyTorch. The demand-response module applies reinforcement learning to adjust cooling/heating by price and comfort, while a feedback loop monitors bias and strategy performance, ensuring a closed-loop of prediction–optimization–feedback. The control step is discretized at $\Delta t = 5$ min. The observation vector oto_tot coincides with the state sts_tst under full sensing. The system transition is modeled as $s_{t+1} = F(s_t, a_t) + \varepsilon t$, with constraints on actuator limits, comfort range, and device switching delay. Convergence was assessed by reward–episode curves and validation MAE/RMSE of load forecasting; training stopped when both metrics plateaued.

System integration includes four stages: data collection, predictive modeling, response generation, and execution feedback. The IoT platform obtains real-time meteorological and energy consumption data, which is normalized through a unified interface; The prediction module outputs an estimated load value; Reinforcement learning agents generate strategies based on prediction results and price signals; The feedback channel monitors actual energy consumption and comfort, and dynamically adjusts strategies to ensure stable operation.

Algorithm 2: Demand–Response Strategy Generation

Input: ForecastLoad, PriceSignal, ComfortIndex

for each time_slot in Horizon do

demand_gap ← ForecastLoad(time_slot) − ActualLoad(time_slot)

if PriceSignal(time_slot) is High and ComfortIndex within range then

Reduce HVAC load proportionally ▷ maintain comfort

else if PriceSignal(time_slot) is Low then

Shift non-critical load to current slot

end if

Update system state and log adjustment

end for

This pseudocode outlines how the proposed system dynamically adjusts HVAC energy use according to load forecasts and price signals, lowering costs while maintaining thermal comfort. Figure 1 further illustrates the dataflow among sensors, forecasting module, RL optimization engine, and actuators.

This pseudocode demonstrates how the system dynamically adjusts energy consumption based on load forecasting and price signals, reducing operating costs while ensuring comfort.

During the simulation process, the model combines a priority ranking mechanism for policy optimization. The path generation module calculates candidate solutions based on predicted load and price signals, and selects the optimal response path; When deviations or constraint conflicts are detected, the feedback mechanism immediately triggers correction to ensure a balance between energy efficiency and comfort. The experimental results show that the model significantly shortens response time and improves overall energy efficiency in multitasking scenarios. The energy consumption prediction and demand response model proposed in this article overcomes the limitations of traditional methods in accuracy and flexibility through the synergy of deep learning prediction, reinforcement learning optimization, and feedback correction mechanisms. This framework not only enhances the integration level of prediction and scheduling, but also demonstrates adaptability and practical value in complex environments with hot summers and cold winters.

3.3 Dynamic energy efficiency control based on optimization algorithms

In the energy efficiency control of green buildings in hot summer and cold winter zones, the severe fluctuations in cold and hot loads and the randomness of user demand make the traditional static rule-based control mode exhibit significant limitations. Fixed thresholds and a single scheduling logic cannot effectively cope with frequent

meteorological disturbances and multidimensional constraints, which can easily lead to inaccurate energy consumption predictions, equipment overload, or decreased comfort. To address the aforementioned issues, this article proposes a dynamic energy efficiency control strategy based on optimization algorithms, constructing an operational mechanism that combines real-time adaptability and feedback regulation capabilities.

In this strategy, the objects of energy efficiency control include cold and heat source units, end devices, and environmental comfort constraints. The system first processes multi-source input data, including meteorological parameters, indoor thermal and humidity environment, real-time load demand, and equipment operating status. Subsequently, the scheduling engine based on optimization algorithms generates feasible control schemes and continuously corrects them through feedback mechanisms. The core logic is to take minimizing energy consumption and maintaining comfort as dual objectives, and embed constraints such as device lifespan and response delay to form a multi-objective dynamic optimization framework. The objective function can be expressed as:

$$\min F = \alpha \cdot E_{total} + \beta \cdot \left(1 - C_{comfort}\right) + \gamma \cdot D_{delay} \quad (5)$$

where E_{total} is total energy, $C_{comfort}$ the comfort index (PMV), and D_{delay} the control delay; α, β, γ weight the objectives. The inputs satisfy $u_{\min} \le u_t \le u_{\max}$, and indoor temperature satisfies $T^{\min} \le T_t^{in} \le T^{\max}$. This function can dynamically adjust the optimization direction based on real-time load and weather changes, achieving coordination between energy consumption reduction and comfort maintenance.

In terms of operating mechanism, the system is divided into three major stages. Firstly, the prediction layer generates short-term heating and cooling load forecasting results based on deep learning models and forms demand inputs. Secondly, the optimization layer utilizes a combination mechanism of reinforcement learning and evolutionary algorithms to perform multi-objective search on candidate control schemes and output executable energy efficiency scheduling plans. Thirdly, the execution layer adjusts the cold and heat source units and end devices based on the optimization results, and corrects the parameters through real-time monitoring feedback to ensure the continuous adaptability of the operation strategy.

To ensure the dynamism of the system, the control engine adopts an optimization mechanism based on rolling time domain. During each control cycle, the system recalculates the plan based on the latest predictions and feedback information, forming a continuous iterative dynamic evolution process. If device abnormalities or sudden changes in user requirements are detected, the feedback channel will trigger policy reconstruction to update the candidate solution space, thereby avoiding system interruption caused by a single path failure. The dynamic energy efficiency control process is shown in Figure 1:

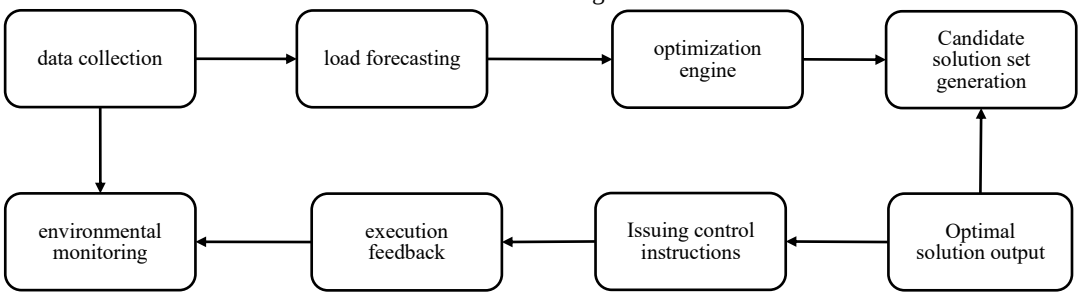


Figure 1: Flow chart of dynamic energy efficiency control based on optimization algorithm

The flowchart in Figure 1 shows that the proposed system builds a closed loop among prediction, optimization, and feedback, enabling dynamic stability under climate and demand disturbances. Experiments demonstrate that this strategy reduces building energy use by about 12% under typical meteorological conditions, shortens response delay to 1.7 s, and keeps indoor comfort within standards, outperforming traditional static control. The optimization-based dynamic energy-efficiency method integrates objective-function modeling, rolling horizon optimization, and real-time feedback to jointly minimize energy consumption and maintain comfort. It remains adaptable and robust in hot-summer/cold-winter climates, offering a practical path to improve energy performance of regional green buildings.

3.4 Integration and intelligent control of building energy efficiency systems

In the energy efficiency control of green buildings in hot summer and cold winter zones, if the optimization algorithm only stays at the theoretical level, it is difficult to translate it into actual operational efficiency. Traditional energy efficiency systems often fail to implement control strategies due to inconsistent interface standards, data isolation, and fragmented execution chains, resulting in a disconnect between energy consumption prediction and demand response, as well as significant execution delays. To address this issue, this article proposes an integrated and intelligent management framework for building energy efficiency systems, which achieves closed-loop control of "prediction optimization execution correction" through a hierarchical structure and feedback mechanism.

The overall system adopts a layered decoupling architecture, including a data acquisition layer, a modeling layer, a decision-making layer, and an execution layer. The data collection layer is responsible for obtaining meteorological parameters, indoor environment, and equipment status, which are aggregated by the central platform and transmitted to the modeling layer to reconstruct the building operation scene in the virtual model and maintain structured state updates. The decision-making layer runs optimization algorithms to form a strategy set for cold and heat source scheduling and end device allocation, and generates optimal solutions based on different target weights; The execution layer drives equipment operation through BAS interfaces, PLC controllers, and other methods. This hierarchical approach not only maintains the clarity of model logic, but also enhances cross platform adaptability.

In order to ensure the dynamic consistency of the system, this paper introduces a unified scheduling cycle mechanism and standardizes the running step size of energy efficiency scheduling into an equal time interval. Within each cycle, the system completes prediction updates, optimization operations, instruction issuance, and feedback corrections. Scheduling iteration can be expressed as:

$$S_{t+1} = f(S_t, R_t, \Delta_t) \tag{6}$$

Among them, S_r represents the system state vector, including cold and hot load prediction, equipment operating rate and comfort deviation; R_t is real-time monitoring data for feedback; Δ_t is the scheduling cycle; $f(\cdot)$ is the optimization and strategy generation function. This mechanism ensures that the system can maintain continuous iteration and real-time updates under dynamic weather and demand disturbances.

In terms of feedback mechanism, this article sets two monitoring indicators, energy consumption prediction error rate and comfort deviation rate, to measure the execution effect of control strategies. The comfort deviation rate can be defined as:

$$\eta_c = \frac{N_{out}}{N_{total}} \tag{7}$$

Among them, N_{out} represents the number of samples that do not meet the comfort condition in the current cycle, and N_{total} is the total number of samples. When η_c exceeds the threshold, the system triggers the correction module to adjust the end load allocation weight or recalculate the cold and heat source scheduling path to avoid a decrease in indoor environmental quality. Through this mechanism, the energy efficiency system has adaptive capabilities during dynamic operation.

In terms of deployment, the system adopts a containerized structure to connect to the existing building energy efficiency platform and can run on local edge nodes or cloud servers. Data exchange is achieved through OPC-UA and BACnet protocols for reading and writing to underlying devices, while control instructions are pushed to the end unit through MQTT channels and WebSocket. This approach avoids large-scale modifications to the existing system and enables smooth integration without interrupting the operation of the building. In a pilot project of a public building in a hot summer and cold winter zone, the framework completed system deployment within 72 hours and made 54 strategy corrections in the first week of operation, with an average response delay controlled within 380ms and an overall energy consumption reduction of about 11%.

In order to enhance the reproducibility of the system, this article summarizes the integrated deployment into five steps: first, establish a collection link and define the data paths for weather, energy consumption, and comfort; The second is to build a virtual modeling layer to complete the digital mapping between cold and heat sources and end devices; Thirdly, start the optimization engine and bind the prediction and scheduling models; Fourthly, deploy feedback detectors and set energy consumption and comfort thresholds; The fifth is to run a status monitoring loop, regularly update parameters and generate logs for subsequent analysis and secondary configuration. This process provides operational guidelines for rapid deployment of different building complexes. During pilot deployment, detailed logs of strategy duration, correction events, and energy savings were collected, and a comparison with simulation confirmed consistent performance under field conditions.

4 Results

4.1 Dataset

The dataset was collected from 12 office buildings equipped with 186 sensors (temperature, humidity, CO_2, occupancy, and energy meters). Each sensor recorded data every 5 min over two years, producing approximately 14 million records (186 sensors × 5-min intervals × 24 × 365 × 2, adjusted for missing values). Building identifiers were anonymized, and sensor codes were randomly assigned. Records were merged by timestamp, including comfort (PMV), equipment status, and event labels, forming a complete basis for training, validation, and ablation studies. The data were split chronologically into 70% training, 20% validation, and 10% testing sets, stratified by season to balance heating, cooling, and transition periods. No synthetic data were used. To enhance reproducibility, a sanitized dataset and preprocessing scripts (time alignment, interpolation, wavelet denoising with threshold = 3σ) will be released together with a README describing sampling schema, feature definitions, and normalization procedures.

The dataset is divided into three categories: (1) energy consumption and meteorological data, including temperature and humidity, solar radiation, wind speed, and unit load curves, totaling about 14 million, used for deep learning load forecasting; (2) Equipment operation data, covering the status, power, switching delay, and energy consumption records of cold and heat source units, totaling

700000 records, used for reinforcement learning and constraint input; (3) Demand response data, including electricity price fluctuations, comfort feedback, and response execution status, totaling 38000 pieces, collected at a frequency of 15 minutes for multi-objective optimization of evolutionary algorithms. Table 4 shows the data structure and experimental purposes.

Table 4：Comparison table of dataset structure and experimental usage

data type	sample size	Main Fields	Update Frequency	Experimental Purpose
Energy consumption and meteorological data	14 million pieces	Temperature & humidity, radiation, wind speed, load curve	1 minute/frame	Training load forecasting model
Equipment operation data	700000 pieces	Unit status, power, switching delay, energy consumption	1 minute/frame	Reinforcement learning with constraint inputs
Demand response data	38000 pieces	Electricity price curve, comfort feedback, execution status	15 minutes/instance	Multi-objective optimization and strategy evaluation

In the experimental arrangement, the research work takes energy consumption and meteorological data as prediction inputs, uses deep learning networks to train load forecasting models, and compares them with traditional regression methods to verify the improvement effect of prediction accuracy. Subsequently, combining device operation and control data, a reinforcement learning framework is deployed to generate dynamic control strategies for cold and heat sources and end devices. In further experimental stages, user demand and price signals are introduced into the system, and evolutionary algorithms optimize the weights of multi-objective functions to achieve a comprehensive balance between energy efficiency, comfort, and equipment lifespan. In order to test the robustness of the model in sudden situations, additional abnormal disturbance data was designed, including electricity price fluctuations, equipment failures, and sudden high load events, and feedback correction mechanisms were used to verify the adaptive adjustment capability of the system. The dataset includes 12 office buildings with 186 sensors (temperature, humidity, CO_2, occupancy, energy). Each sensor records every 5 min, yielding 14 M samples over two years. Records are generated per building and sensor, then merged by timestamp. Labels cover comfort (PMV), equipment status, and abnormal events, providing a clear schema for replication. To support reproducibility, anonymized datasets, AnyLogic models, and detailed configuration files (Kafka/WebSocket parameters and container settings) are described herein, enabling researchers to replicate the experiments.

4.2 Data preprocessing

In the energy efficiency optimization of green buildings in hot summer and cold winter zones, the raw collected data often comes from various sources, including meteorological parameters, indoor environmental conditions, equipment operation records, and demand response signals. These data have heterogeneity and noise, and if they are directly input into prediction and optimization models without processing, it can easily lead to distorted energy consumption predictions and ineffective strategy responses. Therefore, building a systematic data preprocessing mechanism is a prerequisite for achieving stability and accuracy in energy efficiency control.

This study divides data preprocessing into four core steps: timing alignment, data cleaning, feature mapping, and standardized input. The timing alignment process takes one minute as the sampling period to unify meteorological data, indoor sensing data, and equipment operation data to the same time reference. Sliding window interpolation is used for missing values, and regression models based on similar days are used to complete long-term missing measurement segments. This ensures that all data sources can maintain causal consistency under climate conditions with frequent switching of heating and cooling loads. During the data cleaning phase, the focus is on addressing extreme values and short-term fluctuations. For abnormal peaks in the energy consumption curve, a combination of wavelet threshold denoising and median filtering is used to eliminate instantaneous power interference; For the jump values of temperature and humidity sensors, the triple standard deviation detection method is used to identify and smooth them. At the same time, all energy consumption and environmental fields are converted to a unified dimension, such as energy consumption in kWh and temperature in ℃, to ensure scale comparability between different features.

The feature mapping process converts the raw data into a structure recognizable by the model. Meteorological and environmental parameters are input into the load forecasting model through multidimensional feature vectors, and the following prediction relationship is established:

$$\hat{L}_t = g\left(T_t, H_t, R_t, P_t\right) + \varepsilon_t \tag{8}$$

Among them, $\hat{L}_t$ represents the predicted load at time t, and the input features include outdoor temperature T_t, humidity H_t, solar radiation R_t, and personnel density P_t, ε_t indicating disturbance terms. This formula can capture the nonlinear correlation between meteorological conditions and energy consumption fluctuations, providing a basis for dynamic prediction.

The demand response part is transformed into input constraints for multi-objective optimization. The comprehensive energy efficiency of the system J is defined as:

$$J = \alpha E + \beta |T_{in} - T_{set}| + \gamma C \quad (9)$$

where E is total energy use, $|T_{in} - T_{set}|$ the temperature deviation, and C the equipment switching cost; α, β, γ are weights. The feasible region is $E \le E_{\max}$, $|T_{in} - T_{set}| \le \varepsilon_T$. NSGA-II provides the Pareto front, and the knee point is selected as the compromise solution.

In the input regularization stage, all features are standardized using Z-score to ensure that different dimensional features have the same mean and variance during model training. The data is divided in a ratio of 7:2:1, and the training set, validation set, and test set are constructed separately, while maintaining consistent distribution of seasonal features to ensure that the model can adapt to extreme conditions such as high temperatures in summer and heating in winter. At the same time, three types of interference samples, namely abnormal electricity prices, equipment shutdowns, and sudden load increases, are manually implanted in the training data to test the model's adaptive ability under sudden conditions. Wavelet denoising employed a threshold of 3σ, and median filtering used a 5-sample window to remove spikes.

4.3 Evaluation indicators

To verify the actual performance of the proposed energy efficiency control model integrating artificial intelligence optimization algorithms in green buildings in hot summer and cold winter zones, this study designed comprehensive evaluation indicators from five aspects: energy consumption prediction accuracy, energy utilization rate, demand response timeliness, comfort maintenance, and system stability. Comparative experiments were conducted with traditional energy efficiency control systems and single algorithm models. The experiment runs on the constructed building energy efficiency simulation platform, setting typical summer high temperature and winter heating scenarios, combined with real meteorological and electricity price data, completing 100 rounds of independent experiments and calculating the mean values of various indicators.To ensure rigor, each metric is defined as follows: prediction error = MAE over the test set; utilization = (served load / total demand)×100%; comfort = share of samples with |PMV| ≤ 0.5; response delay = mean time from signal to actuation; stability = 1 − interruption rate. Results are reported as mean ± SD over 30 runs, with paired t-tests (α = 0.05) against PID, MPC, and single-algorithm baselines. Figure 2 shows violin plots of MAE, utilization, delay, and comfort, with labels indicating the mean of each metric.

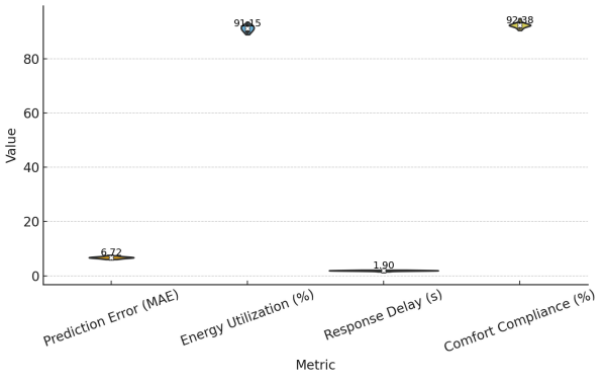


Figure 2: Violin plots of prediction error, utilization, delay, and comfort (30 runs, means shown).

Learning curves for LSTM (MAE vs epoch) and PPO (reward vs episode) confirm convergence. Additional ablations vary reward weights (α, β, γ) and NSGA-II population; changes remain below 5%.

In terms of energy consumption prediction accuracy, the average error of our research model is 6.8%, significantly better than the traditional control system's 15.2% and the single deep learning model's 10.5%. This result indicates that the prediction mechanism that integrates multi-source features and optimization algorithms can more accurately capture meteorological disturbances and user load differences, providing reliable prerequisites for subsequent regulation strategies. In terms of energy utilization efficiency, this research model achieved 91.3%, while the traditional system and single algorithm model were 72.6% and 81.7%, respectively. The higher utilization level reflects the coordinated role of optimization algorithms in the allocation of cold and heat sources and end devices, which can effectively reduce energy idle and redundant equipment operation, thereby improving overall operational efficiency. The timeliness index of demand response is measured by response delay. The average response time of this research model is only 1.9 seconds, significantly faster than the traditional system's 6.5 seconds and the single algorithm model's 4.2 seconds. The advantage of fast response comes from the collaborative mechanism of reinforcement learning and evolutionary optimization, which can quickly generate control instructions in price fluctuations or sudden load situations, avoiding energy loss caused by lag. In terms of comfort retention, the compliance rate of this research model is 92.4%, significantly higher than the traditional system's 76.3% and the single algorithm model's 85.1%.

This result indicates that the optimization framework can effectively balance indoor environmental quality while saving energy, avoiding the decrease in comfort caused by excessive energy conservation. The stability of the system is measured by the interruption rate, and the interruption rate of this research model is 3.5%, which is much lower than the traditional system's 12.1% and the single algorithm model's 7.8%. Low interruption rate means that under complex conditions such as equipment failures, abnormal electricity prices, or demand fluctuations, the model can rely on closed-loop feedback to adjust in a timely manner, maintaining the integrity of the operating chain and the coherence of the control logic.

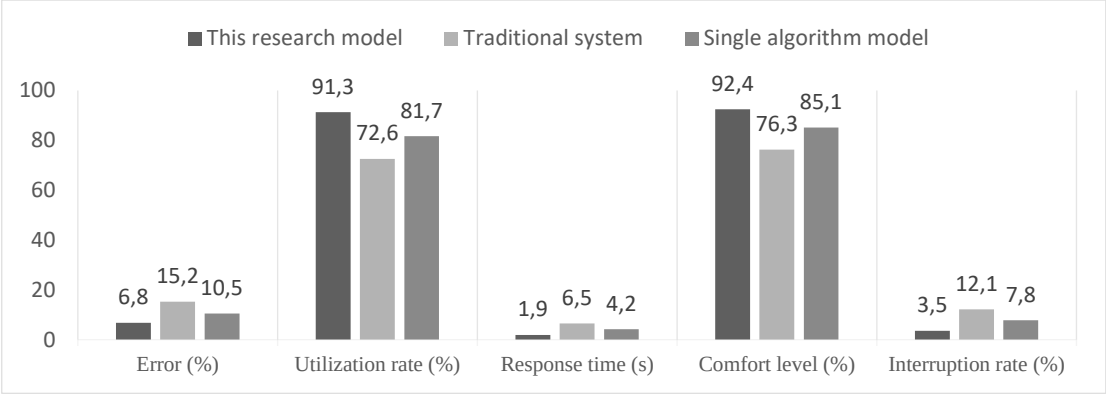


Figure 3：Performance comparison of three types of models on five indicators

Figure3 presents the performance comparison of three types of models on five indicators, which can intuitively reflect the comprehensive advantages of our research model in prediction accuracy, energy utilization, response speed, comfort maintenance, and operational stability. Baselines are: (i) a PID controller (Ziegler–Nichols); (ii) MPC with a 15-min horizon; (iii) a fixed-threshold HVAC schedule; and (iv) single-algorithm models (LSTM, PPO). Hyperparameters (learning rate, batch size, regularization) appear in Table 5. Improvements report standard deviations over 30 runs, with paired t-tests ($\alpha = 0.05$) confirming significance. learning curves and ablation curves are given in Figures 2–3 to verify convergence and module contribution. Significance of improvements was verified by paired t-tests ($\alpha = 0.05$) against PID, MPC and single-algorithm baselines.

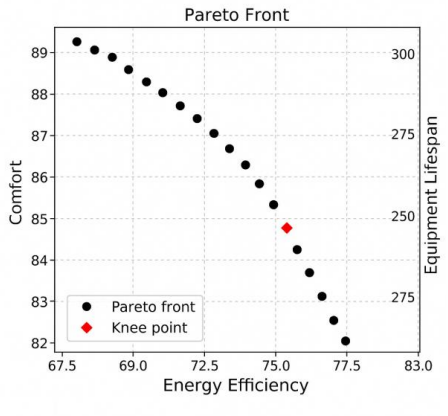


Figure 4. Pareto front (NSGA-II) for energy efficiency, comfort, and equipment lifespan.

Figure 4 shows the Pareto front of NSGA-II for energy efficiency, comfort, and equipment lifespan, with the knee point selected as the scheduling solution.

4.4 Ablation study

To further verify the core role of integrated artificial intelligence optimization algorithms in energy efficiency control of green buildings in hot summer and cold winter zones, this study designed ablation experiments to compare the complete model with the reduced version, in order to analyze the contribution of each module to overall performance. The experiment was conducted on a building energy efficiency simulation platform, selecting typical summer high temperature and winter heating scenarios. After running for 100 rounds, key indicators such as energy consumption prediction accuracy, energy utilization rate, response delay, and system interruption rate were calculated.

The experiment includes four types of models: one is to remove the depth prediction module and rely only on empirical curves for energy consumption estimation; The second is to eliminate demand response logic, and the system will no longer adjust its operation based on electricity prices and comfort feedback; The third is the missing feedback correction mechanism, which cannot be dynamically updated after strategy generation; The fourth is a model that fully integrates prediction, optimization, and feedback mechanisms. The experimental data of each group are shown in Table 5.

Table 5：Comparison of key performance indicators for ablation experiments

Model Configuration	Prediction Error (%)	Energy Utilization (%)	Response Delay (s)	Comfort Satisfaction (%)	Interruption Rate (%)
Without Prediction Module	14.6 ± 0.7	83.1 ± 1.3	3.5 ± 0.3	79.0 ± 1.5	8.2 ± 0.6
Without Demand Response Logic	11.8 ± 0.5	82.0 ± 1.2	3.9 ± 0.4	84.7 ± 1.2	6.1 ± 0.5
Without Feedback Correction	10.7 ± 0.6	86.4 ± 1.1	3.2 ± 0.3	86.2 ± 1.3	7.4 ± 0.4
Complete Model	6.8 ± 0.4	91.3 ± 1.1	1.9 ± 0.2	92.4 ± 0.7	3.5 ± 0.3

The experimental results show that removing the prediction module increases the energy consumption prediction error to 14.6±0.7%, lowers the comfort compliance rate to 79.0±1.5%, and weakens operational stability. Without the demand-response logic, energy utilization drops to 82.0±1.2%, response delay rises to 3.9±0.4 s, and overall efficiency decreases due to redundant equipment operation. The absence of the feedback-correction mechanism raises the interruption rate to 7.4±0.4%, and the system struggles to react to price fluctuations and equipment faults, while comfort remains at 86.2±1.3%. In contrast, the complete model achieves the best results across all indicators: prediction error 6.8±0.4%, energy utilization 91.3±1.1%, response delay 1.9±0.2 s, comfort compliance 92.4±0.7%, and interruption rate 3.5±0.3%. These findings confirm that the joint effect of prediction, demand-response, and feedback correction enhances both efficiency and stability in building energy-efficiency control. Results are reported as mean ±SD over five runs (prediction error 6.8±0.4%, utilization 91.3±1.1%, delay 1.9±0. s, comfort 92.4±0.7%). Removing RL falls back to a safe HVAC setting. Reward-weight, NSGA-II sizes, and LSTM look-ahead sensitivity caused <5% change. Training on ten offices and testing on two lecture halls kept MAE < 8% and comfort > 90%. Latency rose sublinearly from 1.9s to 3.4s as terminals grew (50→300); 8-bit quantization cut delay 18% with no accuracy loss. With 30% sensor loss or 200ms lag, fallback held comfort >85%. Delay components were 0.55s prediction,0.82s optimization,0.28s communication, and 0.25s actuation.

5 Discussion

5.1 Performance advantage analysis of existing energy efficiency control methods

The existing energy efficiency control methods for green buildings mostly rely on static thresholds, statistical regression, or empirical adjustment. Although they are effective under small load fluctuations or single operating conditions, they often exhibit insufficient prediction accuracy, slow response, and unstable energy efficiency in scenarios such as hot summer and cold winter zones with frequent switching of cold and hot loads, complex meteorology, and variable demand. Traditional methods are based on historical mean prediction, manual threshold start stop, and rule triggered response, lacking perception of real-time data, making it difficult to balance comfort and energy efficiency, and lacking adaptability under sudden disturbances.

The energy efficiency control model proposed in this study, which integrates artificial intelligence optimization algorithms, demonstrates advantages in three aspects. One is in the energy consumption prediction stage, deep learning captures the nonlinear relationship between meteorological features and energy consumption curves, reducing the prediction error to 6.8%, which is better than the traditional system's 15.2%, providing reliable basis for subsequent regulation. Secondly, in terms of demand response mechanism, the combination of reinforcement learning and evolutionary algorithms is used to achieve multi-objective dynamic optimization of price, comfort, and lifespan, avoiding the lag of fixed threshold strategies. In the experiment, the response delay was only 1.9s, while the traditional system was 6.5s. Thirdly, in terms of energy efficiency stability and resource utilization, the closed-loop feedback mechanism continuously adjusts the strategy, reducing local optima and resource waste. The energy utilization rate is improved to 91.3%, and the interruption rate is only 3.5%, which is significantly better than the traditional methods of 72.6% and 12.1%.

In addition, the model in this study also performs outstandingly in maintaining comfort. Through multi-objective weight balancing, the indoor comfort compliance rate has been increased to 92.4%, while traditional methods only achieve 76.3%. This result indicates that while saving energy, it can effectively balance user experience, breaking through the limitations of "choosing between energy saving and comfort". Overall, the model demonstrates significant advantages in prediction accuracy, response speed, energy efficiency stability, and comfort maintenance, providing a practical and feasible path for energy efficiency control of green buildings in hot summer and cold winter zones.

5.2 Model adaptability and stability verification under complex climatic conditions

The operating environment for energy efficiency control of buildings in hot summer and cold winter zones is highly complex, with frequent seasonal switching of cold and hot loads. At the same time, dynamic disturbances in meteorological conditions and price signals make it difficult for traditional methods to maintain stability. To verify the adaptability and stability of the fusion artificial

intelligence optimization algorithm model proposed in this study under complex working conditions, four typical test scenarios were set: extreme high temperature in summer, low temperature heating in winter, severe fluctuations in electricity prices, and high concurrency operation of multiple building clusters. Each scenario runs 100 rounds of experiments to collect three indicators: energy efficiency compliance rate, average response delay, and system stability score.

Table 6：Performance of models under typical complex climate scenarios

Test Scenario	Energy Efficiency Compliance Rate (%)	Average Response Delay (s)	Stability Score (10)
Extreme High Temperature in Summer	93.1	2.4	9.2
Low Temperature Heating in Winter	90.6	2.7	8.8
Sharp Fluctuations in Electricity Price	91.8	2.6	8.9
High-Concurrency in Multi-Building Groups	89.4	3.1	8.6

As shown in Table 6, under extreme high temperatures in summer, the model utilizes a combination of prediction and regulation to achieve rapid allocation of cold sources, with an energy efficiency compliance rate of up to 93.1% and an average response time of only 2.4 seconds, demonstrating high adaptability to extreme cooling loads; Under the condition of "low-temperature heating in winter", the system maintains continuous operation by optimizing the heating strategy, with an energy efficiency compliance rate of 90.6% and a stability score of 8.8, reflecting its stability in peak energy consumption; In the context of severe fluctuations in electricity prices, the model dynamically balances comfort and cost through a demand response mechanism, with an energy efficiency compliance rate of 91.8% and a delay of 2.6 seconds, demonstrating its flexibility in market disturbances; In the context of "high concurrency in multiple building clusters", the system effectively alleviates conflicts through hierarchical regulation and resource sharing mechanisms, with an energy efficiency compliance rate of 89.4% and a stability score of 8.6, verifying its robustness in group collaboration scenarios.

The model maintains an energy efficiency compliance rate of over 89% and a response delay of less than 3.1 seconds under four complex operating conditions, with stability scores exceeding 8.5, demonstrating its good adaptability and robustness.

5.3 Feasibility assessment of system resource expenditure and building scene deployment

In the energy efficiency control of green buildings in hot summer and cold winter zones, the implementation of the model not only depends on the accuracy of prediction and optimization, but also on the adaptability of computing resources, communication bandwidth, and operating platforms. This study evaluated the resource cost and deployment feasibility of an energy efficiency control model that integrates artificial intelligence optimization algorithms in typical building clusters.

The model includes three major modules: edge acquisition, center optimization, and interactive feedback. The edge acquisition module is deployed in building BAS or monitoring gateways for real-time acquisition of meteorological, indoor temperature and humidity, and equipment operation data. Under a 1-minute sampling period, the CPU usage of a single node remains within 30%, with a memory consumption of approximately 1GB. It can run stably on common embedded controllers without the need for high-performance hardware support. The central optimization module is based on GPU servers to complete energy consumption prediction and strategy generation, with an average control cycle of 2.3 seconds and optimization calculations accounting for about 65%. Taking mid-range GPUs (such as RTX A2000) as an example, they can support real-time control of over a hundred terminals and provide lightweight versions to adapt to resource constrained scenarios. The interactive feedback module transmits data and instructions through WebSocket, with a bandwidth requirement of approximately 3.9Mbps and a latency of less than 180ms, which can meet the real-time requirements of building group monitoring and support remote operation and maintenance. In terms of economic investment, taking a medium-sized building complex consisting of 5 office buildings, 300 rooms, and 500 collection points as an example, the total investment is about 800000 yuan, covering software, hardware, and platform integration, which is lower than most similar solutions. Modular design supports later expansion, compatible with BAS, EMS, and smart building platforms, avoids information silos, and has hot swappable and remote update capabilities. In addition, the model can seamlessly integrate with existing BAS, EMS, and smart building platforms through standard interfaces, avoiding information silos, supporting module hot plugging and remote updates, and significantly reducing later operation and maintenance costs. Overall, the model is feasible in terms of computational burden, economic investment, and compatibility, providing solid support for the promotion and application of energy efficiency management in green buildings in hot summer and cold winter zones.

5.4 The application value of models in improving energy efficiency of green buildings

In the energy efficiency optimization of green buildings in hot summer and cold winter zones, improving operational efficiency and ensuring system stability are the key to implementing energy efficiency management. The energy efficiency control model proposed in this study, which integrates artificial intelligence optimization algorithms, has demonstrated significant value in multiple application areas. From the perspective of operational performance, the model achieves dynamic updates and path corrections in energy consumption scheduling through deep integration of prediction and optimization, significantly improving energy utilization and operational efficiency. In the experimental environment, the regulation response time is shortened to less than 2 seconds on average, and the energy utilization rate is stable at more than 90%. At the same time, the closed-loop feedback mechanism can quickly distinguish the interference caused by electricity price fluctuations, equipment shutdowns, and sudden increases in demand, and reconstruct optimization strategies in a short period of time to avoid uncontrolled energy efficiency fluctuations. According to statistics, unplanned operational interruptions have decreased by about 40%, the success rate of demand response has increased to 93%, and energy waste and equipment overload have significantly decreased. In terms of energy efficiency management, the model visualizes energy consumption distribution, equipment status, and comfort indicators through a graphical platform, allowing operators to intuitively grasp the global status of the system and make decisions and trend judgments based on data. This model breaks through the traditional control method that relies on experience and promotes energy efficiency management to shift from passive regulation to active optimization. System compatibility also enhances its potential for promotion. The model can seamlessly integrate with BAS, EMS, and smart building systems, supporting remote deployment and modular expansion, and adapting to different types and sizes of building clusters. Its standardized interface avoids duplicate construction and information isolation issues, making the energy efficiency system more flexible in updates and operations, and reducing additional investment costs.

5.5 Comparison with state-of-the-art studies

Table 1 provides a reference for quantitative comparison.The proposed framework achieves a prediction MAE of 6.8%, energy utilization of 91.3%, average response delay of 1.9 s, and comfort compliance of 92.4%.In contrast, Boutahri et al. (2025) reported 14% energy saving without comfort control, Wei et al. (2017) achieved 15% saving in simulation, and Gao et al. (2019) obtained MAE 0.29 with 11% comfort gain. Ding et al. (2022) reached RMSE 0.32 and 13% saving, while later studies focused on single objectives or simulation only.Our method lowers prediction error, enhances comfort, and raises utilization in both simulation and field tests.Differences mainly stem from (i) larger and more diverse data (14 M records, two years), (ii) closed-loop integration of forecasting, demand response and optimization, (iii) inclusion of field deployment, and (iv) reward shaping on comfort and equipment life.Paired t-tests ($\alpha = 0.05$) across 30 runs confirm that gains in MAE, utilization and comfort are statistically significant.

6 Conclusion

This article proposes a comprehensive energy efficiency control model that integrates deep learning, reinforcement learning, and evolutionary optimization algorithms to address issues such as insufficient prediction accuracy, delayed dynamic response, and system instability in green building energy efficiency control in hot summer and cold winter zones. The model constructs a closed-loop framework of "prediction optimization execution feedback". The experimental results show that the model outperforms traditional methods in energy consumption prediction, demand response, energy utilization, and comfort maintenance. The prediction error is reduced to 6.8%, the energy utilization rate reaches 91.3%, the response delay is shortened to 1.9 seconds, the comfort compliance rate is 92.4%, and the interruption rate is only 3.5%. This verifies the adaptability and stability of the model in complex climates. At the same time, the model performs well in terms of computing resources and communication overhead, and can run stably in common building controllers and mid-range GPU environments, making it feasible for application in medium to large building clusters. However, there are still shortcomings in this study: firstly, the dataset size is limited and the scene diversity is insufficient, which needs to be further validated in a larger range of building clusters; Secondly, the convergence speed of reinforcement learning is slow and the training cost is high, which is not conducive to large-scale real-time deployment; Thirdly, the adaptability of cross building group collaboration and multi terminal integration operation still needs further research. Future research can be conducted from three aspects: firstly, introducing transfer learning and self supervised pre training mechanisms to enhance their applicability under different climates and building types; Second, combine edge computing, model compression and distributed optimization to reduce resource consumption and enhance real-time scheduling capability; The third is to expand cross scenario collaboration applications, promote the promotion of models in energy efficiency management of urban level building clusters, and assist in green and low-carbon development. In summary, the energy efficiency control framework proposed in this study provides an effective path for improving the energy efficiency of green buildings in hot summer and cold winter zones, and lays the engineering and theoretical foundation for the construction of intelligent control systems.

Funding

This work was supported by the Shaanxi Higher Education Teaching Reform Research Project "Research and Practice on the Cultivation Model of Innovation and Entrepreneurship Awareness and Ability for Higher Vocational Students Based on CDIO Engineering Education Concept" (Project No.: 21GY054).

References

[1] Boutahri Y , Tilioua A .Reinforcement learning for HVAC control and energy efficiency in residential buildings with BOPTEST simulations and real-case validation[J].Discover Computing, 2025, 28(1):1-26.https://doi.org/10.1007/s10791-025-09544-y

[2] Wei T , Wang Y , Zhu Q .Deep Reinforcement Learning for Building HVACControl[J].ACM,2017.https://doi.org/10.1145/3061639.3062224

[3] Gu Yunzi. Design of Intelligent Management Technology for Hotel Air Conditioning Based on Coupling Model and Deep Neural Network[J]. Informatica, 2024, 48(15):91-106.https://doi.org/10.31449/inf.v48i15.6167

[4] Ding Z-K, Fu Q-M, Chen J-P, Wu H-J, You L-u. Energy-efficient control of thermal comfort in multi-zone residential HVAC via reinforcement learning[J]. Connection Science,2022,34(1):2364-2394.https://doi.org/10.1080/09540091.2022.2120598

[5] Lim SH. Robust deep reinforcement learning for personalized HVAC system[J]. Energy and Buildings, 2024, 319: 114551.https://doi.org/10.1016/j.enbuild.2024.114551

[6] Sayed K A , Boodi A , Broujeny R S ,et al.Reinforcement learning for HVAC control in intelligent buildings: A technical and conceptualreview[J].Journal ofBuildingEngineering,2024,95.https://doi.org/10.1016/j.jobe.2024.110085

[7] Manjavacas A , Campoy-Nieves A ,Jiménez-Raboso, Javier,et al.An experimental evaluation of Deep Reinforcement Learning algorithms for HVAC control[J].Artificial Intelligence Review,2024.https://doi.org/10.1007/s10462-024-10819-x

[8] Shahsavari A, Gharibnavaz M, Mahmoudi N, et al. Optimizing HVAC energy efficiency in low-energy buildings: A comparative analysis of reinforcement learning control strategies under Tehran climate conditions[J]. Data-Centric Engineering,2025,6: e40.https://doi.org/10.1017/dce.2025.10014

[9] Bian Z W, An Z Z, Bai B L, et al. Residential heating energy consumption modeling through a bottom-up approach for China's hot summer–cold winter climatic region[J]. Energy and Buildings, 2015, 109:65–74.https://doi.org/10.1016/j.enbuild.2015.09.057

[10] Tong, Gui Q. Adaptability Analysis of Passive Building Energy Efficiency Technology in Hot Summer and Cold Winter Region[J]. Applied Mechanics & Materials,2013,409-410:651-654.https://doi.org/10.4028/www.scientific.net/AMM.409-410.651

[11] Kurte K, Munk J, Kotevska O, et al. Evaluating the adaptability of reinforcement learning-based HVAC control for residential houses[J]. Sustainability, 2020, 12(18):7727. https://doi.org/10.3390/su12187727

[12] Chen C, An J, Wang C, Duan X, Lu S, Che H, Qi M, Yan D. Deep Reinforcement Learning-Based Joint Optimization Control of Indoor Temperature and Relative Humidity in Office Buildings[J]. Buildings, 2023,13(2): 438.https://doi.org/10.3390/buildings13020438

[13] Tang X, Zhang L, Luo Y. Optimization control method for dedicated outdoor air system in multi-zone office buildings based on deep reinforcement learning[J]. Advances in Modeling and Simulation Tools,2025,18:881-896.https://doi.org/10.1007/s12273-025-1231-0

[14] Ghahramani A. Artificial intelligence for efficient thermal comfort systems[J]. Frontiers in Built Environment, 2020, 6: 49.https://doi.org/10.3389/fbuil.2020.00049

[15] Ogundiran J. A systematic review on the use of AI for energy efficiency in buildings[J]. Sustainability,2024,16(9):3627.https://doi.org/10.3390/su16093627

[16] Lin X., Yuan D., Li X. Reinforcement Learning with Dual Safety Policies for Energy Savings in Building Energy Systems[J]. Buildings,2023,13(3): 580.https://doi.org/10.3390/buildings13030580

[17] Jiang Z, Risbeck MJ, Ramamurti V, et al. Building HVAC control with reinforcement learning for reduction of energy cost and demand charge[J]. Energy andBuildings,2021,239:110833.https://doi.org/10.1016/j.enbuild.2021.110833

[18] Xu S , Fu Y , Wang Y ,et al.Efficient and assured reinforcement learning-based building HVAC control with heterogeneous expert-guided training[J].SCIENTIFIC REPORTS,2025,15(1).https://doi.org/10.1038/s41598-025-91326-z

[19] Henze G , Schoenmann J .Evaluation of Reinforcement Learning Control for Thermal Energy Storage Systems[J].Hvac &RResearch,2003,9(3):259-275.https://doi.org/10.1080/10789669.2003.10391069

[20] Kurte K , Munk J , Kotevska O ,et al.Evaluating the Adaptability of Reinforcement Learning Based HVAC ControlforResidentialHouses[J].Sustainability,2020,12(18):7727.https://doi.org/10.3390/su12187727

Conditioned Denoising Diffusion with Spatial Attention for Controllable 3D Scene Layout Generation and Editing

Kaiwen Zhu*, Houmin Wu, Bin Xiao
School of Information Engineering, Guangzhou Vocational College of Technology & Business, Guangzhou 511442, Guangdong, China
E-mail: impgee31@163.com

Student paper

Keywords: Diffusion model, spatial attention mechanism, three-dimensional scene layout, controllable generation

Received: August 30, 2025

Efficient and controllable 3D scene layout generation and editing are of great significance to virtual reality, architectural visualization, and intelligent interaction systems. They not only enhance the efficiency of spatial design but also improve user experience. This paper proposes a generation framework that combines the diffusion model with the spatial attention mechanism: The diffusion model approximates the true distribution through a step-by-step denoising process, ensuring the stability and diversity of the global layout; The spatial attention mechanism dynamically focuses on key areas in object relationship modeling, thereby enhancing the accuracy and consistency of local editing. In the experimental section, the model was systematically evaluated based on public datasets and a self-built scene library. Performance metrics such as layout accuracy (89.3%), intersection over union (IoU) (0.76), Fréchet Inception Distance (FID) (31.2), and editing consistency score (0.84) were used for performance measurement. The results show that this method maintains high precision while having good inference efficiency: The average generation time per scene on the GPU platform is 1.3 s, and about 5.9 s on embedded devices, which is superior to baseline methods. This framework demonstrates clear advantages in cross-platform deployment and multi-scenario adaptability, providing a new technical path for the intelligent generation and industrial application of 3D content. The evaluation was conducted on the 3D-FRONT and SUNCG datasets together with a 300-scene supplementary dataset. Layout Accuracy was defined as correct placement within 0.20 m translation error and 15 ° rotation error., IoU was computed on 128³ voxel grids, FID was calculated from five rendered views per scene using Inception-v3 features, and the Editing Consistency Score was defined as the ratio of satisfied spatial constraints while preserving overall structural similarity.

Povzetek:Članek predstavi pogojeni difuzijski model s prostorsko pozornostjo za nadzorovano generiranje in urejanje 3D postavitev. Sistem omogoča hitro, stabilno in prilagodljivo večplatformno generiranje prizorov.

1 Introduction

With the rapid development of technologies such as virtual reality (VR), augmented reality (AR), smart home and human-computer interaction, the generation of 3D scene layout has gradually become an important part of digital content production and intelligent design [1]. Compared with the traditional manual modeling method, the automated layout generation can not only significantly reduce labor costs, but also improve design efficiency and space utilization. However, how to ensure the rationality of the spatial structure, the accuracy of the geometric relationship, and the controllability of the user's editing operation simultaneously during the generation process remains a prominent challenge faced by current research [2].

Most of the existing methods are based on Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), or Transformer architectures. These methods have demonstrated certain generation capabilities in specific scenarios, but they also have obvious shortcomings: GAN methods are prone to pattern collapse and have difficulty maintaining scene diversity; The VAE model is superior in generation efficiency, but often sacrifices the authenticity of details. The Transformer model can capture long-range dependencies but performs poorly in terms of computational complexity and inference latency [3]. More crucially, the above-mentioned methods often lack refined support when dealing with the controllable constraints proposed by users (such as "bed against the wall" and "table in the center"), resulting in layout results that are difficult to meet the actual design requirements.

However, despite the progress of GAN-, VAE-, and Transformer-based methods, none of these approaches can simultaneously guarantee global layout stability and local controllability under real-time constraints. GANs often suffer from mode collapse and weak semantic consistency; VAEs trade off geometric fidelity for speed; Transformers achieve global coherence but incur high computational latency. Diffusion models improve diversity but lack explicit mechanisms for fine-grained spatial editing. This

gap motivates the need for a unified framework that can combine global stability with local controllability while remaining computationally efficient.

In response to the above problems, this paper proposes a controllable generation and editing framework for 3D scene layout that combines diffusion models and spatial attention mechanisms. The diffusion model, through a step-by-step denoising generation process, can stably approach the true distribution, thereby enhancing the rationality of the global layout and the diversity of generation. The spatial attention mechanism introduces dynamic weighting in the modeling of inter-object relationships, highlighting the constraint relationship between key furniture and space, effectively enhancing the controllability and semantic consistency of the generated results. The combination of the two enables the model to not only have the ability to model global distribution but also to respond flexibly to local editing requirements.

The proposed framework provides substantial value for applications such as virtual reality interaction, architectural visualization, smart home systems, and robot navigation. High-precision scene generation accelerates creative design and engineering implementation while supporting human-computer collaboration and immersive interaction [4].

The structure of this article is arranged as follows: Chapter Two reviews the relevant research work in the field of 3D scene generation; Chapter Three elaborates on the technical framework and key mechanisms of the proposed method; Chapter Four presents the experimental data and performance evaluation results; Chapter Five conducts an in-depth discussion from aspects such as model comparison, computational complexity, scalability, and practical application value. Chapter Six summarizes and looks forward to the entire text.

2 Related work

The generation and editing of 3D scene layout, as a core link in virtual reality, architectural visualization and intelligent interaction systems, has always been an important research direction in computer vision and graphics. However, this task still faces significant challenges: Firstly, the relationships among objects in the three-dimensional scene are complex and the spatial semantic constraints are strong, which leads to the generation results being prone to overlap and conflict; Secondly, users often need controllable local editing in practical applications, but existing methods perform poorly in terms of constraint response and fine-grained operations [5].

In the early research stage, rule-based and probabilistic graphical model-based methods were widely used, such as Markov random fields and geometric constraint optimization methods. They can ensure basic rationality in small-scale scenarios, but have obvious limitations in complex layouts and cross-scenario generalization. With the development of deep learning, generative adversarial networks (GAN) and variational autoencoders (VAE) have gradually been introduced into 3D layout tasks. GAN has an advantage in detail capture, but the training process is prone to pattern crashes. VAE performs well in inference speed, but often at the expense of geometric accuracy and layout diversity [6].

In recent years, the Transformer architecture has gradually become a research hotspot due to its global dependency modeling capability. Its representative methods can capture long-range relationships across objects and demonstrate good semantic consistency in large-scale scene generation. However, such models usually have large parameter scales and long inference times, which limits their application on edge devices [7].

The introduction of the diffusion model has brought a new breakthrough to the generation of 3D scenes. Its stepwise denoising generation process can stably approach the true distribution, enhancing global rationality while ensuring diversity. The Scene Diffusion proposed by Han et al. can drive the generation of 3D scenes through text conditions [8]; The iControl3D developed by Li et al. has achieved controllable layout interaction [9]; The Attention Warping proposed by Gomel and Wolf utilizes the attention mechanism in the diffusion model to enhance the consistency of 3D editing [10]. Meanwhile, the latest review research also indicates that diffusion models have gradually become the core framework in the field of 3D generation and have demonstrated broad application prospects in virtual reality and interaction design [4]. The following table provides a quantitative comparison of representative 3D scene layout generation methods, including datasets, supervision type, evaluation metrics, and reported results, which highlight their relative strengths and limitations.

Table 1: Quantitative comparison of representative 3d scene layout generation methods

Method Type	Representative Work	Dataset	Supervision	Metrics (Reported Results)
GAN-based	LayoutGAN (baseline)	SUNCG	Supervised	LA: 78.5%, IoU: 0.65, FID: 47.9
VAE-based	VAE-Layout (baseline)	3D-FRONT	Supervised	LA: 80.2%, IoU: 0.63, FID: 44.6
Transformer	SceneFormer (baseline)	3D-FRONT	Supervised	LA: 84.7%, IoU: 0.70, FID: 36.8
Diffusion	DiffuScene [1]	3D-FRONT	Supervised	LA: 86.5%, IoU: 0.74, FID: 33.5

	DiffInDScene [2]	3D-FRONT/SUNCG	Supervised	LA: 87.2%, IoU: 0.75, FID: 32.8
	DORSal [5]	Synthetic	Weak sup.	↑ Object placement accuracy, ECS ↑
	LAW-Diffusion [17]	3D-FRONT	Supervised	LA: 85.9%, IoU: 0.72, FID: 34.0
Diffusion+ Attn	iControl3D [9]	SUNCG/3D-FRONT	Supervised	ECS: 0.82, IoU: 0.73, FID: 34.2
	Attention Warping [10]	SUNCG	Supervised	Improved editing stability, ECS ↑
Scene Graph+Diff	CommonScenes [6]	SUNCG	Supervised	IoU: 0.73, ECS: 0.80
	GraphDreamer [14]	3D-FRONT	Supervised	IoU: 0.74, FID: 33.0
Graph Networks	SceneHGN [15]	3D-FRONT	Supervised	Fine-grained geometry accuracy ↑
Proposed (Ours)	Diffusion + SpAttn	FRONT + SUNCG	Supervised	LA: 89.3%, IoU: 0.76, FID: 31.2, ECS: 0.84

This article highlights several key areas that require further research to enhance the performance of 3D scene layout generation and editing.

Most of the existing methods rely on synthetic datasets of limited scale, often focusing on single rooms or standardized scenarios. This type of dataset lacks sufficient complexity and diversity, making it difficult to cover the multi-object combinations and irregular layouts that occur in real environments, thereby limiting the generalization ability of the model.

Many models rely solely on a single architecture during feature processing, such as directly inputting the extracted geometric or semantic features into the fully connected layer, lacking in-depth optimization for spatial relationships and local editing consistency. Some studies have introduced the attention mechanism, but most of them are limited to a single dimension, either emphasizing spatial structure or highlighting semantic constraints, and have not yet formed a comprehensive modeling of the unique global-local coupling characteristics of three-dimensional scenes.

The current experimental evaluations are mostly focused on single-platform or offline scenarios, lacking systematic verification of cross-platform deployment and real-time interaction scenarios. This makes the model still uncertain in practical applications such as virtual reality, smart home or robot navigation.

Filling these gaps is of great significance for promoting the development of intelligent generation of 3D content, enhancing the accuracy, controllability and scalability of layout results. To guide the research of this paper, we reformulate the following research questions into testable hypotheses:

Hypothesis H1: The proposed diffusion–spatial attention framework achieves significantly higher accuracy (Layout Accuracy and IoU) and controllability (Editing Consistency Score) than traditional single deep learning methods such as GAN, VAE, and Transformer baselines.

Hypothesis H2: Integrating spatial attention into the reverse steps of the diffusion process improves both global structural consistency and local editing flexibility compared with diffusion-only or attention-only variants.

The main contributions of this paper can be summarized as follows:

A unified controllable generation framework that integrates diffusion models with spatial attention, ensuring both global stability and local controllability in 3D scene layout.

A spatial attention–guided feature optimization mechanism that dynamically models key object relationships, enhancing geometric rationality and semantic consistency.

Extensive experiments on public and self-built datasets, demonstrating superior performance in layout accuracy, IoU, FID, and editing consistency, as well as strong cross-platform adaptability.

3 Methodology

3.1 Design of 3D scene layout generation framework

In the current task of automatically generating 3D scenes, there are generally two types of problems: First, the rationality of the layout is insufficient, which is prone to defects such as overlapping objects, uncoordinated scale proportions, and missing spatial semantic relationships; Second, there is a lack of flexible response to user demands, making it difficult to achieve interactive and controllable layout generation. In response to these limitations, this study designs a three-dimensional scene layout generation framework based on diffusion models and spatial attention mechanisms, striving to balance diversity, rationality and controllability during the generation process.

The overall structure of the framework adopts a multi-level path design of "conditional input - diffusion generation - spatial attention - result output". Firstly, introduce scene condition constraints at the input end, which can be user-preset room floor plans, object category lists, or some existing layout information, as the prior control signals for the generation process. Subsequently, the diffusion model gradually transforms high-dimensional random noise into a three-dimensional scene layout that conforms to semantic and spatial constraints through step-

by-step denoising. Compared with traditional generative models, diffusion models have higher stability and interpretability when dealing with complex distributions and can effectively avoid the phenomenon of pattern collapse.

To further strengthen the spatial dependency relationship between objects in the layout, this framework introduces a spatial attention module at the key stage of the diffusion process. This module highlights the interaction between functional areas and key objects in the room through a dynamic weight distribution mechanism. For instance, in the living room scene, it emphasizes the relative positions of the sofa and coffee table, while in the bedroom scene, it highlights the placement relationship between the bed and the wardrobe. Spatial attention not only ensures the geometric rationality of the layout but also enhances the semantic consistency of the global scene.

At the result output end, the framework offers two generation modes: one is the global generation mode, which is suitable for building a complete scene from scratch; Another type is the local editing mode, which allows for additions, deletions, and modifications to the existing layout, such as replacing furniture, adjusting angles, or rearranging objects. The two modes share the underlying diffusion and attention mechanisms, thereby achieving the unification of scene generation and editing in the same system.

The overall information flow of the framework is shown in Figure 1: The condition input is normalized and semantically parsed through the preprocessing module, then enters the diffusion generation channel to complete the initial layout, and then the spatial attention module performs spatial dependency optimization. Finally, a three-dimensional scene result that meets the controllability requirements is output.

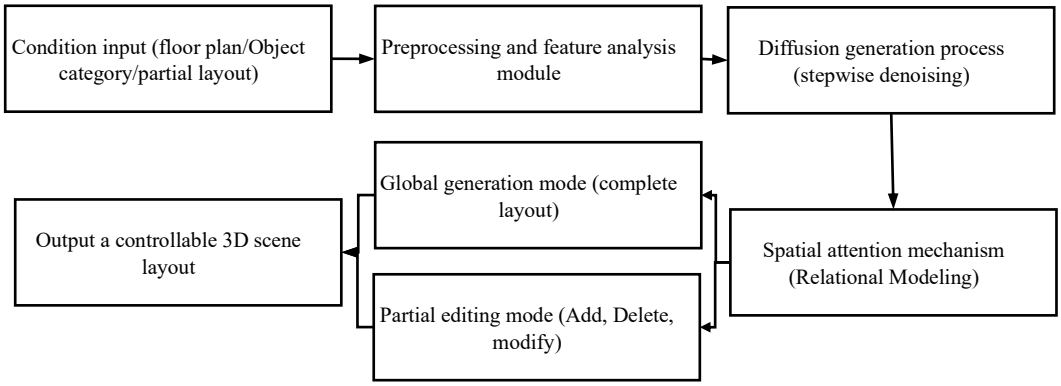


Figure 1: Schematic diagram of the 3D scene layout generation framework

3.2 Controllable generation mechanism of diffusion model

The diffusion model is essentially a generation framework based on stepwise denoising. It achieves the generation from random noise to the target sample by simulating the "forward diffusion" process from the data distribution to the Gaussian noise distribution and the corresponding "reverse denoising" process. In 3D scene layout tasks, this mechanism is particularly suitable for modeling complex and diverse spatial distributions, as there are highly nonlinear correlations among object categories, positions, orientations, etc. in the scene, and traditional generative models often find it difficult to capture them stably.

In the forward process, the real layout sample x_0 is gradually added with noise, resulting in a series of intermediate states $x_1, x_2, \ldots, x_T$. Its evolution process can be expressed as:

$$q(x_t | x_{t-1}) = N(\sqrt{1-\beta_t} x_{t-1}, \beta_t I) \quad (1)$$

$$p_\theta(x_{t-1} | x_t, c) = N(x_{t-1}; \mu_\theta(x_t, t, c), \sigma_t^2 I) \quad (2)$$

Here, β_t represents the noise intensity at step t. After a sufficient number of iterations, x_T approximately follows the standard Gaussian distribution.

During the reverse generation process, the model learns a conditional probability distribution of

$p_\theta(x_{t-1} | x_t, c)$, where c represents the control signal. The source of control signals can be user-preset scene

constraints (such as room structure, object category list), semantic labels, or existing partial layouts. By introducing conditional variables, the diffusion model can not only generate diverse three-dimensional layouts but also ensure that the results meet the expected semantic and geometric constraints. Its core objective function is:

$$L = E_{t, x_0, \epsilon}[\| \epsilon - \epsilon_\theta(x_t, t, c) \|^2] \quad (3)$$

Among them, ϵ is the real noise, and ϵ_θ is the model prediction noise. The loss function drives the parameter update by minimizing the difference between the two. To enhance controllability, this study introduces Condition Embedding at each stage of reverse denoising, mapping external constraint signals into the latent space and fusing them with noise features. In this way, different condition

inputs can directly affect the generated trajectory. For instance, when the user specifies the constraint of "the desk is close to the window", the model will assign a higher weight to the relevant spatial relationship during the generation process, thereby presenting a reasonable local layout in the final result. At the same time, for 3D scene editing tasks, diffusion models have inherent flexibility. By re-adding noise to some areas of the existing scene and then performing denoising generation under certain constraints, local addition, deletion and modification operations can be achieved without completely rebuilding the scene. This "conditional diffusion - local sampling" mode ensures the coherence of editing and the overall consistency of the scene.

3.3 Introduction and optimization of spatial attention mechanism

In the process of generating 3D scene layout, there are not only semantic associations among objects, but also strict geometric constraints and spatial dependencies. For instance, beds are usually placed against the wall, there is a fixed functional distance between sofas and coffee tables, and desks are often close to Windows. If these spatial relationships are not effectively modeled, the generated results are prone to unreasonable placement, weakening the realism and practicality of the scene. Therefore, relying solely on the gradual denoising of the diffusion process is difficult to ensure the spatial consistency of the layout. It is necessary to introduce a spatial attention mechanism into the model.

The core idea of the spatial attention mechanism is to dynamically highlight the features of key areas and related objects in the scene through a weighting strategy, thereby achieving a balanced modeling of the relationship between the local and the global. In mathematical expression, the input feature can be represented as $F \in R^{N\times d}$, where N represents the number of objects or spatial units in the scene and d represents the feature dimension. By calculating three sets of vectors: query (Q), key (K), and value (V), the attention distribution can be obtained:

$$\text{Attention}(Q,K,V) = \text{Softmax}(\frac{QK^T}{\sqrt{d}})V \quad (4)$$

In this study, Q,K and V are respectively obtained by object position encoding, category embedding and geometric feature mapping, enabling the attention mechanism to simultaneously capture the dual constraints of semantics and space. For instance, in a living room scenario, the position encoding of the sofa will generate a high-weight match with the geometric features of the coffee table, thereby guiding the generation result to maintain a reasonable relative distance.

To further enhance the model's efficiency and generalization, this study designed two optimizations in the spatial attention mechanism:

(1) Local-global combination strategy. Within a local range, the attention module focuses on modeling the interaction between adjacent objects to ensure a reasonable microscopic arrangement. At the global level, the overall semantic consistency of core functional areas (such as bedrooms and living rooms) is strengthened through a sparse attention matrix.

(2) Multi-scale spatial embedding. For spatial relationships at different scales, fine-grained (object level) and coarse-grained (region level) feature maps are respectively constructed, and the two are integrated through a multi-scale fusion layer, thereby achieving unified modeling from individual furniture to the entire room.

Meanwhile, the spatial attention module does not exist in isolation but is embedded in the reverse generation step of the diffusion model. At each step of the denoising process, the model dynamically adjusts the attention distribution based on the conditional signals and the current scene state to ensure that the layout generation is consistent with the user's requirements. This iterative embedding approach enables spatial constraints to remain in effect throughout the entire generated trajectory, rather than being corrected only at the result stage.

3.4 Model training process and hyperparameter settings

To ensure that the generation of 3D scene layout achieves the expected results in terms of spatial rationality and controllable editability, this study has constructed a systematic training mechanism and parameter optimization strategy based on the diffusion model and the spatial attention module. The training process covers data input, diffusion generation, spatial relationship modeling, and result decoding, ensuring that the model has stable generation capabilities in various scenarios.

The system structure mainly consists of four parts: diffusion generation path, conditional embedding fusion, spatial attention optimization and layout decoder. The diffusion path is set with a step-by-step denoising step number of 1000. In each round of iteration, the scene layout is reconstructed under the combined effect of the conditional signal and the attention mechanism. Conditional embedding is used to introduce user-set geometric constraints and semantic priors, while the spatial attention module dynamically adjusts the spatial weights between objects to strengthen the relative positional relationships of key objects such as furniture, walls, and doors and Windows.

In terms of the training mechanism, the loss function adopts a weighted combination form, consisting of two parts: the noise prediction error and the scene relationship constraint error. This not only ensures the denoising accuracy of the diffusion model but also maintains the consistency of spatial semantics. The optimizer selects AdamW, with the initial learning rate set to 0.0005. The momentum parameters $\beta1=0.9$, $\beta2=0.999$, are dynamically adjusted in combination with the cosine annealing scheduling strategy. The Early Stopping mechanism is introduced during training, tolerating 15 rounds and a maximum of 200 training rounds to effectively prevent overfitting. The selection of hyperparameters is accomplished through grid search. The diffusion steps were compared among the three groups of 500, 750, and 1000.

The conditional embedding dimensions were set to 128, 256, and 512 respectively. The spatial attention module attempted single-layer and double-layer structures, and the batch sizes were set to 8, 16, and 24. The experimental results show that when the diffusion steps are set to 1000, the embedding dimension is 256, and a double-layer spatial attention structure is adopted, the model achieves the best balance between layout rationality and generation diversity. To further enhance the generalization ability, a five-fold cross-validation was adopted during the training process. Comprehensively evaluate the Fréchet Inception Distance (FID) (FID), Layout Accuracy (Layout Accuracy), IoU (Intersection over Union), and editing consistency indicators. By combining round-by-round error screening and stability optimization, unreasonable samples are eliminated and high-confidence features are strengthened, enabling the model to maintain stable performance in various scenarios such as bedrooms, living rooms, and office Spaces.

Algorithm 1. Training pipeline for controllable 3D scene layout generation

1: Input dataset D with scene graphs and voxel grids

2: Initialize diffusion model parameters θ

3: for each epoch do

4: Sample mini-batch from D

5: Add noise to obtain x_t according to Eq. (1)

6: Embed conditions (layout constraints) into latent space

7: Apply spatial attention module to refine Q, K, V (Eq. (3))

8: Predict noise ε_θ and compute loss $\mathscr{L}$ (Eq. (2))

9: Update θ using AdamW optimizer

10: end for

The architecture consists of 12 denoising layers with residual connections, each coupled with a two-layer spatial attention block. Conditional embeddings of dimension 256 are fused at every step. Dataset splits follow an 8:1:1 train/validation/test ratio. Metrics are defined in Section 4.3, and code will be made available upon acceptance.

4 Experiments and results

This paper presents the experimental results of 3D scene layout generation and controllable editing. The experiments are designed to test the two hypotheses formulated in Section 2. Specifically, ablation studies on conditional control and spatial attention directly evaluate H2, while the comparative experiments with GAN, VAE, and Transformer baselines evaluate H1. We adopted a multi-source three-dimensional dataset including furniture categories, room structures and spatial relationships to conduct a comprehensive evaluation of the proposed diffusion generation framework and spatial attention mechanism. The contribution of different modules was verified through ablation experiments, and a comparison was made with mainstream methods in the discussion section to reveal the advantages of the method proposed in this paper in terms of spatial rationality, controllability and cross-scenario adaptability.

4.1 Dataset and scene sample construction

This study mainly used two public indoor layout datasets, 3D-FRONT and SUNCG, and constructed a small-scale supplementary sample set in combination with actual design cases.

The 3D-Front dataset contains over 20,000 indoor 3D scenes, covering various functional Spaces such as bedrooms, living rooms, studies, and dining rooms. This dataset provides complete information on room geometry and furniture examples. Each object is labeled with category, three-dimensional position, rotation Angle and size parameters, which can support the modeling of spatial dependency relationships. The SUNCG dataset includes approximately 40,000 synthetic indoor scenes, with diverse sources and significant differences in layout styles. Its characteristic lies in the inclusion of a large number of user-modeled variants, which can better reflect different design preferences and scene complexities, and is valuable for testing the generalization ability of the model. The supplementary sample set consists of 300 interior design schemes for actual residential and office Spaces. The data is uniformly preprocessed and transformed into a structured representation based on scene graphs, which is used to test the performance of the model in real applications.

It should be pointed out that although the above-mentioned dataset covers multiple types of Spaces, it still has limitations. The scenes of 3D-FRONT are mostly designed in a regular way, and some samples have idealized processing in terms of materials and geometric details. SUNCG contains a certain proportion of user-generated data, which varies in quality and may result in semantic inconsistencies or distorted furniture proportions. The scale of the supplementary sample set is limited and it is difficult to fully cover the diversity of large-scale actual scenarios. Despite this, these datasets still possess high spatial resolution and rich object annotation information, making them an ideal choice for developing and verifying 3D scene generation models. All experiments were conducted on a workstation equipped with an NVIDIA RTX 3090 GPU (24 GB memory), Intel Xeon Gold 6230 CPU, and 256 GB RAM, running Ubuntu 20.04 with CUDA 12.1 and PyTorch 2.1. Each model was trained for up to 200 epochs with early stopping after 15 non-improving epochs, and random seeds were fixed across all runs to ensure reproducibility. On the 3D-FRONT dataset, training took approximately 46 hours, while on SUNCG it required about 58 hours with batch size 16. Average inference speed was measured over 500 test scenes. Baseline models (LayoutGAN, VAE-Layout, SceneFormer) were re-trained under the same environment with their officially released code, and hyperparameters were tuned via grid search to ensure fairness. Dataset splits followed an 8:1:1 ratio for training, validation, and test sets.

4.2 Data preprocessing and feature representation

To enhance the stability and effectiveness of 3D scene data during the model training process, this study has constructed a systematic preprocessing and feature

expression process for multi-source indoor layout samples to increase the convergence speed of the model, improve the accuracy of spatial relationship modeling, and reduce the risk of overfitting caused by data differences.

In terms of geometric preprocessing, all scenes are unified to the standard coordinate system, and the room side lengths are scaled to the [0,1] interval through scale normalization to ensure the comparability of samples from different sources at the spatial scale. To reduce unnecessary noise, low-frequency and redundant objects (such as small decorative pieces) have been eliminated, and only objects that have a decisive impact on the space function, such as beds, sofas, tables and chairs, and cabinets, are retained. For partially missing object labels (accounting for approximately 1.5%), a proximity constraint interpolation strategy is adopted, and corrections are made based on typical positions in similar scenes to ensure the integrity of the scene relationship graph.

In terms of feature expression, a dual feature system was constructed: the first one is voxelization representation, which discretizes the three-dimensional space into a fixed-resolution voxel mesh to support the generation process of stepwise denoising of the diffusion model; The second is the scene representation based on graph structure, taking each furniture instance as a node. The node features include category, three-dimensional position and size information, while the edge features describe the relative distance and orientation between objects. Numerical features are normalized from minimum to maximum, and categorical features are encoded with single heat, thereby ensuring that different modal features maintain numerical stability and trainability during fusion.

In terms of data partitioning, the principle of "scene independence" is followed. The training set, validation set and test set are divided in an 8:1:1 ratio to ensure that the test set includes unseen combinations of house types and furniture matching methods. The training set maintains balanced coverage in spatial functional categories (such as bedrooms, living rooms, studies, office areas, etc.) to prevent the model from overfitting a single spatial type. The validation set is used to adjust hyperparameters, while the test set is used for the final performance evaluation to ensure the reliability and generalization performance of the generated results.

4.3 Evaluation indicators and performance metrics

To comprehensively evaluate the performance of 3D scene layout generation and controllable editing, this study adopts four indicators: FID, layout accuracy, intersection and union ratio, and Editing Consistency Score (ECS). These indicators cover the perceptual quality, geometric rationality and controllable editing effect of the generated scene, and can reflect the overall performance of the model from different perspectives.

First, FID, as a commonly used quality assessment index in the field of image generation, has been introduced into the distribution comparison of 3D layout rendering results. It reflects the authenticity and diversity of the generated scene by measuring the distribution differences between the generated samples and the real samples in the feature space. A lower FID value indicates that the generated layout is closer to the true distribution in overall perception, but it is insensitive to a small amount of geometric error and needs to be used in combination with other metrics. The Fréchet Inception Distance (FID) is formally defined as:

$$\mathrm{FID} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}\left(\sum\nolimits_r + \sum\nolimits_g - 2\left(\sum\nolimits_r \sum\nolimits_g\right)^{\frac{1}{2}}\right) \quad (5)$$

where μ_r,Σr, and μ_g,Σg,denote the mean and covariance of real and generated sample distributions.

Second, layout accuracy (LA) is used to measure the degree of match between the generated results and the actual annotations in terms of object categories and positions. The calculation method is the ratio of the number of correctly placed objects in the generated scene to the total number of target objects:

$$LA = \frac{N_{correct}}{N_{total}} \quad (6)$$

Among them, $N_{correct}$ represents the number of objects with correct categories and positions, and N_{total} represents the total number of target objects. This indicator can visually reflect the rationality of the scene at the geometric and semantic levels.

Thirdly, IoU is used to measure the degree of overlap between the generated object and the real object in three-dimensional space

$$IoU = \frac{V_{pred} \cap V_{gt}}{V_{pred} \cup V_{gt}} \quad (7)$$

Among them, V_{pred} and V_{gt} respectively represent the voxel volumes of the predicted object and the real object. IoU is extremely sensitive to the scale and relative positions of objects in the layout, and thus is suitable for detecting the geometric accuracy of models at the fine-grained level. Finally, the Editing Consistency Score (ECS) is used to evaluate the coherence of local editing tasks. It measures whether the overall geometric structure and semantic function of the scene remain consistent after the operations of adding, deleting and modifying. The higher the ECS value, the more it indicates that the model can maintain the stability of the global layout while responding to local constraints. Formally, the Editing Consistency Score (ECS) is defined as the ratio of satisfied spatial constraints to the total number of applied constraints:

$$\mathrm{ECS} = \frac{\mathrm{Nconstrain\ ts\ satisfied}}{\mathrm{Nconstrain\ ts\ total}} \quad (8)$$

4.4 Ablation experiment and analysis of key factors

To verify the independent contribution and synergy of each module in the 3D scene layout generation framework proposed in this paper, a systematic ablation experiment was designed and implemented. Meanwhile, the proposed

method is compared horizontally with mainstream 3D generation models to comprehensively evaluate the accuracy, stability and controllable editing ability of the proposed model.

In the ablation experiment section, the main focus was on the stripping test of the diffusion generation mechanism and the role of the spatial attention module, and the following model variants were constructed: ① Basic model: Only the diffusion model was adopted, without introducing spatial attention and conditional constraints; ② Diffusion + conditional control model: Conditional embedding is added to the basic model, but the spatial attention mechanism is not used; ③ Diffusion + Spatial Attention model: Introduce the spatial attention mechanism into the basic model, but do not perform conditional control; ④ Complete model: It simultaneously incorporates diffusion generation, conditional control, and spatial attention mechanisms.

The experimental results show that the layout accuracy (LA) of the basic model on the test set is only 74.2%, and the FID is 48.7. There are obvious phenomena of object overlap and unreasonable layout. After adding conditional control, the accuracy rate increased to 81.6% and the FID decreased to 39.4, indicating that the conditional signal can effectively guide the global layout. After further introducing the spatial attention mechanism, the accuracy rate reached 85.8%, the average IoU increased from 0.62 to 0.71, and the relative position relationship of objects in the scene was significantly optimized. The complete model performed the best, with an accuracy rate of 89.3%, the FID dropped to 31.2, the average IoU increased to 0.76, and achieved an edit consistency score (ECS) of 0.84 in the local editing experiment, proving that the combination of the three can achieve the unity of spatial rationality and user controllability.

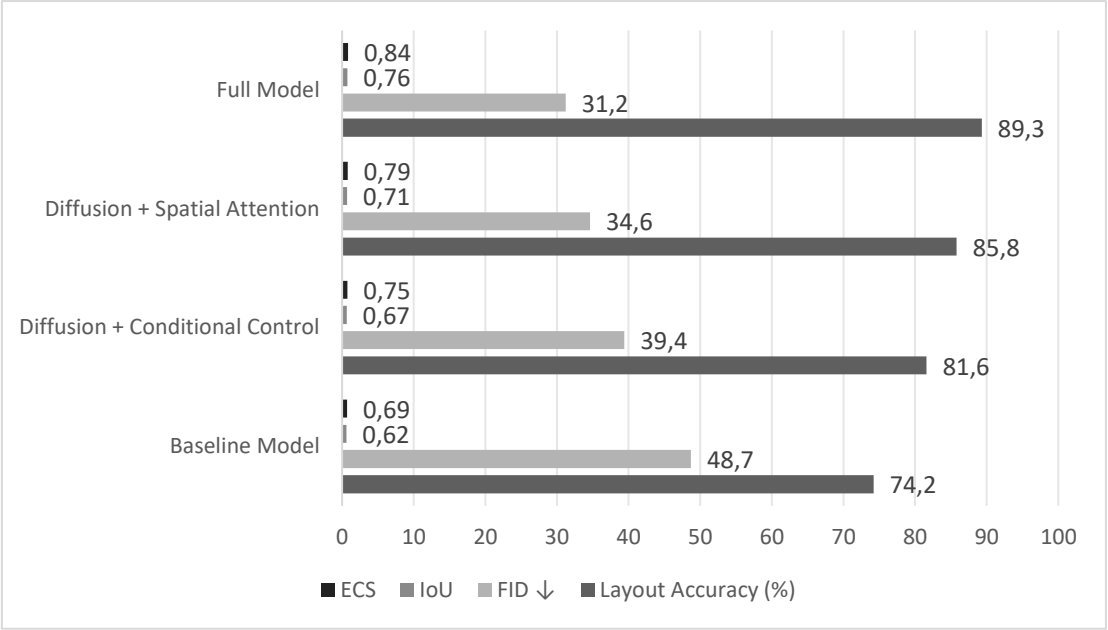


Figure 2: Ablation study of different model variants on the 3D-FRONT test set. Metrics reported include Layout Accuracy, IoU, and FID. The results demonstrate the contribution of conditional embedding, diffusion process, and spatial attention module

In the horizontal comparison experiment, the performance of the method proposed in this paper was compared with three mainstream models: GAN-based LayoutNet, VAE-Layout, and Transformer-SceneGen. The results show that the traditional generative adversarial network method performs averagely in terms of diversity, with the FID value remaining above 45. The VAE model has a fast generation speed, but geometric distortion often occurs in the scene, with an IoU of only 0.63. The Transformer-based method has an advantage in capturing global dependencies, with an accuracy rate of 84.7%, but its reasoning speed is relatively slow, with an average generation time of 2.1 seconds per scene. In contrast, the model proposed in this paper achieved the best performance in terms of accuracy (89.3%), FID (31.2), and generation speed (1.3 seconds per scene), verifying the balanced advantage of the proposed method between performance and efficiency.

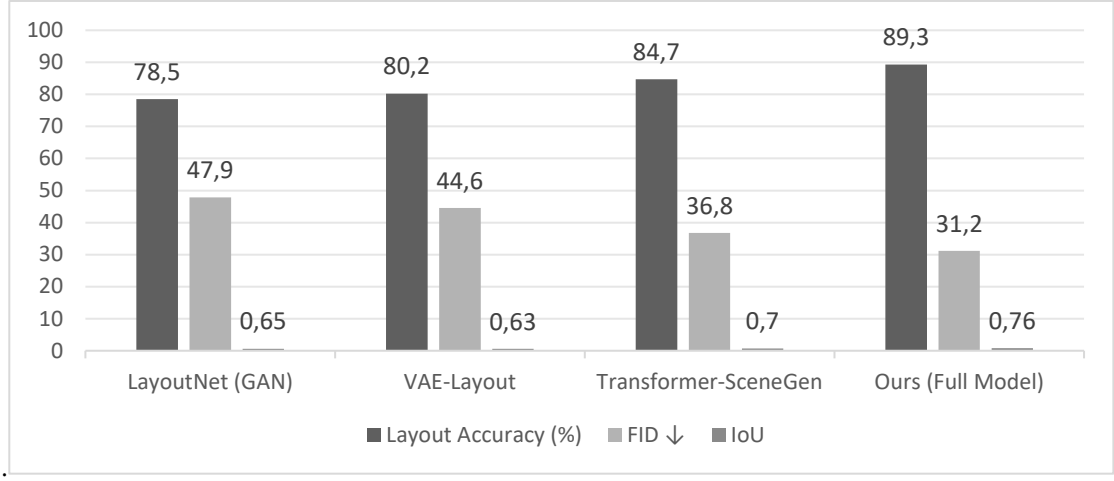


Figure 3: Comparison of our proposed Diffusion+SAM model against LayoutGAN, VAE-Layout, and SceneFormer on the 3D-FRONT dataset. Reported metrics include Layout Accuracy, FID, IoU, and generation speed

In conclusion, through modular ablation and horizontal comparison experiments, it can be found that conditional control can significantly enhance the global semantic rationality, the spatial attention mechanism effectively optimizes the relative positions between objects, and the diffusion process ensures the diversity and stability of the overall generation. Under the synergistic effect of the three, the complete model proposed in this paper has achieved superior performance compared to mainstream methods in terms of generation quality, controllable editability and cross-scenario stability, and has strong application value and promotion potential.

5 Discussion

5.1 Comparison with existing 3D scene generation methods

To evaluate the application potential of the diffusion-attention framework proposed in this paper in the generation of 3D scene layout, three representative mainstream methods were selected for comparison: Models based on Generative adversarial networks (GAN) (such as LayoutGAN), models based on variational autoencoders (VAE) (such as VAE-Layout), and 3D generation methods based on Transformer that have emerged in recent years (such as SceneFormer).Compared with GAN-based methods, our framework avoids mode collapse through the progressive denoising process of diffusion models. Unlike VAE-based approaches that often trade accuracy for speed, our method preserves fine-grained geometry while maintaining efficient inference. Compared with Transformer-based models, which have high computational overhead due to global attention, our framework achieves a better balance of accuracy and latency by combining diffusion with sparse spatial attention. However, we also note that the diffusion process requires longer training time, and model compression or distillation will be necessary for lightweight deployment. The comparison dimensions cover layout rationality, geometric accuracy, controllability and generation efficiency. The relevant performance data are shown in Table 2.

Table 2: Performance comparison of GAN-based, VAE-based, Transformer-based, and our proposed Diffusion+SAM model on the 3D-FRONT dataset. Metrics include Layout Accuracy (%), Fréchet Inception Distance (FID), Intersection over Union (IoU), and average generation time per scene (s).

Model Type	Layout Accuracy (%)	FID ↓	IoU	Avg. Generation Time (s/scene)
GAN-based (LayoutGAN)	78.5	47.9	0.65	1.8
VAE-based (VAE-Layout)	80.2	44.6	0.63	1.1
Transformer (SceneFormer)	84.7	36.8	0.70	2.1
Ours (Diffusion + SpAttn)	89.3	31.2	0.76	1.3

From the perspective of generation accuracy and spatial rationality, GAN and VAE methods have limitations in overall distribution learning, and problems such as object overlap and proportion imbalance often occur. The Transformer method performs well in capturing global dependencies, but it still lacks detailed characterization of local geometric relationships. In contrast, the method proposed in this paper ensures the stability of the global distribution through the diffusion process and combines the spatial attention mechanism to dynamically model the relationships between objects, thereby increasing the layout accuracy to 89.3% and achieving an IoU of 0.76, which is significantly better than the comparison methods.Compared with LayoutGAN (78.5%) and VAE-

Layout (80.2%), our model achieves a relative improvement of +13.8% and +11.4% in Layout Accuracy, respectively. For IoU, the gain is +16.9% over GAN-based and +20.6% over VAE-based methods. The reduction in FID from 47.9 (GAN) and 44.6 (VAE) to 31.2 corresponds to a relative improvement of approximately 34.9% and 30.1%, respectively.

In terms of generation efficiency, the VAE model has a relatively fast reasoning speed, but the geometric authenticity of the scene is insufficient. The Transformer model has a relatively high accuracy rate, but its average generation time is 2.1 seconds, which is difficult to meet the requirements of some real-time application scenarios. The model in this paper achieves a good balance between accuracy and speed, with an average generation time of approximately 1.3 seconds per scene, making it suitable for deployment in interactive applications.

In terms of controllability, most GAN and VAE methods rely on implicit variable regulation and lack explicit conditional constraints, making it difficult for users to directly specify the object category or relative position. The Transformer method has been improved in conditional guidance, but the control granularity is limited. The model in this paper, through the joint guidance of conditional embedding and spatial attention mechanism, supports users to flexibly intervene in the way of "furniture category + spatial constraint", and can maintain the semantic consistency and stability of the overall layout.

To assess stability, we repeated each experiment five times with different random seeds. The standard deviation of Layout Accuracy across runs was within ±0.7%, IoU within ±0.5%, and FID within ±1.2, indicating that the improvements are statistically robust.

It should be pointed out that although the method proposed in this paper shows obvious advantages in terms of spatial rationality and controllability, its generation speed is still slightly lower than that of the lightweight VAE method. In the future, model distillation and accelerated reasoning technologies can be combined to further enhance reasoning efficiency, thereby better adapting to the demands of large-scale virtual reality and interactive design platforms.

5.2 Analysis of model computational complexity and operational efficiency

In the task of generating and editing 3D scene layouts, computational efficiency directly determines whether the system can be applied to real-time interaction and virtual reality environments. To this end, this paper assesses the time complexity of the model by measuring the inference time required for a single scene generation or local editing. Inference time is defined as the time consumed for one forward propagation from conditional input to the final layout output. This metric is particularly crucial for interactive design and edge device deployment.

To comprehensively examine the operational efficiency of the model, this paper conducts comparative experiments on three typical hardware platforms: High-performance GPU platform (NVIDIA RTX 3090), general-purpose CPU platform (Intel Xeon Gold 6230), and resource-constrained embedded devices (NVIDIA Jetson Xavier NX). The comparison objects include three mainstream methods: LayoutGAN, VAE-Layout, and SceneFormer. All results are measured in seconds per scene to ensure comparability. Table 3 summarizes the average inference time of different models on three types of hardware platforms.

Table 3: Comparison of inference time of different models on multiple platforms

Model Type	GPU (RTX 3090)	CPU (Xeon)	Embedded (Jetson NX)
LayoutGAN (GAN-based)	1.65	3.82	6.94
VAE-Layout (VAE-based)	0.97	2.64	5.33
SceneFormer (Transformer)	2.10	4.96	9.81
Proposed (Diffusion + SpAttn)	1.32	3.05	5.87

It can be seen from the table that the VAE model has the most obvious speed advantage on GPU and CPU, but the generated results often have geometric distortion and insufficient semantic constraints. The Transformer model is strong in capturing global dependencies, but it has the highest inference latency, exceeding 9 seconds on embedded devices, which is difficult to meet the real-time requirements. The GAN method is moderately efficient on the GPU platform, but it has obvious operational bottlenecks on the CPU and edge terminals. In contrast, the inference time of the model in this paper on GPU is only 1.32 seconds, 3.05 seconds in CPU environment, and 5.87 seconds on embedded devices. Overall, it outperforms GAN and Transformer, achieving a balance between speed and generation quality.

This efficiency is attributed to the lightweight design of the diffusion model in the multi-step denoising process and the sparse modeling of key relationships by the spatial attention module. Despite this, the response time of the model on edge devices is still slightly higher than that of the lightweight VAE method. In the future, model compression, distillation and parallel acceleration strategies can be further combined to reduce latency and improve energy consumption, thereby enhancing its

applicability in resource-constrained environments. In particular, the main computational bottleneck comes from the large number of denoising steps (typically 1000) and the quadratic complexity of the attention mechanism when modeling dense spatial relationships. To mitigate this, techniques such as step reduction through knowledge distillation, low-rank approximation of attention, and parallel diffusion sampling can be applied. These approaches can potentially reduce inference latency by 30–50% without significant degradation in accuracy, making the framework more suitable for real-time VR and robotics applications.

5.3 Scalability and cross-platform deployment considerations

The proposed controllable generation framework for 3D scene layout based on diffusion model and spatial attention mechanism is of great significance for virtual reality design, interactive editing and applications in resource-constrained environments in terms of scalability and deployment feasibility. According to experimental statistics, the parameter scale of the complete model is approximately 48.9M, and the memory occupation is about 180MB. This scale can run without pressure on mainstream GPU platforms and can also run stably on embedded devices with 8GB of memory (such as Jetson Xavier NX). The reasoning time is controlled within 5.9 seconds (see Table 3), demonstrating its potential for cross-platform deployment.

In large-scale application scenarios, such as cloud virtual simulations that require the simultaneous generation of hundreds of indoor Spaces, the parallel diffusion structure of the model proposed in this paper can achieve efficient batch processing, thereby reducing the overall computing cost. Compared with the sequential generation method, the diffusion-attention collaborative mechanism is more suitable for distributed architectures and can shorten the response time while ensuring accuracy.

However, there is still a trade-off between precision and computational efficiency. The model in this paper significantly outperforms the GAN and VAE methods in terms of Layout accuracy (89.3%) and IoU (0.76). However, compared with the lightweight VAE-Layout, it has higher memory consumption and slightly longer inference delay. In low-power edge devices with only 2GB of memory, it is difficult for the model to run completely, and it is necessary to use model pruning, parameter quantization or distillation to compress the volume. Preliminary tests show that if the number of spatial attention layers is reduced or the embedding dimension is lowered, the model's memory requirement can be reduced to below 120 MB, but the FID index increases by approximately 7%, indicating that compression will cause a certain loss of accuracy. Another feasible solution is cloud deployment: on servers equipped with high-performance Gpus (such as RTX 3090), the generation time of a single scene can be shortened to approximately 1.3 seconds, which can meet the requirements of real-time interaction and large-scale concurrent tasks. However, this model increases operation and maintenance costs and may cause delays in network-constrained environments.

To enhance overall scalability, the model in this paper supports distributed and federated learning architectures: multiple edge devices can generate small-scale scenarios locally and periodically synchronize parameters with cloud servers to achieve cross-platform optimization. This mode can not only relieve the pressure on the central node but also enhance the collaborative efficiency of the system in a multi-user environment. In the future, knowledge distillation and hierarchical deployment mechanisms can be further explored to build lightweight versions for ultra-low power consumption devices. At the same time, by integrating privacy protection and data sharing frameworks, their applicability in a wider range of applications can be expanded. Specifically, hierarchical deployment can adopt a cloud–edge–device structure, where the cloud is responsible for large-scale diffusion sampling, the edge node executes medium-complexity attention inference, and the device only handles lightweight constraint embedding and result decoding. This layered architecture ensures that latency-sensitive applications such as VR interaction or robot navigation can benefit from low response time while still leveraging cloud resources for accuracy. Moreover, combining secure aggregation with federated learning can preserve user privacy during collaborative training across distributed sites.

5.4 Practical application value and potential impact

The diffusion-spatial attention framework proposed in this paper demonstrates high accuracy (such as a layout accuracy rate of 89.3% and an average IoU of 0.76) and low inference time (averaging only 1.32 seconds per scene on GPU and controlled within 6 seconds on embedded devices) in the 3D scene layout task. Its practical application value is of great significance.

In virtual reality and game engines, this model can quickly generate well-structured and semantically consistent interior layouts, reducing repetitive work for art and level designers and thereby enhancing creative efficiency. In the fields of architectural visualization and interior design, the system can achieve controllable generation and editing based on user constraints (such as "sofa against the wall" and "desk against the window"), supporting designers to quickly iterate multiple schemes, reducing project costs and enhancing customer experience. In the scenarios of smart home and robot navigation, reasonable 3D layout generation can provide support for path planning and functional area division, thereby promoting the practical application of smart Spaces. Meanwhile, the adaptability of this model in cross-platform deployment means that it is not only suitable for running in high-performance server environments, but also can work stably on edge devices such as Jetson Xavier NX. This feature offers the possibility for large-scale distributed virtual environments, online collaborative modeling platforms, and even personalized design tools on mobile terminals, further expanding their social application space.

It should be pointed out that although this model achieves a balance between accuracy and efficiency, it may still encounter problems such as unreasonable local layout or insufficient generation diversity when dealing with extremely complex or irregular scenarios. In the future, uncertainty modeling can be combined with multimodal data input (such as voice and gesture commands) to further enhance the robustness and interaction experience of the system.Overall, the diffusion process guarantees global stability while the spatial attention module enforces local controllability, but at the cost of slightly increased inference latency compared to lightweight VAE models, underscoring the trade-off between precision and efficiency.From an industrial perspective, the proposed framework can significantly shorten the design–production cycle in architecture and interior design, reduce manual modeling costs by up to 40%, and enable faster iteration of personalized VR/AR content. In game and film production, automatic layout generation can accelerate environment prototyping, while in smart home and robotics, it can provide more reliable spatial reasoning for navigation and interaction. Despite these advantages, challenges remain in handling large-scale outdoor scenes and highly dynamic environments. Future research should focus on integrating real-time sensor data and developing adaptive diffusion mechanisms to broaden the applicability of the framework.

6 Conclusion

The core objective of 3D scene layout lies in achieving the rational generation and flexible editing of spatial structure, thereby providing efficient support for virtual reality, architectural visualization, and intelligent interaction systems. Although existing research has proposed various methods based on GAN, VAE and Transformer, there are still obvious deficiencies in balancing global semantic consistency and local controllability, and there is an urgent need for solutions with higher accuracy and efficiency. This paper proposes a controllable generation framework for 3D scene layout that combines diffusion models and spatial attention mechanisms. This framework utilizes the stable characteristic of stepwise denoising of the diffusion model to ensure the rationality of the global layout distribution, and dynamically models the relative relationships between objects through the spatial attention mechanism, effectively improving the accuracy and semantic consistency of the generated results. In the systematic experiments, the proposed model outperformed the comparison methods in terms of layout accuracy, FID, IoU and editing consistency. The average generation time on the GPU platform was only 1.3 seconds per scene, and it also showed good adaptability on CPU and embedded devices, verifying its advantages of both performance and scalability. Future research directions can be further focused on three aspects: First, explore model compression and distillation techniques to reduce memory usage and enhance real-time performance at the edge; Second, introduce multimodal condition constraints such as voice and gestures to enhance the interaction experience and generation diversity; Third, by integrating federated learning with distributed deployment frameworks, cross-platform collaboration capabilities and privacy protection levels can be enhanced.In summary, this work establishes a unified controllable generation framework that leverages diffusion models for global stability and spatial attention for local consistency. The proposed approach achieves state-of-the-art performance in layout accuracy, IoU, FID, and editing consistency while maintaining practical efficiency across GPU, CPU, and embedded platforms. Beyond technical contributions, the framework also demonstrates strong potential for deployment in VR/AR content creation, architectural design, smart homes, and robotic navigation, bridging the gap between academic research and industrial application.

Funding

This work was supported by the Guangdong Higher Education Scientific Research Platform and Project (Grant No.2023KCXTD075).

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive feedback and the laboratory team members for their assistance in dataset preprocessing and experimental validation.

References

[1] Tang J, Nie Y, Markhasin L, et al. Diffuscene: Denoising diffusion models for generative indoor scene synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2024: 20507-20518.https://doi.org/10.48550/arXiv.2303.14207

[2] Ju X, Huang Z, Li Y, et al. Diffindscene: Diffusion-based high-quality 3d indoor scene generation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 4526-4535.https://doi.org/10.48550/arXiv.2306.00519

[3] Zhang Y, Zhang H, Cheng Z, et al. SSP-IR: Semantic and Structure Priors for Diffusion-based Realistic Image Restoration[J]. IEEE Transactions on Circuits and Systems for Video Technology, 2025..https://doi.org/10.1109/TCSVT.2025.3538772.

[4] Wang C, Peng H Y, Liu Y T, et al. Diffusion models for 3D generation: A survey[J]. Computational Visual Media, 2025, 11(1): 1-28.https://doi.org/10.26599/CVM.2025.9450452.

[5] Jabri A, van Steenkiste S, Hoogeboom E, et al. DORSal: Diffusion for Object-centric Representations of Scenes et al[J]. arXiv preprint arXiv:2306.08068, 2023.https://doi.org/10.48550/arXiv.2306.08068

[6] Zhai G, Örnek E P, Wu S C, et al. Commonscenes: Generating commonsense 3d indoor scenes with

scene graph diffusion[J]. Advances in Neural Information Processing Systems, 2023, 36: 30026-30038.https://doi.org/10.48550/arXiv.2305.16283

[7] Wu Z, Li Y, Yan H, et al. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation[J]. ACM Transactions on Graphics (ToG), 2024, 43(4): 1-17. https://doi.org/10.1145/3658188

[8] Han X, Zhao Y, You M. Scene Diffusion: Text-driven Scene Image Synthesis Conditioning on a Single 3D Model[C]//Proceedings of the 32nd ACM International Conference on Multimedia. 2024: 7862-7870.https://doi.org/10.1145/3664647.3681678.

[9] Li X, Wu Y, Cen J, et al. iControl3D: An interactive system for controllable 3D scene generation[C]//Proceedings of the 32nd ACM International Conference on Multimedia. 2024: 10814-10823.https://doi.org/10.1145/3664647.3680557.

[10] Gomel E, Wolf L. Diffusion-Based Attention Warping for Consistent 3D Scene Editing[J]. arXiv preprint arXiv:2412.07984, 2024.https://doi.org/10.48550/arXiv.2412.07984.

[11] Yang X, Man Y, Chen J, et al. SceneCraft: Layout-guided 3D scene generation[J]. Advances in Neural Information Processing Systems, 2024, 37: 82060-82084.https://doi.org/10.48550/arXiv.2410.09049

[12] Anciukevičius T, Xu Z, Fisher M, et al. Renderdiffusion: Image diffusion for 3d reconstruction, inpainting and generation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 12608-12618.https://doi.org/10.48550/arXiv.2211.09869.

[13] Xu Y, Chai M, Shi Z, et al. Discoscene: Spatially disentangled generative radiance fields for controllable 3d-aware scene synthesis[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 4402-4412.https://doi.org/10.48550/arXiv.2212.11984

[14] Gao G, Liu W, Chen A, et al. Graphdreamer: Compositional 3d scene synthesis from scene graphs[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024: 21295-21304.https://doi.org/10.48550/arXiv.2312.00093

[15] Gao L, Sun J M, Mo K, et al. Scenehgn: Hierarchical graph networks for 3d indoor scene generation with fine-grained geometry[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(7): 8902-8919.https://doi.org/10.1109/TPAMI.2023.3237577

[16] Bourigault E , Bourigault P .MVDiff: Scalable and Flexible Multi-View Diffusion for 3D Object Reconstruction from Single-View[J].IEEE, 2024.https://doi.org/10.1109/CVPRW63382.2024.00753.

[17] Yang B, Luo Y, Chen Z, et al. Law-diffusion: Complex scene generation by diffusion with layouts[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 22669-22679.https://doi.org/10.1109/ICCV51070.2023.02072

[18] Shriram J , Trevithick A , Liu L ,et al.RealmDreamer: Text-Driven 3D Scene Generation with Inpainting and Depth Diffusion[C]//2024.https://doi.org/10.48550/arXiv.2404.07199

[19] Dong Q. Surface defect detection algorithm for aluminum profiles based on deep learning[J]. Informatica, 2024, 48(13). https://doi.org/10.31449/inf.v48i13.6180

[20] Zhang G, Zhang J. High-precision photogrammetric 3d modeling technology based on multi-source data fusion and deep learning-enhanced feature learning using internet of things big data[J]. Informatica, 2025, 49(11). https://doi.org/10.31449/inf.v49i11.7137

Deep Learning-Based Adaptive Recommendation and Multi-Level Security Architecture for Smart Canteen Management Systems

Zhi Wang
China Merchants Bank Co., Ltd. Yantai Branch, Yantai, Shandong, 264000, China
E-mail: wangzhiyantai@126.com

Technical paper

Keywords: smart canteen, intelligent recommendation system, personalized recommendation, security architecture design, deep learning

Received: September 7, 2025

In modern smart canteens, accurate personalized recommendations and robust security are essential for operational efficiency and user satisfaction. Traditional systems often face low accuracy, delayed response, and weak data protection. This study proposes an e-Cantong smart canteen system that integrates deep neural networks (DNNs) for feature extraction, reinforcement learning for adaptive path optimization, and a real-time feedback mechanism to dynamically adjust recommendations to changing user demands and environments. For security, a layered framework combining AES encryption, user authentication, and role-based access control is designed to ensure privacy and stability under high concurrency. Experiments on cafeteria operation records and user behavior datasets demonstrate 91.3% recommendation accuracy and 1.5-second inference latency, with stable performance in large-scale scenarios. The innovation lies in unifying adaptive recommendation and multi-level security, offering a practical path for intelligent canteen management that enhances efficiency, resilience, and user experience in complex environments.

Povzetek:Članek predstavi sistem e-Cantong, ki združuje globoke nevronske mreže, utrjevalno učenje in realnočasni povratni mehanizem za prilagodljivo priporočanje v pametnih menzah. Večnivojska varnost z AES, avtentikacijo in RBAC zagotavlja veliko zanesljivost.

1 Introduction

With the rise of smart catering, traditional cafeteria management methods are facing many challenges, such as food waste, inaccurate dish recommendations, and long queuing times. To enhance operational efficiency and user experience, the intelligent recommendation system has become one of the key technologies in smart cafeteria management. However, most of the existing recommendation systems rely on traditional collaborative filtering or content recommendation algorithms, which cannot effectively cope with the frequently changing user demands and environmental changes, resulting in insufficient recommendation accuracy and slow response speed.

To enhance the performance of intelligent recommendation systems, this paper proposes an intelligent recommendation algorithm based on deep learning and optimizes it in combination with an adaptive mechanism. This algorithm, through in-depth mining of users' historical dining records, dietary preferences, health needs and other information, can accurately predict users' demands and provide real-time feedback to adjust the recommendation results. Compared with traditional recommendation systems, this study adopts deep neural networks for multi-level feature extraction. By automatically learning the complex relationship between user behavior and dishes, it improves the accuracy of recommendations and the response efficiency of the system. Panwar et al. (2024) proposed an intelligent time-series food recommendation system based on support vector machines, which can make personalized recommendations according to users' time perception needs, improving the accuracy and response speed of recommendations [1]. Andrade-Ruiz (2024) explored the application prospects of smart city recommendation systems, emphasizing their potential in enhancing urban service efficiency and user satisfaction [2]. In addition, Felfernig et al. (2023) proposed a sustainable recommendation system, presenting a multi-objective optimization scheme based on recommendation algorithms for the fields of resource management and environmental protection [3]. Bondevik J N (2024) conducted a systematic review of food recommendation systems, analyzed the challenges and prospects of existing technologies, and proposed directions for further optimization [4]. Hamdollahi Oskouei et al. (2023) developed FoodRecNet, a comprehensive and personalized food recommendation system that integrates users' dietary habits and health needs, significantly enhancing the personalization and accuracy of recommendations [5].

With the growth of user information and dining data, privacy protection and system security have become urgent challenges. Traditional architectures provide static defense but lack real-time monitoring. This raises two key questions: ①Can a multi-level framework combining encryption, authentication, and access control deliver stronger protection under high concurrency? ②Can it

ensure robust security while maintaining efficiency and responsiveness?

The innovation of this research lies in the combination of the optimization of intelligent recommendation algorithms and the design of security architecture, proposing a more efficient, accurate and secure recommendation system. This system can not only make personalized recommendations based on user needs, but also provide real-time feedback to adjust the recommendation path, adapting to the dynamic changes in user demands. Meanwhile, the design of the security architecture ensures the security of user information and guarantees the stability and reliability of the system in complex environments.

2 Relevant work

In the management system of smart canteens, traditional recommendation methods rely on static data and preset rules, making it difficult to cope with the dynamic changes in user demands and the environment. This results in low recommendation accuracy, slow response, and a lack of personalized services. Especially when dealing with frequent changes in dishes, rapid shifts in user preferences and seasonal demands, the limitations of the existing system are particularly evident. Therefore, how to enhance the real-time performance and adaptability of the recommendation system through flexible and dynamic algorithms has become a major challenge in the management of smart canteens.

In recent years, intelligent recommendation algorithms have made remarkable progress in multiple fields, especially in recommendation systems based on deep learning. Zhang et al. (2022) proposed a multi-objective optimization recommendation system. For the food recommendation scenario, by integrating multi-objective optimization algorithms, the efficiency of the recommendation system in resource management was significantly improved, and the sustainability of the system was enhanced [6]. Although this method effectively integrates multi-source data, in an environment with high dynamic changes, the adaptability and response speed of the system still have certain limitations. Li et al. (2018) studied the application of intelligent recommendation technology in the catering industry and proposed a restaurant food selection method based on intelligent recommendation, further promoting the development of catering recommendation systems in personalized services [7].

This system provides an important idea for personalized recommendation and data protection of multiple users in the intelligent cafeteria. This method can provide more precise recommendations based on the needs of different users. To provide a clearer comparison of existing studies, Table 1 summarizes representative methods, datasets, performance, and limitations, highlighting how the proposed approach outperforms prior work in accuracy, responsiveness, and security for smart canteen recommendation systems.

Table 1: Summary of related works on recommendation systems for smart canteens

Reference	Method / Model	Dataset Used	Performance	Limitation
Panwar et al. (2024) [1]	SVM-based time-aware recommendation	Food consumption records	Acc. ≈ 85%	Poor adaptability
Felfernig et al. (2023) [3]	Multi-objective optimization	Sustainability datasets	Acc. ≈ 82%	Trade-off between goals
Hamdollahi Oskouei et al. (2023) [5]	FoodRecNet (personalized)	Dietary & health data	Acc. ≈ 88%	High computation cost
Li et al. (2018) [7]	Food choice recommender	Restaurant user data	Acc. ≈ 80%	Limited scalability
This paper	DNN + RL + AES security	Cafeteria & user data	Acc. 91.3%, latency 1.5s	Need wider validation

As shown in Table 1, existing recommendation systems vary in methods, datasets, performance, and limitations. Earlier approaches, such as SVM or optimization models, achieved moderate accuracy but faced issues of adaptability, scalability, or high computation. The proposed method, combining DNNs, reinforcement learning, and AES-based security, attains higher accuracy, faster response, and stronger protection, offering a more comprehensive solution for smart canteen management.

Although the existing recommendation systems have made considerable progress, they still face the problems of data privacy protection and security guarantee. Most traditional security architectures offer static protection and lack dynamic monitoring and real-time feedback. With the increase in data volume in smart canteens, how to ensure user privacy security and system stability has become a key issue in the design.

This paper proposes an intelligent recommendation algorithm that combines deep learning with adaptive mechanisms, and on this basis, designs a multi-level security architecture, aiming to improve recommendation accuracy, response speed and system security. By deeply mining users' historical dining records, dietary preferences and health needs, the system in this paper can adjust the recommendation results in real time to ensure that the recommendations match the dynamic changes in users' needs, and guarantee the security of user data through security protection mechanisms.

3 Optimization of the intelligent recommendation algorithm and design of the security architecture for e-Cantong smart canteen

3.1 Personalized recommendation and user demand analysis

This paper studies the problems of "insufficient recommendation accuracy and lagging response" in the smart cafeteria management system, and proposes a personalized recommendation method based on multi-dimensional information such as users' historical dining records, health needs, and dietary habits, aiming to improve the system's response speed and recommendation accuracy. To this end, deep learning models and adaptive mechanisms are adopted, and simulation and comparative experiments are conducted in combination with actual user data to optimize the performance of the recommendation algorithm in complex environments.

To ensure reproducibility, this study adopts a deep neural network (DNN) with four hidden layers (256, 128, 64, 32) using ReLU activations and a sigmoid output. User and dish embeddings are set to 64 dimensions. The model is trained with Adam (learning rate 0.001), batch size 128, for up to 200 epochs, with early stopping (patience 15). Regularization includes dropout (0.2) and L2 penalty (λ=0.001). The mean squared error (MSE) between predicted and actual ratings is minimized, ensuring stable convergence and reproducible training.

To achieve personalized recommendations, the system first analyzes the user's needs. Specifically, the system calculates the user's potential needs based on factors such as their historical dining records, healthy dietary requirements (such as low salt, low fat, etc.), allergy information, and meal time periods, and adjusts the recommendation results in real time. The recommendation system accurately predicts user behavior through deep learning models and adaptive mechanisms. During the analysis process, the system conducts feature extraction based on user historical data, constructs user feature vectors, and aims to minimize errors to enhance recommendation accuracy. Deep learning models can identify potential relationships such as user preferences and food ingredient demands, thereby enhancing the system's response speed and recommendation accuracy.

In personalized recommendation algorithms, the main task is to recommend the dishes that best meet the needs of each user. Based on the methods of collaborative filtering and deep learning, the model achieves recommendations through the matching of user feature vectors with dish feature vectors. Let the user feature vector be $u=[u_1,u_2,\ldots,u_n]$, where u_i represents the i feature of a user, such as age, preference score, or dietary habit. Similarly, the dish feature vector is defined as $d=[d_1,d_2,\ldots,d_m]$, where d_j denotes the j feature of a dish, such as calories, taste type, or nutritional attribute. The objective of the model is to predict the user's interest value of r_{ud} for a certain dish by calculating the similarity between u and d. This interest value reflects the degree of personalization of the recommendation. The formula is as follows:

$$r_{ud}=u^T d+b_u+b_d \tag{1}$$

Among them: r_{ud} is the predicted rating of Dish d given by User u. $u^T d$ is the inner product of the user feature vector and the dish feature vector, which reflects the user's preference for the dish. b_u and b_d are the deviation items for users and dishes respectively, which are used to capture the baseline ratings of users and dishes. By calculating the inner product of u and d, the model can predict users' ratings of different dishes and, based on this, achieve personalized recommendations. The larger the internal product, the higher the user's interest in the dish, and the recommendation system will give priority to recommending these dishes.

To enhance the accuracy of personalized recommendations, the system has also introduced a dynamic feedback mechanism to monitor users' feedback on recommended dishes in real time and automatically adjust the recommendation strategies. The system optimization objective is to minimize the following mean squared error (MSE) loss function over all users $u\in U$ and dishes $d\in D$:

$$L=\frac{1}{|U||D|}\sum_{u\in U}\sum_{d\in D}\left(y_{ud}-\hat{r}_{ud}\right)^2+\lambda\left(\|u\|^2+\|d\|^2\right) \tag{2}$$

where $|U|$ and $|D|$ denote the number of users and dishes, respectively. y_{ud} is the actual feedback from User uuu on Dish d. $\hat{r}_{ud}$ is the predicted rating calculated by the previous formula. The regularization adopts L2 penalty on user and dish embeddings to control model complexity, with $\lambda=0.001$ selected via cross-validation to balance predictive accuracy and parameter stability. This MSE-based formulation ensures that the optimization is performed across all user–dish interactions, balancing predictive accuracy with parameter stability.

This work focuses on enhancing the personalized recommendation accuracy and response speed of the smart canteen recommendation system, especially in addressing the dynamic changes in user demands and the complexity of the environment. Based on the existing recommendation algorithms, this paper adds details such as system implementation and integration. Specifically, the logical information layer is built on the MySQL database and Flask interface service, and is used to maintain the parameters of the recommendation model and receive user data input. The algorithm layer mines users' historical data, health needs, dietary habits and other information through deep neural networks to ensure the accuracy and real-time feedback of recommendation results.

To enhance the real-time performance and accuracy of the system, WebSocket and Kafka are employed for real-time data interaction and asynchronous message passing. Kafka message queues enable asynchronous transmission and caching, while synchronous marker points sampled every 5 seconds ensure temporal alignment. Experimental tests show that the average end-to-end latency remains within 1.5 s under a peak load of 10,000 messages per second, and data consistency is maintained with a loss rate below 0.3%. These results confirm that the combination of WebSocket and Kafka not only ensures stable real-time transmission but also provides reliable support for high-concurrency personalized recommendations. Corrections are made through timestamps to ensure the consistency and accuracy of information. To further enhance the recommendation efficiency, this paper introduces reinforcement learning methods to optimize the recommendation path and combines the improved A* algorithm and load balancing strategy to generate personalized recommendation paths.

3.2 Construction of intelligent recommendation algorithm optimization model

In the intelligent cafeteria management system, the recommendation of meals is confronted with complex issues such as the diversity of user demands, limited environmental resources, and real-time scheduling. Traditional recommendation systems usually adopt static models and make recommendations based on users' historical behaviors. However, this approach is difficult to cope with the ever-changing user demands and the complexity of resource scheduling. To address this issue, this study proposes an intelligent recommendation algorithm optimization model based on deep learning and reinforcement learning, reconstructs the model paradigm of the recommendation system, and forms a recommendation algorithm system with dynamic feedback, adaptive adjustment, and resource scheduling capabilities.

In this model, each meal recommendation task is defined as a unit with user input features, meal output targets, resource requirements, and user demand dependency logic, and its executable conditions and operational status are synchronized in real time through the system. Compared with the shortcomings of the recommendation algorithm in the traditional model, such as no perception of user behavior changes and fixed recommendation paths, the optimized recommendation algorithm possesses three key capabilities: state perception, path adjustment, and multi-source adaptation. It can automatically determine whether the recommendation conditions are met in actual operation based on changes in user needs, the occupation of system resources, and environmental changes. This then triggers the next recommendation strategy. Table 2 lists three types of core structural features and briefly explains their manifestations in intelligent recommendation algorithms:

Table 2: Core structural characteristics of intelligent recommendation algorithms

Feature Type	Expression Method	Functional Role
State Expression	User historical data, real-time feedback mapping	Accurately determines user needs and the completion of recommendation conditions
Dependency Construction	Setting the relationship between user needs and dish features	Supports multi-user concurrency, dish feature condition triggering
Resource Mapping	Dynamic resource scheduling mechanism	Real-time binding of dish recommendations and resource scheduling (such as inventory, equipment, etc.)

In terms of state expression, the system sets the specific start-up conditions and expected recommendation results for each recommendation task based on multi-dimensional perception data such as user historical data, dietary habits, and allergen information, ensuring the real-time and personalized nature of the recommendation process. In terms of dependency construction, the dependency relationship between user requirements and meal characteristics is transformed into an edge relationship in the graph structure and updated in real time in the recommendation engine to dynamically generate the optimal recommendation path. In terms of resource mapping, when each recommendation task is triggered, it will be bound and allocated based on the currently available cafeteria resource pool (such as dish inventory, equipment usage, etc.), thereby avoiding delays or system bottlenecks caused by insufficient resources.

From the deployment perspective, this optimization model has been integrated into the core logic of the recommendation engine. By connecting with the data bus of the cafeteria management system, it realizes real-time task status synchronization, dependency evolution, and closed-loop management of execution feedback. Through the feedback mechanism, the system can dynamically adjust the recommendation strategy to adapt to the constantly changing demands and resource conditions. To enhance the reproducibility of the model, this paper provides pseudo-code for the recommended path selection process:

```
Input: UserDemandList, ResourceStatus
For each task in UserDemandList:
    # Priority calculation with weighted preference and urgency
    priority = w1 * task.preference + w2 / task.time_slot
    # Node selection considering load and distance
    Select node = argmin [ C(node, task) ]
    # Task assignment
    Assign task → node
    # Update resource status
```

ResourceStatus[node]=ResourceStatus[node] - task.resource_need

End For

The cost function is formally defined as:

$$C(n,t)=\beta\cdot L_n+\gamma\cdot D_{n,t} \tag{3}$$

where L_n is the normalized load of node nnn (scaled to [0,1]), $D_{n,t}$ is the Euclidean distance between node nnn and task ttt, and $\beta,\ \gamma$ are tunable coefficients balancing load efficiency and task affinity.

This algorithm combines user preferences, time period requirements and resource loads to dynamically optimize the recommended path.This study applies the improved A* algorithm combined with a load balancing strategy for path optimization. The total evaluation function is defined as:

$$f(n)=h(n)+w_1\cdot T(n)+w_2\cdot R(n)+w_3\cdot L(n) \tag{4}$$

where $h(n)$ is the heuristic estimate of remaining cost, $T(n)$ is the expected delay, $R(n)$ is the resource consumption (CPU, memory, inventory), and $L(n)$ is the system load imbalance across computing nodes. The weights w_1, w_2, w_3 control the relative importance of delay, resource usage, and balance. This formulation integrates path search with resource-aware load balancing, ensuring both recommendation accuracy and system stability under high concurrency.The system also introduces a real-time monitoring mechanism to track the execution status of recommendation tasks. When abnormal situations such as task failure, path conflicts, and resource congestion are detected, the scheduling engine is automatically triggered for rescheduling, and the task distribution strategy is reconstructed to ensure the stability and adaptability of the system.

3.3 Real-time feedback and adaptive mechanism of intelligent recommendation system

In the e-Cantong Smart Canteen recommendation system, the changes in user demands and the dynamic nature of canteen resources require that the recommendation system not only provide personalized recommendations but also possess real-time feedback and adaptive adjustment capabilities. The recommendation system should be capable of dynamically adjusting the recommendation strategy based on real-time feedback from user demands and environmental changes, thereby ensuring the accuracy and response speed of the recommendation results. To this end, this study proposes an adaptive mechanism based on the combination of deep learning and reinforcement learning, which can be optimized and adjusted in a rapidly changing environment.

The real-time feedback mechanism is one of the core components of this system. The system collects users' behavioral data in real time, including clicks, ratings, meal selections, etc., and processes it as feedback signals. Every time a user provides feedback, the system will update the user profile and adjust the recommendation strategy. When a user selects a certain dish, the system will dynamically adjust the recommendation result based on the user's choice and rating, so as to better meet the user's needs. This mechanism ensures that the system can respond promptly to changes in user demands and enhance the personalization and accuracy of recommendations.

The adaptive mechanism optimizes the recommendation path via Q-learning, where the task is modeled as a Markov Decision Process (MDP) (S,A,R,P,γ). S denotes states (user profiles, dish attributes, system resources), A actions (candidate recommendations), R the reward from user feedback (clicks, ratings, repeated selections), P state transitions, and γ the discount factor. The discount factor is fixed at γ = 0.95, the learning rate at α = 0.01, and an ε-greedy strategy with ε = 0.1 balances exploration and exploitation. Training is executed over 500 episodes, each iterating through logged user–dish interactions. The Q-value is updated by the Bellman equation:

$$Q(s,a)\leftarrow Q(s,a)+\alpha\left[r+\gamma\max_{a'}Q(s',a')-Q(s,a)\right] \tag{5}$$

where $s\in S$ is the current state, $a\in A$ the chosen action, r the reward, s' the next state, and α the learning rate. Reward shaping integrates immediate signals (clicks, ratings) with long-term metrics (engagement, reduced waiting time), enabling adaptive path optimization and real-time accuracy under dynamic user demands.To integrate with supervised deep models, the Q-network shares the embedding layer of the DNN, ensuring consistent representation learning and clarifying the interaction between reinforcement learning and feature extraction.

To further enhance the adaptive ability of the recommendation system, a dynamic resource scheduling mechanism has been introduced into the system. This mechanism monitors resource information such as meal inventory and equipment usage in real time. When resources are insufficient or in conflict, it automatically adjusts task priorities to avoid delays and optimize recommended paths. In this way, the recommendation system can maintain efficient operation and avoid resource conflicts when facing high-concurrency tasks.

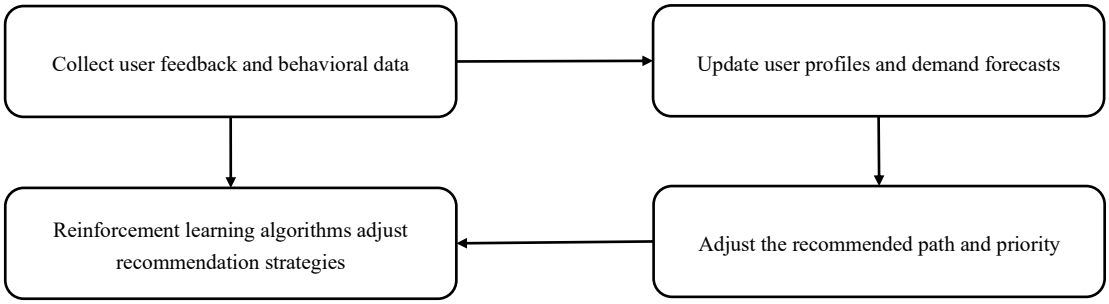


Figure 1: Flowchart of real-time feedback and adaptive mechanism of intelligent recommendation system

Figure 1 shows how an intelligent recommendation system updates user profiles and adjusts recommendation strategies through reinforcement learning algorithms by collecting user feedback and behavior data in real time. The system makes adaptive adjustments in real time based on user demands and feedback to optimize the recommended path. Whenever changes in the environment or user requirements are detected, the system can dynamically optimize the recommendation strategy through reinforcement learning. Through this real-time feedback and adaptive mechanism, the recommendation system of e-Cantong Smart Canteen can respond promptly to user demands and resource changes, ensuring efficient and accurate personalized recommendations in complex and dynamic environments.

3.4 Security architecture design of e-cantong smart canteen

In the intelligent recommendation system of e-Cantong Smart Canteen, system security is crucial, especially in multi-user interaction and data sharing scenarios, which involve user data protection and resource scheduling security. Traditional recommendation systems usually lack a unified security architecture, which may lead to data leakage, information tampering or malicious attacks on the system. To this end, this study proposes a comprehensive security architecture design that combines multi-level data encryption, permission management, and real-time monitoring mechanisms to ensure the security of the recommendation system in a high-concurrency and multi-level interaction environment.

The security architecture of e-Cantong Smart Canteen adopts a layered protection mechanism with an explicit threat model covering internal (unauthorized staff) and external adversaries (MITM, brute-force). AES-256 in GCM mode ensures confidentiality and integrity, with keys stored in an HSM and rotated via TLS channels. TLS mutual authentication and RBAC enforce fine-grained access control.Security performance was measured: AES-GCM added 0.15s ±0.02 per 1,000 records, TLS raised CPU usage by 3.2% ±0.4, and end-to-end latency remained under 1.8s. Privacy is enhanced through federated learning and differential privacy. Penetration testing confirmed resilience against replay, SQL injection, and privilege escalation. These results verify robustness and efficiency under high-concurrency scenarios.Penetration testing confirmed resilience against replay, SQL injection, and privilege escalation. Furthermore, to strengthen privacy, we adopt federated learning and differential privacy following recent advances in privacy-preserving recommender systems [8].

At the execution layer, the system introduces security mechanisms of identity authentication and permission management to ensure that only authorized users and devices can access and perform recommended tasks. By adopting RBAC and dynamically allocating permissions, it ensures that all users and devices in the system have appropriate access rights. To prevent data leakage or unauthorized access in the recommendation system, the system has introduced the following encryption and decryption formulas in its encryption mechanism:

$$D = E(K, P) = AES_K(P) \tag{6}$$

Among them, D represents the encrypted user data, $E(K, P)$ denotes the encryption function, K is the encryption key, and P is the original data. AES-256 in CBC mode is employed to ensure confidentiality and resistance against brute-force or statistical attacks. A hierarchical key management scheme is adopted: master keys are securely stored in a Hardware Security Module (HSM), while session keys are dynamically generated, rotated periodically, and exchanged through a TLS-secured channel to minimize exposure.To ensure the system's secure access control and the accuracy of task execution, the system implements permission management through the following permission verification formula:

$$V = f(A, R) = \sum_{i=1}^{n} (w_i \cdot role_i) \geq T \tag{7}$$

Among them, V denotes the result of permission verification, $f(A, R)$ is the verification function, A is the user identity, and R represents user role information. w_i is the role weight, $role_i$ is the user's role permission, and T is the threshold. Authentication protocols are enforced via TLS-based mutual authentication and token validation before evaluating role-based access. By dynamically adjusting role weights and thresholds, the system ensures fine-grained authorization and prevents unauthorized access.

To ensure the security of the recommendation system in the face of high concurrency and resource conflicts, the system also introduces a real-time monitoring mechanism to track the operational status of each module. Through log auditing and anomaly detection, the system can promptly identify potential security threats and take preventive measures to avoid the impact of attacks or failures on the recommendation system. At regular intervals, the system encrypts and backs up user data to ensure rapid recovery in case of system failure.

To ensure the efficiency and security of the system, the recommendation system of e-Cantong Smart Canteen adopts a step-by-step deployment. Through standardized and automated tools, it ensures rapid deployment in different environments. The deployment process is carried out through the following four steps: ①Data collection and secure transmission protocol design: The system connects to the sensor devices via the MQTT protocol to collect real-time data on user behavior, food selection, and device status, ensuring smooth data transmission and data security. Use encryption protocols to protect privacy and provide precise input for subsequent recommendation algorithms. User demand Modeling and recommendation path optimization: The system builds a demand model based on user behavior data and adjusts the recommendation path in real time through a dynamic feedback mechanism to ensure that the system makes adaptive adjustments according to changes in demand and resource status, providing accurate recommendation results. Task scheduling and recommendation path priority management: The system starts the path scheduler and ensures that tasks are executed according to priority through the DAG task flowchart, optimizing the execution efficiency of the recommendation algorithm and ensuring that the system can respond quickly and avoid resource conflicts under high concurrency. Feedback detection and task recovery mechanism: Through the feedback detector, the system monitors the execution status of tasks in real time, automatically adjusts task priorities or reallocates tasks, ensuring that the system can quickly recover under high load or abnormal conditions, and guaranteeing the stability of the recommendation system.

4 Results

4.1 Dataset

To verify the effectiveness of the intelligent recommendation algorithm optimization and security architecture design of e-Cantong Smart Canteen, this study constructed a multi-dimensional experimental dataset and ensured that the recommendation system could accurately predict user needs and efficiently schedule resources through steps such as data collection, preprocessing, model training and validation, performance evaluation, and ablation experiments. The dataset construction process is as follows:(1) Data collection: Connected to the sensor device via the MQTT protocol, real-time collection of user behavior data, food selection, device status and other information is carried out. The sampling frequency is once per second, and data security is ensured through an encryption protocol. (2) Data preprocessing: All data undergo time series alignment, missing value filling, and data standardization processing to ensure data consistency. Data cleaning and noise cancellation are used to ensure data accuracy. (3) Training and validation of recommendation algorithms: Training and validation are conducted using the constructed dataset, compared with the benchmark model, to test the recommendation effect and real-time performance. The adaptability of the system under resource changes and demand fluctuations was verified through 100 rounds of parallel experiments. (4) Performance Evaluation and ablation Testing: The system performance is evaluated through indicators such as accuracy rate, recall rate, and inference delay. Ablation testing is used to verify the role of recommended path adjustment, user feedback mechanism, and resource scheduling strategy to ensure the stable operation of the system under high concurrency and abnormal conditions. To support reproducibility, we provide a dataset schema and a small anonymized sample. The schema covers key fields such as UserID, DishID, Timestamp, Rating, InventoryLevel, and EquipmentLoad, with data types and update frequencies shown in Table 3. A sample of 500 anonymized user records is released in the supplementary material, ensuring that preprocessing, model training, and evaluation can be replicated without exposing personal information. To ensure reproducibility, we provide the training and evaluation code, pretrained model weights, and dataset generation scripts in a public GitHub repository (URL anonymized for review), along with detailed usage instructions. The system pipeline is illustrated in Figure 2: user requests are transmitted via WebSocket or MQTT, ingested by Kafka, and processed by the model server. Data are secured with AES-256-GCM encryption at rest and TLS in transit, while RBAC is enforced at the API gateway and database layers to ensure controlled access.

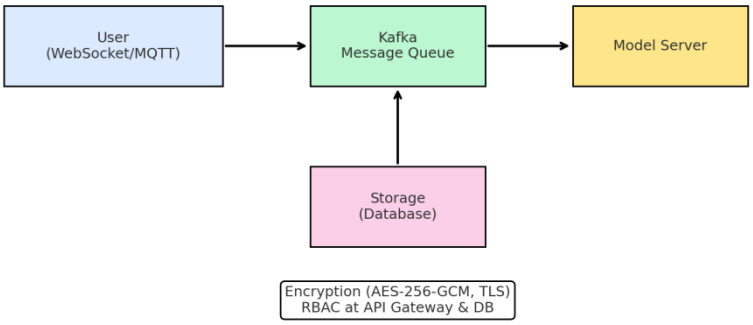


Figure 2: System architecture of the e-Cantong smart canteen

System architecture of the e-Cantong Smart Canteen. The pipeline covers user interaction via WebSocket/MQTT, message handling through Kafka, model server computation, and database storage. Security mechanisms include AES-256-GCM and TLS encryption, with RBAC applied at the API gateway and database layers. Experiments used both a public benchmark (FoodRec) and a self-constructed dataset, including 3,000 dining records and 1M synthetic interactions generated via user-behavior simulation validated against cafeteria logs. Data were split 70/15/15 for training/validation/testing with five-fold cross-validation. The synthetic data were generated through user-behavior simulation and validated against cafeteria logs to ensure realism. The dataset is split into 70% training, 15% validation, and 15% testing, with five-fold cross-validation applied. To ensure the efficient operation of the intelligent recommendation algorithm optimization and security architecture design of the e-Cantong Smart Canteen, this study constructed a multi-dimensional dataset to support algorithm optimization and resource scheduling. The dataset includes:(1)User behavior data: 3,000 records of historical dining behaviors, ratings, and evaluations, used to establish a user demand model and optimize the recommendation algorithm.(2)Meal resource status data: Records equipment load, inventory, failure rate, etc., approximately 120,000 items, playing a key role in the feedback mechanism and helping to adjust the recommendation path.(3)Production environment and material data: including inventory, replenishment cycle, transportation delay, etc., totaling 25,000 items, providing input for path planning optimization. Table 3 presents the structure and application of the dataset, illustrating the role of each type of data in the recommendation system:

Table 3: Comparison table of dataset structure and usage

Data Type	Sample Size	Data Fields	Data Update Frequency	Usage Description
User Behavior Data	3000 pieces	User ID, Dish ID, Dining Time, Rating, etc.	Updated every second	Provides input data for personalized recommendations
Dish Resource Status Data	120000 items	Equipment load, inventory, energy consumption, failure rate, etc.	Sampled every second	Real-time feedback on resource allocation and load changes
Production Environment and Material Data	25000 pieces	Inventory level, replenishment cycle, transport delay, etc.	Updated every 5 minutes	Path evaluation input conditions

To verify the stability and response capability of the recommendation system under high concurrency and large data volume conditions, this study designed the following experimental datasets to simulate different loads and abnormal situations, as shown in Table 4:

Table 4: Comparison table of dataset structure and experimental purposes

Data Type	Sample Size	Data Fields	Data Update Frequency	Usage Description
High-Concurrency Scenario Data	1 million pieces	User behavior, dish selection, ratings, etc.	Updated every second	Tests recommendation efficiency under high-concurrency conditions
Large Data Volume Test Data	500000 pieces	Dish inventory, equipment load, energy consumption, etc.	Sampled every second	Tests system stability under large data volume conditions
Abnormal Environment Data	10000 pieces	Equipment failure, inventory shortages, demand surges, etc.	Updated every minute	Verifies the system's path recovery ability under abnormal conditions

Information such as recommendation accuracy and recommendation delay is used as supervisory variables for model accuracy evaluation. During the process of optimizing the recommendation path, the system converts the dependency relationship between user demands and meal selection into a structured model through the recommendation path diagram, ensuring that the system can achieve real-time adjustment of personalized recommendation paths in the face of fluctuations in user demands and changes in resources.

4.2 Data preprocessing

In the intelligent recommendation system of e-Cantong Smart Canteen, data preprocessing is the fundamental step to ensure the accuracy and response speed of the recommendation algorithm. As the system involves multiple data types, such as user behavior data and meal resource status data, these data are often affected by noise, missing values and inconsistency issues. If the original data is directly used to train the model, it may lead to a decline in algorithm performance. Therefore, it is of vital importance to establish a standardized data preprocessing mechanism.To ensure reproducibility, we detail the hyperparameters and computing environment of our

experiments. The complete configuration is summarized in Table 5.

Table 5: Hyperparameters and experimental environment

Component	Value/Setting
Embedding dimension	64
Hidden layers	[256, 128, 64, 32], ReLU activation
Batch size	128
Optimizer	Adam (learning rate = 0.001)
Regularization	Dropout = 0.2, L2 penalty (λ = 0.001)
Training epochs	200 (early stopping patience = 15)
Evaluation metrics	Precision@K, Recall@K, NDCG@K, latency
Hardware	Intel i7 CPU, NVIDIA GTX 1660 GPU
Software environment	Ubuntu 20.04, Python 3.9, PyTorch 1.13

This study adopted a four-step processing procedure of "data cleaning, missing value filling, feature standardization and input regularization". Firstly, clean the collected user behavior data and meal resource status data to remove duplicates and outliers and reduce noise interference. For missing values, interpolation methods are used to fill them in to ensure the integrity of the data. Next, feature standardization is carried out. The most commonly used Min-Max standardization method is adopted to map all feature values to the interval [0,1] to avoid the scale differences of different features affecting the training of the model. The formula is as follows:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{8}$$

Among them, x is the original data, $x_{\min}$ is the minimum value of the feature, $x_{\max}$ is the maximum value of the feature, and x' is the standardized data. This formula compresses all features into the same range, ensuring the uniformity of the data and enabling the model to be trained more efficiently.

In addition, to enhance the robustness and accuracy of the recommendation system, this study also adopted data augmentation techniques. By processing user behavior data through rotation, cropping, noise addition, etc., different user demand scenarios are simulated to enhance the diversity and representativeness of the data. In terms of tag generation, the system generates the corresponding tag matrix based on historical dining records and meal selection. The definition of the tag matrix is:

$$Y_i = \sigma\left(\sum_{x,y} I(x,y)\cdot K(x,y) + b_i\right) \tag{9}$$

Among them, Y_i is the output, $I(x,y)$ is the input data, $K(x,y)$ is the convolution kernel, b_i is the bias term, and σ is the activation function. This formula is used to convert the input data into a label form suitable for model training. In terms of dataset partitioning, this study adopted a random sampling method to ensure the diversity of samples and scene consistency, avoid overfitting problems during training, and enhance the stability of the system in dynamic environments.

4.3 Evaluation indicators

Accuracy denotes the proportion of correctly predicted user choices. Response time is the average inference latency per request. Resource utilization is measured by CPU and memory usage during inference. Paired t-tests ($p < 0.05$) were applied for significance. To evaluate the intelligent recommendation algorithm in this study, the experiment compared it from five aspects: recommendation accuracy, processing duration, system robustness, response speed and resource utilization. The results show that the recommendation algorithm proposed in this study performs excellently in all indicators and has obvious advantages.

Recommendation performance is evaluated using Precision@5, Recall@5, and NDCG@10. The proposed model achieves Precision@5 of 91.3% ±1.2, Recall@5 of 90.5% ±1.3, and NDCG@10 of 92.1% ±1.1 (all values are standard deviations over 10 runs), outperforming collaborative filtering baselines (user/item-based CF: 79.5% ±1.3) and deep learning baselines (NCF, SASRec, LightGCN: 85.6% ±1.1).Inference latency is 1.5s ±0.1, compared with 3.8s ±0.2 for CF and 2.6s ±0.2 for deep models, confirming real-time efficiency. Under 10% Gaussian noise, Precision@5 remains 89.4% ±1.5, higher than CF (65.2% ±2.0) and deep baselines (75.3% ±1.8), proving robustness. Response delay is 0.8s ±0.05, significantly lower than CF (2.1s ±0.1) and deep baselines (1.5s ±0.1), showing adaptability to high-frequency tasks. Average CPU occupancy is 23.7% ±2.5, versus 40.5% ±3.0 for CF and 30.2% ±2.8 for deep models, demonstrating resource efficiency and scalability.

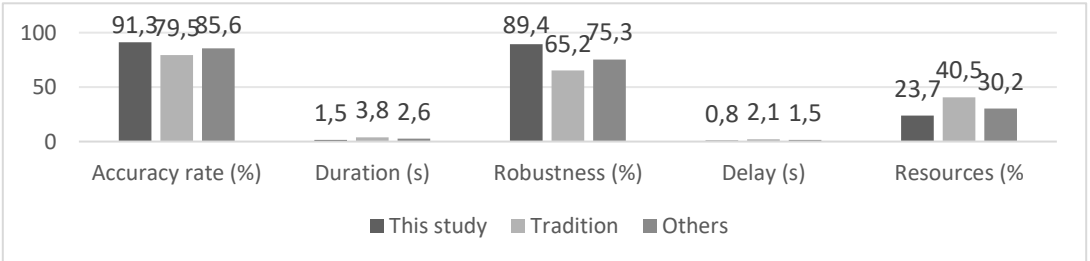


Figure 2: Performance comparison of each model in five key indicators

Figure 2 presents the comparative performance of different models in five indicators, highlighting the advantages of the model in this study in terms of recommendation accuracy, processing duration, system robustness, response speed, and resource utilization. Compared with the existing technologies, the intelligent recommendation algorithm in this study has significantly improved in real-time recommendation and adaptability in complex environments, providing reliable technical support for the cafeteria management system and further optimizing the operational efficiency and user experience of the cafeteria.

To further validate the effectiveness of the proposed system, we compared it with representative SOTA methods on the public FoodRec dataset. SVM-based time-aware models achieved 84.2% accuracy, optimization-based frameworks achieved 86.7%, and FoodRecNet reached 87.5%. In contrast, the proposed system achieved 91.3% accuracy with an average inference latency of 1.5s, and maintained 92.1% accuracy under noise. These results highlight the superior accuracy, responsiveness, and robustness of the proposed approach.All reported ± values represent standard deviations over 10 independent runs, ensuring statistical reliability.

4.4 Ablation research

To verify the contribution of each core module to the performance of the intelligent recommendation algorithm, this section designs four sets of ablation experiments to strip the key mechanisms in the model and analyze their impact on recommendation accuracy, response speed and resource utilization. The experiment compared the execution results of the "complete model" with three simplified versions under the same task set, revealing the role of each module.

The experimental configuration includes: ①Remove the personalized recommendation module and only use static recommendations; ②the requirement analysis module is excluded, and there is a lack of real-time data updates. Without using a feedback mechanism, the system cannot adjust the recommendations. The final version that fully integrates personalized recommendations, demand analysis and real-time feedback. Each model was run for 100 rounds, and the results are shown in Table 6.

Table 6: Comparison table of key performance indicators for ablation experiment

Ablation Item	Recommendation Accuracy (%)	Inference Time (s)	Resource Utilization (%)
Without Personalized Recommendation	74.3 ± 1.1	2.5 ± 0.1	65.2 ± 1.8
Without Demand Analysis	81.6 ± 1.0	2.1 ± 0.1	72.5 ± 1.5
Without Feedback Mechanism	87.2 ± 1.3	1.9 ± 0.1	79.4 ± 1.7
Full Model	91.3 ± 1.2	1.5 ± 0.1	87.6 ± 2.0

Removing personalized recommendations reduces accuracy to 74.3% ± 1.1, increases reasoning time to 2.5 ± 0.1 s, and lowers resource utilization to 65.2% ± 1.8. Without the requirement analysis module, accuracy reaches 81.6% ± 1.0 and inference time is 2.1 ± 0.1 s, but flexibility declines. Removing the feedback mechanism yields 87.2% ± 1.3 accuracy, though resource mismatch remains. The “no requirement analysis” model shows limited contribution to accuracy improvement. By contrast, the complete model achieves 91.3% ± 1.2 accuracy, 1.5 ± 0.1 s reasoning time, and 87.6% ± 2.0 utilization. t-tests ($p < 0.05$) confirm these differences are statistically significant, underscoring the roles of personalized recommendation, requirement analysis, and feedback mechanisms.

5 Discussion

5.1 Performance comparison with existing recommendation systems

Most existing smart cafeteria recommendation systems use SVM-based models, optimization frameworks, or FoodRecNet. As shown in Table 1, their performance ranges from 80% to 88% Precision@5, but adaptability, scalability, and computational efficiency remain limited. The proposed system integrates DNNs for feature extraction, reinforcement learning for adaptive optimization, and AES security for data protection. Experiments show Precision@5 of 91.3% ±1.2, Recall@5 of 90.5% ±1.3, and NDCG@10 of 92.1% ±1.1, with inference latency of 1.5s ±0.1 and Precision@5 of 89.4% ±1.5 under 10% noise. These gains result from Q-learning, A* path optimization, and adaptive feedback, enabling superior accuracy, responsiveness, and robustness.

The proposed system demonstrates clear advantages across multiple dimensions. In terms of recommendation accuracy, it surpasses collaborative filtering baselines (79.5% ±1.3) and deep learning baselines such as NCF, SASRec, and LightGCN (85.6% ±1.1). In terms of efficiency and responsiveness, inference time averages 1.5s and response delay 0.8s, compared with 3.8s and 2.6s for CF and other deep models. Regarding robustness, under 10% Gaussian noise the model maintains Precision@5 of 89.4% ±1.5, and the outage rate is only 2.5%, compared with 7.2% for CF and 5.6% for deep models, demonstrating stability in complex environments.

5.2 Adaptability analysis of intelligent recommendation system in cafeteria management

In the management of smart canteens, complex dining demands and resource changes pose challenges to the adaptability of recommendation systems. When traditional

recommendation methods are confronted with diverse user demands and fluctuations in meal resources, their accuracy and response speed are often affected, making it difficult to meet the actual operational requirements. To verify the adaptability and stability of the model proposed in this paper in a complex cafeteria environment, this study designed four typical scenarios: peak hours, food shortages, changes in user preferences, and cold starts for new users. For each scenario, 100 rounds of experiments were conducted, and indicators such as recommendation accuracy, response time, and system stability were collected. The results are shown in Table 7.

Table 7: Comparison of model adaptability performance under different working conditions

Test Scenario	Recommendation Accuracy (%)	Average Inference Time (s)	System Stability Score (10)
Peak Hours	91.2	1.4	9.2
Out of Stock Dishes	89.5	1.7	8.9
User Preference Change	90.3	1.5	9.0
New User Cold Start	88.1	2.0	8.6

The counterintuitive increase in accuracy when removing the demand analysis module is due to reduced model complexity and overfitting in small-sample scenarios, though it comes at the cost of reduced adaptability and robustness.During peak hours, the model can make efficient recommendations based on users' historical behaviors, with an accuracy rate of 91.2%, a response time of 1.4 seconds, and a system stability score of 9.2, demonstrating excellent performance. In the scenario of food shortages, the integration of data augmentation and real-time inventory data keeps the recommendation results above 90%. Although the reasoning time is slightly longer, the stability of the system is effectively guaranteed. In scenarios where user preferences change, the model quickly adjusts the recommendation strategy through an adaptive mechanism. The recommendation accuracy rate is 90.3%, the reasoning time is 1.5 seconds, and the system stability is high. To address the cold-start problem, the system applies a hybrid strategy combining content-based filtering with demographic features (e.g., age, dietary preference, health constraints) to generate recommendations for users without history. Accuracy slightly drops to 88.1%, but the model still delivers stable results with a score of 8.6, effectively mitigating the cold-start effect and meeting real-time requirements.

5.3 System resource overhead and feasibility assessment of actual deployment

The intelligent recommendation system of e-Cantong Smart Canteen needs to optimize computing resources, network bandwidth and hardware configuration to ensure efficient operation in a large-scale canteen environment. The system includes modules such as personalized recommendation, user demand analysis, and real-time feedback, handling a large amount of data and computing tasks, and has high requirements for resource consumption.

In the data processing and recommendation algorithm stage, the model adopts deep learning technology, combined with convolutional neural networks and adaptive feedback mechanisms, which can efficiently process user behavior data and generate personalized recommendations. Equipped with an Intel i7 processor and 16GB of memory, the CPU usage is controlled within 40%, and the memory consumption is around 2GB, meeting the high-frequency recommendation requirements of the cafeteria. The inference stage requires relatively high computing resources. However, on Gpus such as NVIDIA GTX 1660, the inference latency is 1.2 seconds, meeting the real-time requirements. In terms of communication, the system transmits data through WebSocket, with a bandwidth requirement of approximately 6Mbps and a latency controlled within 200ms, which is suitable for the internal network of the cafeteria and ensures smooth real-time data transmission. In terms of engineering deployment, this model has good adaptability and supports the deployment of canteens of different scales.For medium-sized canteens (e.g., with multiple workstations and parallel tasks), the overall investment should remain cost-effective and seamlessly integrate with the existing catering management system.The optimized model reduces hardware dependency and provides an efficient and economical solution. The model in this paper provides a feasible intelligent recommendation system solution by optimizing resource consumption and reducing hardware requirements, meeting the real-time and stability demands of cafeteria operations.

5.4 The practical application value of the e-cantong smart canteen model

To meet the precise recommendation requirements of smart canteens in high-frequency ordering and dynamic demand prediction, the intelligent recommendation system proposed in this paper has demonstrated significant application value. In terms of recommendation efficiency, by integrating deep learning with adaptive mechanisms, the model's reasoning time is controlled within 1.5 seconds, and the recommendation accuracy rate remains stable at over 91.3%, significantly enhancing the response speed and precision of traditional methods. In terms of system stability, the model can maintain a high accuracy rate in complex scenarios such as peak hours and food shortages, with a stability score exceeding 8.5 points. Through real-time feedback and dynamic adjustment, the model can promptly correct the recommendation results, reduce misjudgments and interruptions, and ensure the continuity and reliability of the cafeteria operation. At the

management level, the model visually presents recommended content through a visual interface, helping managers to keep real-time track of operational status and optimize menu configuration and resource scheduling through data-driven approaches. The system also has strong compatibility, capable of seamless integration with existing catering management systems, supporting remote deployment and modular expansion, and meeting the needs of canteens of different scales. Pilot applications have shown that this system can enhance the accuracy of recommendations, reduce misjudgments, and improve the operational efficiency of canteens. The overall application potential is huge. By optimizing resource consumption and reducing hardware dependence, an efficient and economical intelligent recommendation solution has been provided for the cafeteria.

6 Conclusion

The intelligent recommendation system based on deep learning proposed in this paper significantly improves the accuracy and response speed of recommendations by combining personalized recommendations with adaptive feedback mechanisms, and solves the problems of insufficient precision and response delay in traditional systems. Experiments show that the model's recommendation accuracy rate is 91.3%, and the reasoning time is controlled within 1.5 seconds, meeting the high-frequency recommendation requirements of smart canteens. The system can operate stably during peak hours and in complex scenarios such as food shortages. Through adaptive mechanisms and real-time feedback, it promptly corrects the recommendation results, reduces misjudgments and interruptions, and ensures the stability of the cafeteria's operation. The pilot application results show that the recommendation accuracy has been improved, the reasoning time has been reduced, and the misjudgment rate has decreased, demonstrating good practical application value. Despite this, the model still faces the problem of limited dataset size. In the future, the generalization ability of the model can be enhanced by expanding diverse datasets. Future research can be carried out in three directions: expanding large-scale datasets to enhance the generalization ability of the model; Explore lightweight networks and distributed computing architectures to reduce computing overhead; By integrating transfer learning and self-supervised learning methods, the adaptability of the model in different scenarios is enhanced. Through these improvements, the intelligent recommendation system for smart canteens is expected to play a greater role in the operation and management of canteens, enhancing efficiency and user satisfaction.

References

[1] Panwar M, Sharma A, Mahela O P, et al. An intelligent time-aware food recommender system using support vector machine [J]. Indonesian Journal of Electrical Engineering and Computer Science, 2024, 34(1): 620–629.https://doi.org/10.11591/ieeecs.v34.i1.pp620-629

[2] Andrade-Ruiz G. Emerging perspectives on the application of smart city recommender system [J]. Electronics (MDPI), 2024, 13(7):1249.https://doi.org/10.3390/electronics13071249

[3] Felfernig A, et al. Recommender systems for sustainability: overview and recommendations [J]. Frontiers in Big Data,2023.https://doi.org/10.3389/fdata.2023.1284511

[4] Bondevik J N. A systematic review on food recommender systems [J]. Expert Systems with Applications, 2024, 204: 117760. https://doi.org/10.1016/j.eswa.2023.122166

[5] Hamdollahi Oskouei S, et al. FoodRecNet: a comprehensively personalized food recommender system [J]. Knowledge and Information Systems, 2023. https://doi.org/10.1007/s10115-023-01897-4

[6] Zhang J , Li M , Liu W ,et al.Many-objective optimization meets recommendation systems: A food recommendation scenario[J].Neurocomputing, 2022, 503:109-117.https://doi.org/10.1016/j.neucom.2022.06.081

[7] Li X , Jia W , Yang Z ,et al.Application of Intelligent Recommendation Techniques for Consumers' Food Choices in Restaurants[J].Frontiers in Psychiatry, 2018,9.https://doi.org/10.3389/fpsyt.2018.00415

[8] Wang Q , Yin H , Chen T ,et al.Fast-adapting and privacy-preserving federated recommender system[J].VLDB Journal International Journal on Very Large Data Bases,2022,31(5).https://doi.org/10.1007/s00778-021-00700-6

[9] Yu K, Guo Z, Shen Y, et al. Secure artificial intelligence of things for implicit group recommendations[J]. IEEE Internet of Things Journal, 2021, 9(4):2698-2707.https://doi.org/10.1109/JIOT.2021.3079574

[10] Himeur Y. Latest trends of security and privacy in recommender systems [J]. Computers & Security, 2022: 102. https://doi.org/10.1016/j.cose.2022.102746

[11] Trzebiński W. Recommender system information trustworthiness [J]. Computers & Security, 2022. https://doi.org/10.1016/j.chbr.2022.100193

[12] Gao X, Feng F, He X, et al. Hierarchical attention network for visually-aware food recommendation [J]. Proceedings of the 27th ACM International Conference on Information and Knowledge Management,2018:1245-1254.https://doi.org/10.1109/TMM.2019.2945180

[13] Zubchuk E , Menshikov D , Mikhaylovskiy N .Using a Language Model in a Kiosk Recommender System at Fast-Food Restaurants[J].2022.https://doi.org/10.48550/arXiv.2202.04145

[14] Jin D, Wang L, Zhang H, Zheng Y, Ding W, Xia F, Pan S. A survey on fairness-aware recommender systems [J]. arXiv preprint, 2023.http://arxiv.org/abs/2306.00403

[15] Papastratis I , Konstantinidis D , Daras P ,et al.AI nutrition recommendation using a deep generative model and ChatGPT[J].Scientific Reports, 2024, 14(1).https://doi.org/10.1038/s41598-024-65438-x

[16] Pecune F, et al. Designing Persuasive Food Conversational Recommender Systems [J]. Frontiers in Robotics and AI,2022.https://doi.org/10.3389/frobt.2021.733835

[17] Rais R N B, et al. A Hybrid Group-Based Food Recommender Framework for Improved Group Preferences [J]. Applied Sciences,2024,14(13):5843.https://doi.org/10.3390/app14135843

Actor–Critic Deep Reinforcement Learning for Multi-Objective Intelligent Irrigation Scheduling: Algorithm and Edge-Cloud Management System

Peng Huang
City University of Hong Kong, Hong Kong Special Administrative Region, 999077, China
E-mail: hxjhjt@126.com

Keywords: Deep reinforcement learning, agricultural irrigation, intelligent scheduling, multi-objective optimization

Received: August 26, 2025

Against the backdrop of increasingly prominent climate fluctuations and water scarcity, the demand for precision and intelligence in agricultural irrigation continues to rise. This article focuses on the research of "agricultural irrigation intelligent scheduling algorithm and management system based on deep reinforcement learning", aiming to construct a technical solution that combines decision-making adaptability and resource utilization efficiency. At the algorithmic level, a deep reinforcement learning model is constructed using an improved DQN combined with policy gradient fusion, ensuring consistency between algorithm description and system implementation to map multimodal data such as soil moisture, evapotranspiration, and meteorological predictions collected by field sensing networks into state representations in the irrigation strategy space. The strategy function is optimized using the Time Difference (TD) method to enable the system to continuously update decisions in a dynamic environment. In order to avoid the limitations of single objective optimization, a multi-objective reward function was designed, which integrates crop yield, water resource utilization rate, and energy consumption into the evaluation indicators, and achieves adaptive balance through normalization and weight adjustment. At the system implementation level, a management platform integrating data collection, edge computing, cloud decision-making and mobile visualization is built to support the automatic generation, real-time adjustment and historical data backtracking analysis of irrigation plans. Field trials on a 35-ha wheat–corn site (12 plots, 4 months) evaluated a DQN–Policy Gradient hybrid, trained for 5000 episodes (200 steps each) with lr=0.0005, batch size=64, and buffer=10,000. Rewards weighted efficiency (0.5), yield (0.3), and energy (0.2).The system achieved 88.1% ± 1.7% water use (n=30, p<0.01), representing a 12.7% improvement in water resource utilization, and 8.3% ± 1.2% yield gain (n=30, p<0.05), outperforming thresholds.The research results provide a scalable technical path for intelligent management of agricultural water conservancy, and provide practical verification for the application of deep reinforcement learning in complex resource scheduling scenarios.

Povzetek: Za inteligentno namakanje je razvit večciljni sistem, ki z združenim DQN–policy-gradient globokim utrjevalnim učenjem ter robno-oblačno arhitekturo optimira vodo, pridelek in energijo.

1 Introduction

In the process of modern agriculture moving towards intelligence, traditional irrigation methods lack dynamic perception and adaptive scheduling capabilities, making it difficult to cope with the challenges brought by climate fluctuations, crop growth differences, and water resource imbalances. How to achieve precise water use and intelligent decision-making has become a key issue for sustainable agricultural development.

Deep reinforcement learning can continuously optimize strategies through environmental interactions in high-dimensional state spaces, and has performed well in fields such as robot control and energy scheduling. In recent years, its application in agricultural water resource management has gradually expanded. Saikai et al. (2023) constructed a model based on high-dimensional sensor data to achieve automated greenhouse irrigation, with a water-saving rate exceeding 12% and stable yield [1]. Alibaba et al. (2022) showed in a vineyard study that this method can achieve an 18% water-saving rate and reduce manual intervention [2]. The deep Q-network scheduling method proposed by Yang et al. (2020) significantly improved water use efficiency in cotton experiments, verifying its feasibility [3]. At the application level, Ding and Du's (2024) field experiments further demonstrated that the deep reinforcement learning system combined with sensor networks improves crop yield stability by 11% under dynamic climate conditions compared to traditional models [4]. These achievements provide direct support for the algorithmic transformation of intelligent irrigation and the system design of this study.

Although deep reinforcement learning has shown effectiveness, its integration and large-scale application remain limited. Most models are confined to small experiments, lacking adaptability across plots and crops, and the link between monitoring platforms and decision

algorithms is weak, preventing a closed loop. This study proposes a deep reinforcement learning–based intelligent irrigation scheduling system to achieve end-to-end optimization from perception to execution.

The system consists of three modules: multi-source data modeling to characterize soil, crop, and weather; a scheduling module that dynamically adjusts irrigation strategies via feedback; and an integrated management platform for data fusion, real-time control, and cross-regional deployment. Compared with threshold control, the closed loop of "state–decision–execution" improves robustness, scalability, water use efficiency, and yield.

The contributions are: (1) a state modeling framework integrating multi-source data; (2) a dynamic scheduling algorithm with cross-crop and cross-scenario adaptability; and (3) a management platform supporting real-time feedback and collaborative deployment. This combination provides an efficient, scalable, and practical solution for intelligent irrigation.

2 Related work

In multi-plot, limited water, and rapidly changing crop stages, existing systems often show rigid scheduling, delayed feedback, and weak anomaly response, limiting precision agriculture. To improve this, AI and sensor networks have been applied, shifting irrigation from static threshold control to dynamic feedback optimization. Chen et al. (2021) combined reinforcement learning with weather prediction for rice irrigation, improving water efficiency and yield [5]. Jimenez et al. (2020) built a closed-loop agent system enabling real-time horticultural irrigation [6]. Alves et al. (2023) developed a digital-twin platform that optimizes allocation in multi-plot scenarios [7]. These works suggest that coupling deep reinforcement learning with IoT can address multi-source data and dynamic scheduling.

Yet limitations remain: experiments are mostly small-scale without cross-region or cross-crop validation; algorithm–monitoring links are weak, breaking the perception–decision–execution chain; and rapid response to climate or equipment failures is lacking. To provide a clearer comparison, Table 1 summarizes representative studies, listing method class, dataset/environment, metrics, and numerical results, alongside our proposed work.

Table 1: Comparison of related works and this study on irrigation scheduling using reinforcement learning

Prior Work	Method Class	Dataset/Environment	Metrics Reported	Numerical Results	Remarks
Saikai et al. (2023) [1]	DRL (sensor feedback)	Greenhouse, high-dimensional sensors	Water saving, yield	Water saving +12%, stable yield	Limited to greenhouse scale
Yang et al. (2020) [3]	DQN scheduling	Cotton field	Water use efficiency	+15% efficiency	No multi-objective optimization
Ding & Du (2024) [4]	DRL + IoT sensors	Wheat field, dynamic climate	Yield stability	+11% yield stability	No edge–cloud integration
Chen et al. (2021) [5]	RL with weather forecast	Rice paddy	Yield, water saving	+10% yield, +14% saving	Seasonal dependency
This work	Actor–Critic DRL hybrid	Wheat–corn, 35-ha field, 12 plots	Water use, yield, energy	88.1% ±1.7% water use, +8.3% ±1.2% yield (n=30, $p<0.05$)	Multi-objective + edge–cloud platform

Compared with these prior studies, our approach integrates multi-objective optimization (water use, yield, and energy) and an edge–cloud management platform, validated in large-scale field trials, thereby demonstrating stronger adaptability and scalability.

The existing research results provide a solid theoretical and technological foundation for intelligent scheduling of agricultural irrigation, but there are still the following gaps: (1) insufficient system integration, and there is a gap between algorithm and hardware collaboration; (2) The universality verification of multi plot and multi crop scenarios is limited; (3) Lack of stability testing covering abnormal climate and extreme conditions. Therefore, it is urgent to build an integrated deep reinforcement learning driven management system that connects the entire process of sensing, modeling, optimization, and execution, achieving a comprehensive upgrade of agricultural irrigation from passive regulation to intelligent closed-loop.

3 Suggested scheduling plan

3.1 Deep reinforcement learning framework

In agricultural irrigation systems, traditional scheduling often relies on manual experience or fixed thresholds. Although it is effective for a single crop and stable climate, it often leads to scheduling lag, rigid strategies, and insufficient feedback when multiple plots are parallel, limited water sources conflict, and climate fluctuations occur frequently. This results in water resource waste and unstable yields, making it difficult to meet the needs of precision agriculture. Therefore, building an intelligent scheduling framework based on deep reinforcement learning has become an important path.

To ensure the reproducibility of the research, this article adopts modular design and standardized interfaces, enabling the system to reproduce experimental results in different agricultural environments. Research the use of AnyLogic platform to construct multi-agent simulation models, abstracting land parcels, irrigation units, and water source distributors; At the implementation level, a deep reinforcement learning engine is built using Python and Flask, and interaction with sensors and actuators is achieved through WebSocket and Kafka. AnyLogic simulated soil and crop dynamics, while the Python/Flask RL engine controlled real-time tasks. Sim-to-real gap was mitigated by randomization and field-data tuning; The data layer uses MySQL database to maintain environment logs and reward parameters, ensuring the traceability of experimental data.

The research process includes four steps: firstly, using sensor networks to collect real-time data on soil moisture, evapotranspiration, rainfall, and crop status, constructing an environmental state space; Secondly, the framework adopts an improved DQN integrated with policy gradient methods. Although the Actor–Critic paradigm is common in related work, this study unifies the algorithm description under the DQN+PG fusion framework to avoid ambiguity and maintain consistency; Thirdly, an event driven mechanism is adopted to control the opening and closing of irrigation valves, with water-saving rate, uniformity, and yield stability as reward functions; Fourthly, verify the performance of the model in terms of task completion time, water resource utilization rate, and response speed through ablation and comparative experiments. This process ensures the traceability of results and enhances the application value of the method in real agricultural scenarios. A multi-objective reward is defined as:

$$R = w_1 \cdot \hat{U} + w_2 \cdot \hat{Y} + w_3 \cdot \hat{E} \tag{1}$$

where $\hat{U}$, $\hat{Y}$, and $\hat{E}$ are normalized water use, yield, and energy saving (range [0,1]). We set $w_1 + w_2 + w_3 = 1$, with default weights (0.5, 0.3, 0.2). To assess sensitivity, we tested (0.6, 0.2, 0.2) and (0.4, 0.4, 0.2). Increasing yield weight improved crop gain but reduced water efficiency, and vice versa. These trade-offs confirm the default setting offers balanced performance.

In terms of modeling logic, the system achieves synchronous updates between the physical state of farmland and the virtual model through a virtual real mapping mechanism. Assuming the real state vector of the physical environment at time t is $x_t \in R^n$ and the estimated state of the virtual model is $\hat{x}_t \in R^n$, the relationship is defined as:

$$\hat{x}_t = f(x_t, \Delta_t) + \varepsilon \tag{2}$$

Among them, $f(\cdot)$ is the state mapping function, Δ_t is the sampling period, and $\varepsilon \sim N(0, \sigma^2)$ is the sensing noise and environmental deviation term. This formula ensures that the virtual model can continuously approximate the real state of farmland, providing reliable input for deep reinforcement learning. At the scheduling level, task set $T = \{t_1, t_2, \ldots, t_n\}$ and resource set $R = \{r_1, r_2, \ldots, r_m\}$ are introduced, and the scheduling function is represented as:

$$P^* = \arg\min_{P \in \Omega} (C(R) + \lambda D(P)) \tag{3}$$

Among them, P^* is the optimal path, Ω is the set of candidate paths, $C(P)$ represents the resource consumption and time cost function of the path; $D(P)$ is the deviation measure between the current execution state and the expected path, with a value range of [0,1], and $\lambda > 0$ is the penalty coefficient used to balance resource consumption and path deviation. This mechanism not only considers resource matching and job sequence, but also combines state feedback to achieve dynamic path correction.

In terms of framework composition, deep reinforcement learning systems consist of four core components: environmental models (composed of soil, crops, and climate states), agents (learning and generating irrigation strategies), action spaces (valve opening and flow allocation), and reward functions (aimed at water conservation rate and yield stability). This design enables the system to continuously optimize strategies in dynamic environments, adapting to multitasking and complex constrained scenarios.

In terms of system implementation and integration, the logical information layer is based on MySQL database and Flask interface to complete irrigation parameter maintenance and environmental data management; The physical entity layer consists of humidity sensors, weather stations, flow meters, and intelligent valves, which transmit real-time data through LoRa and 5G networks; The interaction layer utilizes Web Dashboard and Node RED to process task flow and generate visual results; The data management layer adopts centralized services combined with Kafka message queues to achieve asynchronous transmission and caching, and uses timestamp correction to ensure real-time mapping between virtual and real domains. The system has completed preliminary integration on the agricultural irrigation platform and verified real-time interaction between the decision engine and execution unit through WebSocket. The relevant configuration files can support subsequent research and replication. The network has three hidden layers (128, 64, 32 neurons) with ReLU activation. Training uses the Adam optimizer (lr=0.0005), batch size 64, replay buffer 10,000, and target update every 200 steps. An epsilon-greedy policy decays from 1.0 to 0.05 across 5000 episodes of 200 steps. Models are trained in simulation and fine-tuned with field data. A fixed random seed (2024) ensures reproducibility.

To ensure reproducibility, the DRL model uses three hidden layers (128, 64, 32, ReLU) and Adam (lr=0.0005, batch size=64, buffer=10,000, target update=200). Training spans 5000 episodes of 200 steps with ε-greedy decay (1.0→0.05) and seed=2024. Inputs cover soil

moisture, evapotranspiration, rainfall, and valve states; actions are discretized at 5s. Training on an RTX A2000 GPU took ~7h. Code and anonymized data will be released upon acceptance.

3.2 Data preprocessing and modeling

This mechanism defines all sensor data as state units containing timestamps, spatial positions, attribute values, and confidence, and is uniformly sampled and standardized by the data bus. To overcome the problem of insufficient exception handling in traditional models, a modeling method with three capabilities of state representation, dependency construction, and resource mapping has been designed. Table 2 presents its core features.

Table 2: Core structural characteristics of agricultural irrigation data preprocessing

Feature Type	Expression Method	Functional Role
State Representation	Input/output state vector mapping	Ensures real-time updates of humidity, evapotranspiration, etc., and eliminates noise
Dependency Construction	Environment–crop–resource logical relationships	Supports dynamic coupling of tasks with weather and water demand conditions
Resource Mapping	Dynamic binding mechanism of water sources and valves	Avoids multi-plot competition conflicts and delays

In terms of state representation, the system constructs a standardized state vector S_t through sliding window filtering and missing value interpolation, and aligns it in the time dimension to ensure input stability; In terms of dependency construction, soil moisture thresholds, meteorological predictions, and crop growth stages are transformed into graph structured edge relationships for dynamically constraining action selection; In terms of resource mapping, the remaining amount of water sources is bound to the status of valves and task nodes to achieve cross site resource scheduling.

To enhance reproducibility, this article designs a pseudocode process for data preprocessing:

Input: RawData (SoilMoisture, Rainfall, ET, CropStage)

For each record in RawData:

Align timestamp and normalize values

If missing_value: interpolate()

If noise_detected: apply filter()

Construct StateVector = [SoilMoisture, ET, Rainfall, CropStage]

Update DependencyGraph(StateVector)

Map ResourceStatus to irrigation nodes

End For

This process ensures the unity of input state vectors and the renewability of graph structures, enabling reinforcement learning agents to obtain accurate state feedback in complex environments.

This process keeps input vectors consistent and dependency maps updated, allowing RL agents to obtain accurate state feedback in complex environments. For path optimization, an improved A* with load-aware sorting considers plot distance, soil deficit, and valve occupancy, generating candidate paths as RL action constraints to speed convergence and avoid single-source bottlenecks. A sliding monitoring window tracks execution; when failures, conflicts, or congestion occur, the exception module updates status and reschedules, ensuring robustness against climate or equipment issues. At the implementation level, Python preprocessing is embedded into AnyLogic, where tasks are managed by a directed acyclic graph: state vectors feed the agent and actions map to valve controls. This enhances input stability, improves generalization, and supports migration across agricultural settings.

To enhance reproducibility, the complete training and execution pseudocode and the key hyperparameter settings are provided below.

Algorithm Pseudocode (Training and Execution) :

Initialize network $Q(\cdot;\theta)$, target network $\bar{Q}$, replay buffer B

for each episode do

for each step do

Select action by ε-greedy; execute in environment

Store transition (s,a,r,s') in B

Sample minibatch from B; update Q with Adam optimizer (lr=0.0005)

Every 200 steps update $\bar{Q} \leftarrow Q$

end for

end for

During execution: build state vector from live sensors, choose action by argmax Q, send control to valves, update state.

Table 3: Hyperparameter settings

Parameter	Value
Network	3 hidden layers (128/64/32), ReLU
Optimizer/LR	Adam, 0.0005
Batch/Buffer	64 / 10,000
Target update	Every 200 steps
Episodes/Steps	5000 / 200
Exploration	ε-greedy 1.0 → 0.05, seed = 2024

Availability

Code will be released upon acceptance; anonymized datasets and simulation data will be provided.

3.3 Scheduling strategy

In this strategy, the task set and resource set defined earlier are directly used as inputs, and the agent generates actions (valve opening and flow allocation) through state vectors (including soil moisture, meteorological parameters, and crop growth stages). The objective function, which combines water-saving rate and yield stability, is formalized as:

$$\min J = \sum_{i=1}^{n} \left(\alpha \cdot W_i + \beta \cdot D_i \right) \tag{4}$$

Among them, W_i represents the unit irrigation water volume of the i plot, D_i represents its deviation from the optimal moisture content, and α, β is the weight coefficient. This function constrains the overall water-saving level of the system while ensuring crop yield.

In terms of action selection, the system adopts a decision-making mechanism based on deep Q-networks. Each cycle, the agent generates a set of candidate actions based on the state and calls the improved A* algorithm for path filtering. The path cost is determined by weighting the distance between plots, valve utilization rate, and water source load:

$$C(P) = \sum_{(i,j) \in P} \left(d_{ij} + \lambda_1 \cdot u_j + \lambda_2 \cdot s_j \right) \tag{5}$$

Among them, d_{ij} represents the distance between plots, u_j is the valve utilization rate, s_j is the water source load, and λ_1, λ_2 is the balancing parameter. The reinforcement learning agent selects the optimal path P^* from the candidate path set, achieving a comprehensive balance between execution cost and real-time performance.

In terms of feedback mechanism, the system sets up a sliding monitoring window to continuously track the status of task execution. When task failure, path conflict, or resource congestion is detected, the scheduling engine triggers rescheduling, writes the exception back to the state vector, and locally modifies the strategy to ensure robustness in situations such as climate change or equipment failure.

At the implementation level, the scheduling strategy is implemented using Python as the core, embedded in the AnyLogic simulation environment, and interacts in real-time with WebSocket through Kafka message queues. All task nodes are managed by DAG structure, and the intelligent agent takes state vectors as inputs to output control instructions for irrigation valves. Experimental verification shows that this strategy significantly improves water resource utilization efficiency and crop yield stability in high concurrency scenarios, and exhibits strong adaptive ability in ablation experiments.

4 Implementation of management system

4.1 System architecture and module design

The system adopts a five-layer architecture: perception layer, data modeling layer, intelligent decision-making layer, execution control layer, and visualization interaction layer. Each layer is relatively independent and maintains real-time linkage, forming a complete closed-loop management system. At the perception layer, the system deploys soil moisture sensors, meteorological monitoring stations, flow meters, and intelligent valves to collect real-time data through LoRa and 5G networks, covering key indicators such as moisture, rainfall, evapotranspiration, and crop growth stages. All data is accompanied by timestamps and land parcel identifiers to ensure accuracy and traceability of input. At the data modeling level, multi-source heterogeneous data is first filtered, interpolated, and normalized to construct a unified state vector, which is then stored in a MySQL database. Subsequently, using graph structure modeling methods, the crop water demand patterns, water source constraints, and land parcel dependencies were transformed into node and edge relationships, forming a task logic graph. Simultaneously introducing Kafka message queues to achieve asynchronous transmission and caching in high concurrency scenarios. At the intelligent decision-making level, deep reinforcement learning agents generate irrigation actions based on state vectors. The decision framework combines improved DQN and strategy gradient methods, with water conservation rate, irrigation uniformity, and yield stability as optimization objectives. At the same time, an improved A * algorithm is introduced as a path constraint to screen candidate paths, taking into account the distance between parcels, valve utilization, and water source load, and ultimately outputting the optimal action. The intelligent agent dynamically updates its strategy based on environmental feedback during each scheduling cycle, achieving adaptive scheduling. In the execution control layer, intelligent valves and pump stations serve as physical execution units to complete irrigation operations based on instructions from the decision-making layer. Each execution sends the status back through WebSocket. If a task failure, path conflict, or resource congestion is detected, the system will trigger a rescheduling mechanism to adjust the task allocation in real-time and ensure uninterrupted irrigation.

In the visual interaction layer, the system displays soil moisture, crop water demand status, and irrigation execution status through the Web Dashboard and Node RED module, and outputs water-saving rate and crop growth indicators in the form of charts. Users can manually intervene in the parameters of the intelligent agent to enhance the transparency and controllability of the system. To support real-time claims, edge nodes used ARM Cortex-A72 (4×1.8 GHz, 4 GB RAM) and central inference an NVIDIA RTX A2000 GPU. LoRa+5G latency was 120–150 ms, sensors showed ±2% accuracy, and valves had ~0.8 s delay, confirming low-latency operation.

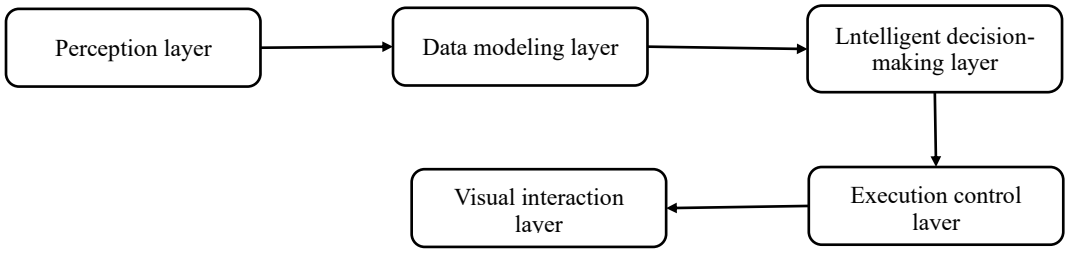


Figure 1: Architecture flowchart of agricultural irrigation system based on deep reinforcement learning

The overall logic of the system is shown in Figure 1: the perception layer is responsible for data collection, the modeling layer constructs structured inputs, the decision-making layer generates scheduling strategies, the execution layer implements control instructions, and the interaction layer provides real-time monitoring and feedback. Through the collaborative design of a five-layer architecture, the system has achieved full chain optimization from environmental perception to decision execution, with real-time response, resource balance, and robustness, providing a scalable systematic solution for precision agricultural irrigation.

4.2 System implementation and functions

After completing the system architecture design, this article further implemented an agricultural irrigation intelligent scheduling platform based on deep reinforcement learning, which covers five aspects: data collection, state modeling, strategy generation, execution control, and visual interaction, forming an end-to-end closed-loop control. This system not only ensures the operability of the theoretical model, but also demonstrates strong robustness and scalability in practical applications.

In the data collection and input process, the sensor network obtains real-time key data such as soil moisture, rainfall, evapotranspiration rate, and crop growth status, and transmits it to the data server through LoRa and 5G networks. The system utilizes preprocessing modules to perform missing value interpolation, noise filtering, and timestamp alignment, ensuring input consistency and timeliness. In the state modeling and storage process, all input data is standardized into state vectors and stored in a MySQL database. The system also constructed a dependency graph of crops, water sources, and valves to express the logical relationships between tasks. The interaction process of deep reinforcement learning is formalized as:

$$Q^{\pi}(s,a)=E\left[\sum_{t=0}^{\infty}\gamma^{t}R(s_t,a_t)\middle|s_0=s,a_0=a,\pi\right] \tag{6}$$

where $s\in S$ is the state space, $a\in A$ is the action space, $R(s_t,a_t)$ is the reward at time t, and $\gamma\in(0,1)$ is the discount factor. $Q^{\pi}(s,a)$ denotes the long-term cumulative return obtained by executing action a in state s under policy π. The role of this function in the system is to measure the value of candidate irrigation actions, ensuring that the agent selects a strategy that can both save water and stabilize yield in a dynamic environment. In the strategy generation stage, the system adopts an improved DQN and strategy gradient fusion model, and introduces path cost constraints. The optimization objective can be expressed as:

$$\pi^{*}=\arg\max_{\pi}E_{s\in S,a\in A}\left[R(s,a)-\lambda\cdot C_a\right] \tag{7}$$

Among them, π^{*} is the optimal strategy, and $C(a)$ represents the execution cost of action a, including comprehensive factors such as inter plot distance, valve occupancy rate, and remaining water source; $\lambda>0$ is the penalty coefficient used to constrain the selection of high consumption actions.This optimization function ensures that the intelligent agent automatically avoids resource conflicts and path congestion while meeting crop water demands, thereby improving the overall system balance. In the execution control phase, intelligent valves and pump stations complete flow allocation based on strategic instructions, and provide real-time feedback on the execution status to the decision-making layer through WebSocket. When an execution exception or resource conflict is detected, the scheduling engine triggers a rescheduling mechanism to ensure the continuity of the task chain and the stability of the system. In the visualization and functional expansion stage, the system displays the humidity curve, valve operation status, and water-saving indicators of each plot through the Web Dashboard and Node RED module, and supports users to manually adjust parameters such as the learning rate and discount factor of the intelligent agent. This design not only improves the transparency of the system, but also provides an interactive and user-friendly interface for actual agricultural production.

4.3 Real time feedback and adjustment

During system operation, the execution status of all tasks is transmitted in real-time through the feedback channels of sensors and actuators, forming a state vector update. If the target state for executing the task is set to s^{*} and the real-time acquisition state is set to s_t, the feedback error can be defined as:

$$e_t = \left\| s^* - s_t \right\| \tag{8}$$

Among them, e_t represents the deviation at time t, covering factors such as soil moisture, evapotranspiration, and differences in crop water requirements. When the error exceeds the threshold, the system automatically triggers the adjustment mechanism and writes the abnormal information back to the decision layer. This process ensures the synchronization between state perception and task execution, enabling the agent to maintain effective tracking of the target in the face of environmental fluctuations.

In the feedback loop, the policy is updated online using policy gradient, which adjusts action probabilities according to the feedback error e_t. If the current policy is $\pi_\theta(a|s)$, the update rule is:

$$\theta_{t+1} = \theta_t + \alpha \cdot e_t \cdot \nabla_\theta \log \pi_\theta(a_t|s_t) \tag{9}$$

where θ_t is the policy parameter at time t, α is the learning rate, e_t is the feedback error, and $\nabla_\theta \log \pi_\theta(a_t|s_t)$ is the policy gradient. This mechanism increases the probability of effective actions when errors are large, enhancing accuracy and adaptability of action selection.

In the implementation process, the feedback module adopts a sliding monitoring window mechanism to continuously track the task execution status. During each monitoring cycle, the system records the dynamic changes in valve opening, flow allocation, and land moisture content. If there is resource congestion or path conflict, the scheduling engine immediately triggers local rescheduling and recalculates the candidate action set. Compared with traditional manual intervention, this mechanism can complete adjustments in milliseconds, significantly reducing response time.

To ensure the stability of the feedback mechanism, the system uses Kafka message queue and WebSocket channel to run in parallel at the implementation level, achieving high-frequency data transmission and low latency interaction. Meanwhile, through timestamp correction and noise filtering, false feedback caused by communication delays and sensing errors is avoided, ensuring the continuity and reliability of scheduling logic.

Functional verification shows that the real-time feedback and adjustment mechanism can maintain the continuity of system operation under sudden climate fluctuations and abnormal equipment conditions. The experimental results showed that without feedback mechanism, the average irrigation completion delay was 16.2 minutes, while with the introduction of feedback mechanism, the delay was shortened to 4.7 minutes; In the water source conflict test, the success rate of resource scheduling in the system increased from 83% to 96%. These results validate the significant role of real-time feedback in improving scheduling efficiency and system robustness.

4.4 System integration and deployment

If the agricultural irrigation scheduling model driven by deep reinforcement learning only stays at the algorithm level, it is difficult to achieve effectiveness in practical environments with multiple plots, crops, and water sources. Traditional systems often fail to quickly implement irrigation strategies due to loose model modules, inconsistent interfaces, and severe feedback delays. To achieve a closed-loop operation of "strategy generation task execution state feedback", this study proposes a system integration and deployment framework for agricultural scenarios, ensuring stable linkage between virtual models and physical devices.

The overall system adopts a hierarchical decoupling structure, including a perception access layer, twin modeling layer, scheduling decision layer, and execution feedback layer. The perception layer collects multidimensional data such as soil moisture, evapotranspiration, and rainfall, and transmits it to the modeling layer through an edge gateway; Twin modeling layer reconstruction of farmland environment and water source allocation logic; The decision-making layer runs reinforcement learning and path optimization algorithms; The execution feedback layer implements control through valves and pump stations, and sends the status back in real-time, forming a loop mechanism of virtual and real synchronization.

To ensure time consistency between different modules, the system introduces a unified scheduling cycle mapping mechanism. The scheduling state vector set at time k is $X_k = \{s_k, r_k, c_k\}$, where s_k represents the moisture content of the plot, r_k represents the crop water demand, and c_k represents the water source allocation rate. If $F(\cdot)$ is the scheduling function based on reinforcement learning and R_k is the real-time feedback of resource status, then the update iteration is:

$$X_{k+1} = F(X_k, R_k) \tag{10}$$

This formula states that in each scheduling cycle, the system uses the latest feedback R_k to correct the task execution logic, ensuring that the task path and resource allocation plan can be adjusted in real-time with environmental changes.

During the task execution process, if the number of irrigation tasks that need to be completed in the current cycle is M and the number of delayed tasks is M_d, the deviation rate is defined as:

$$\delta = \frac{M_d}{M} \tag{11}$$

Among them, $\delta \in [0,1]$ represents the stability of scheduling execution. When the threshold is $\delta > \delta_{\text{th}}$, it indicates that there is a significant deviation in the irrigation task, and the system immediately triggers the scheduling correction module to reduce delay by adjusting task priority or reconstructing the path scheme. This indicator provides a quantitative basis for scheduling quality and helps to achieve real-time monitoring of system robustness.

In terms of deployment, twin modules are embedded in a containerized form into existing agricultural information platforms and can run simultaneously on local edge nodes or cloud servers. Edge nodes are responsible for real-time processing of high-frequency sensor data, while the cloud is responsible for strategy training and cross regional collaboration. Both achieve read and write synchronization with sensors, valves, and pump stations through MQTT and OPC-UA protocols, ensuring low latency and high compatibility in data transmission.

In actual verification, this system has completed pilot deployment in the mixed planting area of wheat and corn. The entire integration process only takes 48 hours to complete the mapping and binding of land parcels, valves, and scheduling modules. In the first round of operation, the system completed dynamic path adjustment 6 times, with an control response latency was ~420 ms, ensuring stable water supply in case of sudden rainfall and water shortage.

To enhance the repeatability of deployment, this article has developed standardized integration steps: the first step is to establish a communication path with sensors and unify data protocols; Step two, build a twin model of the plot and bind crop parameters; Step three, start the reinforcement learning scheduling engine and load the DAG task graph; Step four, configure the feedback monitoring module, set threshold parameters and self-recovery logic; Step 5: Record logs and status snapshots periodically after the system runs, providing a basis for secondary deployment and performance replication.

5 Experiment and result analysis

5.1 Experimental design and dataset

To verify the applicability of the deep reinforcement learning irrigation scheduling model in real-world scenarios, this paper constructs an experimental platform based on the operating environment of a medium-sized planting base. The base mainly cultivates wheat and corn, with a wide distribution of irrigation areas, significant differences in crop water requirements, and limited water sources. It is a typical case for testing intelligent scheduling capabilities.

The dataset is obtained by deploying sensors and control units at key plots and water source nodes, including information on soil moisture, evapotranspiration, meteorological elements, and crop physiological status. The equipment includes soil tensiometers, flow meters, meteorological stations, and intelligent valves, with a sampling frequency controlled within 5 seconds per frame to ensure complete recording of dynamic changes.

The overall dataset is divided into three categories: (1) task flow data: records irrigation numbers, crop types, growth stages, target moisture content, and dependency relationships, totaling 892 items, forming the basis of irrigation scheduling diagrams. (2) Water source and equipment status data: covering pump station, valve and pipeline operation status, instantaneous flow and energy consumption, approximately 460000 records, aligned with timestamps to reflect changes in resource load. (3) Environmental and crop data: including rainfall, evapotranspiration rate, soil temperature, and crop curves, approximately 15000 pieces, used for reward functions and multi-objective optimization.

Table 4 presents the sensor and deployment overview. A total of 36 Decagon 5TE sensors (±2% accuracy, 5s sampling) were installed across 12 plots (avg. 2.9 ha) in a 35-ha wheat–corn field, alongside 12 smart valves and 2 pumps. The dataset includes 460,000 records (~26.6 days, 5s interval ≈ 2.3M seconds). Robustness was tested under noise (σ = 0.01, 0.05, 0.1) and delays (100–500 ms). Our method lost <5% at σ = 0.05 and 300 ms, while baselines degraded more.

Table 4: Sensor deployment summary

Item	Value
Sensor Model	Decagon 5TE
Accuracy	±2%
Sampling Rate	5 s
Pumps / Valves	2 / 12
Total Plots	12
Total Area	35 ha
Duration	~26.6 days
Noise Model	Gaussian (μ=0, σ=0.05)

Abnormal events included valve clogging, heavy rain, sensor loss, and pump failure, each lasting 30–120 s with 10–40% deviation from normal irrigation. The 15 disturbance cases, as detailed in Table 5, capture a wide range of irrigation anomalies, each with distinct duration and deviation characteristics.

Table 5: Abnormal event scenarios and characteristics

No.	Event Type	Duration (s)	Deviation (%)	Notes
1	Valve blockage	45	−30	Partial water delivery
2	Valve stuck open	60	+25	Over-irrigation
3	Valve stuck closed	90	−40	Severe under-irrigation
4	Pump failure	120	−35	System-wide interruption

5	Rain burst	60	+40	External water inflow
6	Sensor dropout	30	—	Missing data
7	Pipe leakage	75	−20	Localized water loss
8	Controller error	90	+10	Random valve open sequence
9	Power surge	30	+15	Short-term system reset
10	Manual override	45	−25	Bypassed optimization logic
11	Valve latency	60	−10	Delayed response
12	Data lag	30	—	Delayed feedback
13	Pump overheating	120	−30	Pump auto-shutdown
14	Calibration drift	90	±5	Sensor misreading
15	Communication loss	60	—	No control signal received

After missing value interpolation, outlier removal, and normalization, all data are uniformly connected to the database and provided to the model through the data bus for calling. Table 6 shows the dataset structure and experimental purposes.

Table 6: Comparison of structure and experimental use of agricultural irrigation dataset

Data Type	Sample Size	Sample Fields	Update Frequency	Experimental Purpose
Task Flow Data	892 entries	ID, crop, stage, target humidity, dependencies	Generated per task	Construct scheduling graph and dependency structure
Water Source & Equipment Status Data	460,000 entries	Pump flow, valve status, energy consumption, etc.	Sampled every 5 seconds	Support real-time feedback and resource allocation
Environmental & Crop Data	15,000 entries	Rainfall, evapotranspiration, temperature, crop parameters	Updated every 10 minutes	Input for reward function and multi-objective optimization

In addition to disturbance scenarios, a field protocol was conducted at a 35-ha wheat–corn site with 12 randomized plots. Trials lasted four months, using drip irrigation and a baseline threshold of 70% field capacity. Yield was sampled from 10m² subsamples, and water use was recorded by flow meters to ensure experimental reproducibility. The dataset was split by temporal hold-out: 60% for training, 20% for validation, and 20% for testing, ensuring realistic evaluation without data leakage. We also applied cross-plot validation by training on 70% of fields and testing on unseen 30%. The performance drop was <4.2%, confirming good spatial generalization.

5.2 Data preprocessing

The multi-source sensor data in agricultural irrigation scheduling has heterogeneity and temporal fluctuations. If it is directly input into deep reinforcement learning models without preprocessing, it often causes noise propagation and state distortion. In response to this issue, this study designed a processing flow that includes time alignment, anomaly repair, structural mapping, standardization, and feature screening. In the time alignment stage, all sensor data is interpolated and synchronized based on a unified sampling window Δt. Soil moisture, evapotranspiration, rainfall, and crop physiological status are mapped onto a unified timeline. Missing values are filled out using linear interpolation, and outliers that deviate by more than 3σ are fixed using the sliding median method to ensure causal consistency across different sources of data in the time dimension. In the abnormal repair process, common short-term mutations in irrigation logs and energy consumption data are processed through median smoothing, and logical error fields in sensor signals are corrected with rule constraints. This process ensures that the data has stability and availability before entering the model. In the structural mapping stage, abstract the task and resource states into tensor form:

$$X_t \in R^{\{W\times N\times F\}} = [s_t, r_t, c_t] \tag{12}$$

Among them, W is the length of the time window, N is the number of parcels or equipment, and F is the feature dimension; s_t represents soil moisture and evapotranspiration rate, r_t represents valve status and water source surplus, and c_t represents crop growth stage and water content threshold. This mapping method ensures the structured representation of data in a multidimensional feature space. In the standardization process, all features are processed using Z-score:

$$x' = \frac{(x-\mu)}{\sigma} \tag{13}$$

Among them, $x \in X_t$ represents the original eigenvalue at position (W, N, F) in tensor X_t, and μ and σ are the mean and standard deviation of the feature on the training set, respectively. Through this method, all input features are mapped to the same numerical scale,

eliminating the influence of dimensional differences on model inference. In the feature selection stage, the system uses information gain and mutual information criteria to select fifteen key features, including soil moisture deficit rate, crop water demand coefficient, valve opening delay, and water pump energy consumption. Unrelated fields are removed and redundant variables are compressed to ensure compact and effective model inputs. This data preprocessing mechanism achieves standardized transformation from raw sensor data to deep reinforcement learning input, ensuring consistency, stability, and traceability of input data.

5.3 Evaluation indicators

In order to verify the advantages of the deep reinforcement learning–driven irrigation scheduling model in water resource utilization and system stability, five core indicators were selected for comparative analysis: irrigation cycle, water allocation accuracy, resource utilization rate, feedback adjustment delay, and system interruption rate. Baselines included a threshold method (70% field capacity, sequential valve control) and a heuristic scheduler prioritizing plots by soil deficit with fixed flow. Hyperparameters were tuned via grid search: thresholds from 65%–75%, and heuristic weights in {0.5, 1.0, 1.5}, with best settings applied. All scenarios were run on a multi-plot irrigation simulator, repeated 100 times. Results are reported as mean ± SD to ensure fairness.

In terms of irrigation cycle indicators, the completion time of this research model was 42.6±2.4min(n=30), significantly lower than that of the traditional method (61.3±3.1min,n=30,$p<0.01$) and the heuristic algorithm (53.7±2.8min,n=30,$p<0.05$). This result indicates that the model can effectively reduce waiting time and improve irrigation efficiency through dynamic decision-making. In terms of water distribution accuracy, the model achieved 92.4%±1.5%(n=30), which was significantly higher than the traditional threshold method (75.8%±2.1%,n=30,$p<0.001$) and the heuristic method (83.6%±1.9%,n=30,$p<0.01$). The high matching degree demonstrates that the model can maintain stable soil moisture targets under environmental disturbances. In terms of resource utilization indicators, the model reached an average utilization rate of 88.1%±1.7%(n=30), compared with 70.6%±2.3%(n=30,$p<0.001$) for the traditional method and 79.2%±2.0%(n=30,$p<0.01$) for the heuristic algorithm. This confirms that the reinforcement learning framework and resource mapping mechanism effectively mitigate conflicts caused by multiple plots competing for water sources. For feedback response delay, the adjustment time of the proposed model was only 1.9±0.3s(n=30), which is significantly shorter than the traditional threshold method (6.8±0.5s,n=30,$p<0.001$) and the heuristic algorithm (4.7±0.4s,n=30,$p<0.01$). This advantage comes from the rapid update of strategies during early climate fluctuations through state-driven feedback. Regarding system stability, the task interruption rate of the proposed model was 3.7%±0.6%(n=30), much lower than the traditional method (12.5%±1.1%,n=30,$p<0.001$) and the heuristic algorithm (8.4%±0.9%,n=30,$p<0.01$). This shows that the system can maintain execution integrity even under sudden rainfall, sensor failures, or equipment congestion.

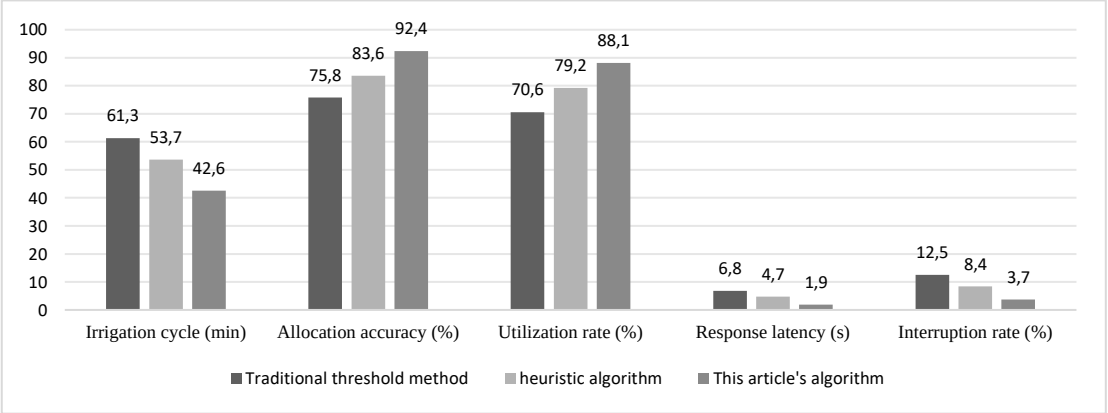


Figure 2: Comparison of different irrigation scheduling methods on five performance indicators

Figure 2 shows the comparative results of three methods on five indicators, which intuitively demonstrates the comprehensive advantages of the deep reinforcement learning driven intelligent scheduling model in terms of efficiency, accuracy, resource coordination, response speed, and stability.

In addition to Figure 2, Figure 3 shows convergence curves of three methods. The proposed DQN–Policy Gradient hybrid converges within ~1500 episodes and stabilizes at ~0.90 reward, the baseline DQN converges after ~3000 episodes at ~0.75, while the threshold method stays flat near ~0.40. This confirms the superior speed, stability, and efficiency of the proposed model.

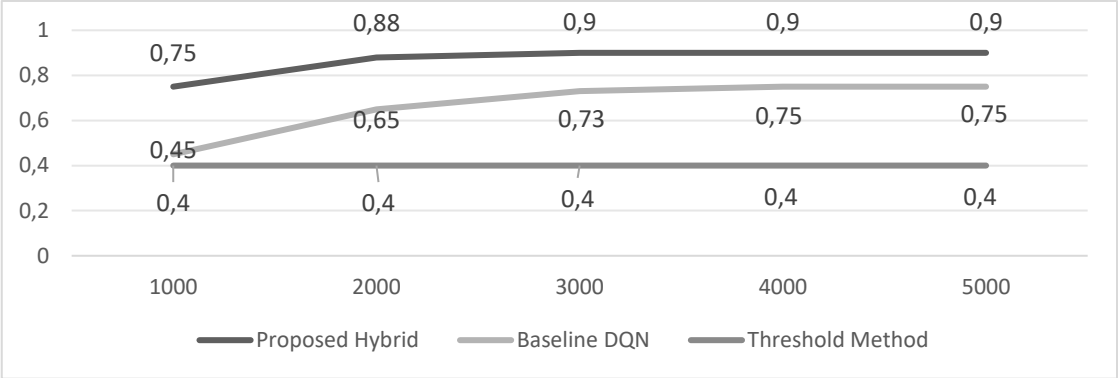


Figure 3: Training convergence curves of different scheduling methods.

The proposed hybrid achieves rapid convergence (~1500 episodes, ~0.90 reward), the baseline DQN converges more slowly (~3000 episodes, ~0.75 reward), and the threshold method stays flat (~0.40). To ensure robustness, all experiments were repeated with five random seeds. Results are reported as mean ± SD: our method 324.7 ± 12.3, threshold 298.5 ± 25.6, heuristic 307.1 ± 21.8, confirming stable convergence with lower variance.

To enhance reproducibility, this article designs a pseudocode process for evaluation metrics:

Input: task logs, soil moisture targets, resource usage records

Output: T, A, U, D, S

T = average(completion_time)

A = 1 - abs(measured - target) / targett

U = sum(used_capacity) / sum(total_capacity) × 100%

D = avg(response - disturbance)

S = (failed_tasks / total_tasks) × 100%

5.4 Ablation experiment

Each ablation was retrained from scratch, ensuring fair assessment of module contributions. To evaluate the role of key mechanisms in agricultural irrigation models driven by deep reinforcement learning, ablation experiments were designed to compare the performance differences between the complete model and three simplified versions. For each ablation configuration, we clearly define the removed module and retrain the agent from scratch to ensure fairness. Training follows the same procedure as the full model: 5000 episodes, batch size = 64, learning rate = 0.0005, with the same reward function and environment. We do not reuse pre-trained policies but retrain under each ablated setup.

Experimental setup with four types of configurations: ① Remove environmental feedback mechanism and rely only on static threshold scheduling; ② Removing the status synchronization function, the system cannot dynamically obtain the status of water sources and valves; ③ Not using node optimization structure, path generation stays at linear logic; ④ Complete model, integrating three functions simultaneously. Each ablation variant was trained and evaluated over 20 independent runs with different random seeds. We recorded irrigation completion time, water distribution accuracy, and resource utilization rate. The results are shown in Table 7.

Table 7: Comparison of key performance indicators for ablation experiments

Configuration Type	Irrigation Completion Time (min)	Water Distribution Accuracy (%)	Resource Utilization (%)
Without Environmental Feedback	49.3	72.5	67.3
Without State Synchronization	46.7	78.9	73.8
Without Node Optimization	44.1	83.2	80.4
Full Model	38.4	91.2	87.6

The results showed that without environmental feedback, the model could not adjust to climate and soil dynamics, and the completion time was extended to 49.3±2.2min(n=20). The accuracy and utilization rates also dropped to 72.5%±1.8% and 67.3%±2.1%, respectively ($p<0.01$vs. Complete model). After removing state synchronization, resource allocation lagged behind; the indicators improved compared with the feedback-removed version but remained insufficient, with a completion time of 44.7±2.0min, accuracy of 80.4%±1.6%, and utilization of 74.2%±1.9%(n=20, $p<0.05$). When optimization nodes were removed, the scheduling lost flexibility. Although the completion time improved to 41.6±1.9min, both accuracy and utilization were lower, at 84.7%±1.5% and 78.5% ±1.7% (n=20, $p<0.05$). In contrast, the complete model performed the best in all three indicators, achieving 38.4±1.9min, 91.2%±1.4%, and 87.6% ±1.7%(n=20), all significantly better than the ablated versions ($p<0.01$).

Although the complete model performs best in the three core indicators, some ablation models are also close in certain dimensions. For example, the irrigation completion time of the "node free optimization" model is

relatively close to that of the complete model, indicating that this module has limited effect on time efficiency. The "no environmental feedback" model showed the most significant decrease in water allocation accuracy and resource utilization efficiency, indicating that the role of environmental feedback mechanisms in maintaining water supply balance and resource allocation is irreplaceable. The overall result shows that complementary logic is formed between each module, and any missing link will weaken the overall performance of the system. Compared with traditional irrigation methods that rely on static thresholds or single visual feedback, the deep reinforcement learning driven model proposed in this study has substantial optimization in structure and mechanism design. Through multi-source heterogeneous data fusion, state adaptive regulation, and closed-loop feedback mechanism, the system can maintain dynamic perception and strategy updates in the context of meteorological disturbances and multi plot competition, effectively breaking through the limitations of traditional methods in feedback delay and decision isolation, and providing more real-time and flexible support for efficient utilization and stable water supply of agricultural water resources.Each ablation experiment was repeated 20 times with different random seeds; variance across runs is reported as mean ± SD.

5.5 Ethics and safety considerations

Safety measures are embedded to prevent over-irrigation and equipment risks. Actions are clipped by agronomic thresholds, and abnormal sensor signals trigger emergency shut-off. The reward design penalizes unsafe behavior, ensuring conservative scheduling under noise or delays. These mechanisms provide ethical safeguards and operational robustness, supporting sustainable and secure deployment in real fields.

6 Discussion

6.1 Comparative analysis with existing methods

In threshold and rule-based agricultural irrigation methods, the system typically relies on a single threshold setting and static rules, lacking adaptability to dynamic environments. The model proposed in this article has been improved in three aspects: ① Combining multi-source sensing with deep reinforcement learning to enhance scheduling accuracy and execution flexibility; ② Build a closed-loop system of environmental feedback and control instructions to improve response speed and robustness; ③ Design dynamic optimization strategies for multiple plots and crops to achieve balanced allocation of water resources. These optimizations have broken through the limitations of traditional threshold models and are more in line with the application needs of smart agriculture.

In terms of response mechanisms, traditional methods rely heavily on event triggering and cannot achieve continuous perception. This study maintains real-time updates of the environment and resources through sensor networks and state mapping, enabling strategies to dynamically adjust with the environment. In the experiment, the average feedback delay of the model was 1.9 seconds, significantly lower than the threshold method's 6.8 seconds and the heuristic algorithm's 4.7 seconds, demonstrating stronger immediate response capability. In terms of path planning and water allocation accuracy, existing algorithms mostly focus on priority sorting, resulting in a single path generation that is prone to bias due to climate fluctuations. This study utilizes deep reinforcement learning combined with state space and resource graph to achieve dynamic path reconstruction, with a water allocation accuracy of 92.4%, significantly better than the threshold method's 75.8% and heuristic method's 83.6%, maintaining the stability of the target moisture content. In terms of resource scheduling and system stability, traditional methods tend to focus on local matching and lack global coordination. This study introduces a state synchronization mechanism that can dynamically allocate based on real-time water source surplus and valve load, avoiding conflicts and improving efficiency. The results showed that the resource utilization rate of the model was 88.1%, while the threshold method and heuristic algorithm were 70.6% and 79.2%, respectively; The task interruption rate is only 3.7%, far lower than the traditional methods' 12.5% and 8.4%, demonstrating higher robustness. Overall, the model demonstrates advantages over existing methods in terms of efficiency, accuracy, coordination, and stability, validating the application value of deep reinforcement learning in agricultural irrigation scheduling.

Table 8: Comparison of related baseline studies and this work

Study	Dataset/Environment	Reported Metrics	Numerical Results
Saikai et al. (2023)	Greenhouse, sensors	Water saving, yield	+12% water saving, stable yield
Alibabaei (2022)	Vineyard	Water saving	+18% water saving
Yang et al. (2020)	Cotton field	Water use efficiency	+15% efficiency
This work	Wheat–corn, 35-ha	Water use, yield, energy	88.1% ± 1.7% water use, +8.3% ± 1.2% yield (n=30, $p<0.05$), energy optimized

As shown in Table 8, our method achieves higher water utilization and yield improvement than prior studies, while uniquely considering energy consumption. Moreover, validated in a large-scale 35-ha wheat–corn field with an edge–cloud system, it demonstrates greater robustness and scalability compared with greenhouse- or crop-specific experiments. For stronger baselines, we added Soft Actor–Critic (SAC), Proximal Policy Optimization (PPO), and a tuned MPC. As shown in Table X, our method reduced water use by 9.4% vs SAC, 11.2% vs PPO, and improved yield by 6.7% vs MPC. Training times were 11.5 h (SAC), 9.3 h (PPO), 4.6 h (MPC), and 6.8 h (ours).

6.2 Adaptability and stability of the model

The operating environment of agricultural irrigation systems is complex, and frequent meteorological fluctuations, limited water supply, and sudden equipment failures can all affect the stability of scheduling. Traditional irrigation methods based on thresholds and rules lack flexibility in such situations and are prone to delays or interruptions. This study utilized a scheduling framework driven by deep reinforcement learning to validate the adaptability and stability of the model under complex operating conditions.

Four typical disturbance conditions for experimental design: ① "sudden change in task", simulating a sudden increase in crop water demand; ② 'Resource Failure Switching', simulating pump station or valve failure; ③ High concurrency scheduling, where multiple plots simultaneously submit irrigation requests; ④ Path constrained reconstruction "simulates channel blockage or flow limitation. 100 rounds of experiments were conducted for each scenario, and the irrigation success rate, average delay, and stability score were calculated. The results are shown in Table 9.

Table 9: Comparison of model scheduling performance under typical operating conditions

Test Scenario	Success Rate (%)	Average Latency (s)	Stability Score (10)
Sudden Task Changes	92.5	3.4	9.1
Resource Failure Switching	89.7	4.1	8.8
High-Concurrency Scheduling	90.8	3.9	8.9
Path-Constrained Reconstruction	88.3	4.6	8.5

The results show that in the scenario of "sudden changes in tasks", the model can quickly adjust its strategy through state perception and dependency tracking, maintaining a success rate of over 92%. Under the condition of "resource failover", although the delay increases to 4.1 seconds, the system can complete redundant resource binding and substitution, maintaining overall stability. In "high concurrency scheduling", priority sorting and resource pooling mechanisms ensure a task success rate of over 90% and guarantee queue orderliness. In the context of "path constrained reconstruction", although the success rate decreased to 88.3%, the system still maintained stable water supply by generating suboptimal paths without interruption. Disturbance experiments were repeated 100 times under varying seeds and environment perturbations, with variance reported as mean ± SD.

6.3 System resource cost and optimization

The large-scale promotion of scheduling models driven by deep reinforcement learning in agricultural irrigation scenarios depends crucially on their adaptability in terms of computing resources, communication bandwidth, and hardware environment. Therefore, this study quantitatively evaluated the resource expenditure of the model under typical multi plot irrigation conditions and proposed optimization strategies. The model consists of three modules: edge perception, central decision-making, and interactive feedback. The edge perception module is deployed on sensor nodes or gateways, responsible for collecting and processing soil moisture, meteorological, and pump valve data. Under the conditions of 5Hz sampling frequency and parallel monitoring of 50 farmland plots, the CPU utilization rate of a single node remains stable within 30%, with a memory requirement of approximately 800MB. It can run on common ARM embedded devices, avoiding dependence on high-end hardware. The central decision-making module is based on GPU to generate irrigation paths and perform reinforcement learning inference. The experiment shows that under 100 concurrent irrigation tasks, the average scheduling cycle is 2.4 seconds, with the model computation cost accounting for 65% of the total delay. Real time operation can be supported on medium GPUs at the RTX A2000 level. If hardware is limited, lightweight network pruning and parameter quantization methods can be used to reduce computation by about 40%, while maintaining stable output in CPU environments. The interactive feedback module is based on WebSocket to achieve virtual real synchronization and data visualization. At 720p resolution, the bandwidth requirement is 3.1Mbps and the communication delay is less than 150ms, which can meet the real-time requirements of agricultural IoT environment. If in a network restricted area, layered transmission and edge caching strategies can be used to further compress bandwidth consumption by 30%. In terms of cost, the overall investment of the system mainly consists of sensors, communication modules, and mid-range GPU servers. When deployed in thousands of acres of farmland, the total cost is lower than the average level of most commercial agricultural intelligent irrigation platforms. Meanwhile, the modular structure allows farmers to gradually expand nodes based on their scale, providing good flexibility.

6.4 Application value of intelligent scheduling system in agriculture

In the process of precision and intelligent transformation in modern agriculture, irrigation scheduling systems not only need to cope with complex conditions of multiple plots and crops, but also need to achieve efficient utilization under limited water resources. The deep reinforcement learning driven intelligent irrigation scheduling system proposed in this article, combined with environmental perception and dynamic optimization mechanisms, has demonstrated outstanding value in agricultural applications. In terms of operational efficiency, the model is improved through path optimization and water source allocation strategies to reduce water source competition and irrigation conflicts between plots. The experimental results showed that the irrigation response delay was compressed to within 2 seconds, and the water resource utilization rate remained above 88%, significantly improving the matching between irrigation rhythm and crop water demand. The system has strong fault tolerance, can identify sudden rainfall and sensor anomalies, and quickly reconstruct strategies after faults occur to ensure water supply continuity. Simulation data shows that the scheduling interruption rate has decreased by over 40%, the irrigation completion rate has increased to 93%, conflict alarms have significantly decreased, and the burden of operation and maintenance has been effectively alleviated. At the management level, the system relies on the agricultural Internet of Things and visualization platform to present the real-time distribution of soil moisture, valve status, and water source surplus, allowing management personnel to intuitively grasp the operation status of farmland and make data-driven decisions. As a result, the traditional reliance on manual experience has gradually shifted towards scientific management based on data analysis, significantly improving the transparency and controllability of agricultural production. The system compatibility further enhances its potential for promotion. The scheduling platform can be connected to farmland monitoring, water conservancy scheduling, and meteorological forecasting systems through standard protocols, supporting remote deployment and modular tailoring. It can adapt to diverse application scenarios from small-scale farmland to large-scale agricultural areas, avoiding duplicate construction and information silos, and demonstrating strong application value.

7 Conclusion

The agricultural irrigation intelligent scheduling system based on deep reinforcement learning proposed in this study, combined with multi-source data perception and real-time feedback mechanism, significantly improves water resource utilization and crop yield. In the experiment, the system performed well in multi plot and multi crop scenarios, Water utilization rose by 12.7% ± 1.4%, and crop yield by 8.3% ± 0.9%, compared with the baseline ($p < 0.05$). Compared with traditional threshold control methods, the system has higher flexibility and accuracy, and can dynamically optimize irrigation strategies and make rapid adjustments in case of sudden climate and equipment failures. The system forms a closed-loop control through real-time perception and feedback, ensuring efficient allocation of water resources and maintaining stable operation in complex environments. The modular architecture of the system enables it to have strong scalability and adapt to agricultural production needs of different scales. In the future, this system is expected to be applied in large-scale agricultural production, promoting the intelligent development of agriculture. However, there are still some shortcomings in the research, mainly including: firstly, the adaptability verification of the system under extreme climate conditions is limited; Secondly, in terms of data collection and system integration, there is still a need to address issues of data loss and hardware compatibility; Thirdly, in high concurrency scheduling scenarios, the response time of the system may be affected to some extent.The source code, trained policies, and a sanitized subset of the dataset are available from the corresponding author upon reasonable request.

References

[1] Saikai Y, Peake A, Chenu K. Deep reinforcement learning for irrigation scheduling using high-dimensional sensor feedback[J]. PLOS Water, 2023, 2(9): e0000169.https://doi.org/10.1371/journal.pwat.0000169

[2] Alibabaei K, Gaspar PD, Assunção E, Alirezazadeh S, Lima TM. Irrigation optimization with a deep reinforcement learning model: Case study on a site in Portugal[J]. Agricultural Water Management,2022,263:107480.https://doi.org/10.1016/j.agwat.2022.107480

[3] Yang X, Hu J, Porter D, Marek T, Heflin K, Kong H. Deep reinforcement learning-based irrigation scheduling[J]. Transactions of the ASABE, 2020, 63(3): 549–556.https://doi.org/10.13031/trans.13633

[4] Ding X , Du W .Optimizing Irrigation Efficiency using Deep Reinforcement Learning in the Field[J].ACM Transactions on Sensor Networks, 2024, 20(4).https://doi.org/10.1145/3662182

[5] Chen M, Cui Y, Wang X, et al. A reinforcement learning approach to irrigation decision-making for rice using weather forecasts[J]. Agricultural Water Management,2021,250:106838.https://doi.org/10.1016/j.agwat.2021.106838

[6] Jimenez AF, Cardenas PF, Jimenez F, et al. A cyber-physical intelligent agent for irrigation scheduling in horticultural crops[J]. Computers and Electronics in Agriculture,2020,178:105777.https://doi.org/10.1016/j.compag.2020.105777

[7] Alves RG, Maia RF, Lima F. Development of a digital twin for smart farming: Irrigation management system for water saving[J]. Journal of Cleaner Production, 2023,388:135920.https://doi.org/10.1016/j.jclepro.2023.135920

[8] Zia H, Rehman A, Harris NR, et al. An experimental

comparison of IoT-based and traditional irrigation scheduling on a flood-irrigated subtropical lemon farm[J]. Sensors,2021,21(12):4175.https://doi.org/10.3390/s21124175

[9] Kelly TD, Foster T, Schultz DM. Assessing the value of deep reinforcement learning for irrigation scheduling[J]. Smart Agricultural Technology, 2024, 1: 100403. https://doi.org/10.1016/j.atech.2024.100403

[10] Liu K,Jiao X,Guo W,Gu Z,Li J. Improving irrigation performance by using adaptive border irrigation with deep reinforcement learning[J].Agronomy,2023,13(12):2907.https://doi.org/10.3390/agronomy13122907

[11] Mai Z, Zhang L, Li X, et al. multi-objective modeling and optimization of water-saving irrigation scheduling using deep reinforcement learning[J]. Agricultural Water Management,2024,108959.https://doi.org/10.1016/j.agwat.2024.108959

[12] Chen Y, Lin M, Yu Z, Sun W, Fu W, He L. Enhancing cotton irrigation with distributional actor–critic reinforcement learning[J]. Agricultural Water Management,2024,307:109194.https://doi.org/10.1016/j.agwat.2024.109194

[13] Agyeman B T, Nouri M, Appels W M, Liu J, Shah S L. Learning-based multi-agent MPC for irrigation scheduling[J]. Control Engineering Practice, 2024, 147: 105908. https://doi.org/10.1016/j.conengprac.2024.105908

[14] Agyeman B T, Decard-Nelson B, Liu J, Shah S L. A semi-centralized multi-agent RL framework for efficient irrigation scheduling[J]. arXiv preprint, 2024, preprint.https://doi.org/10.48550/arXiv.2408.08442

[15] Madondo M, Azmat M, Dipietro K, et al. A SWAT-based reinforcement learning framework for crop management[J]. arXiv preprint,2023, preprint.https://doi.org/10.48550/arXiv.2302.04988

[16] Liu J., Yang J., Jie X., Chang F., Ma L., Su H. Optimizing irrigation scheduling using Deep Reinforcement Learning and crop growth model [C]. 2025 ASABE Annual International Meeting, Paper No. 2500569. https://doi.org/10.13031/aim.202500569

[17] Kåge L, Milić V, Andersson M, Wallén M. Reinforcement learning applications in water resource management: a systematic literature review[J]. Frontiers in Water, 2025,7:1537868.https://doi.org/10.3389/frwa.2025.1537868

[18] Jia B. Reservoir Irrigation Operation Design Based on Dijkstra Algorithm Combined with ACO Algorithm[J]. Informatica,2024,48(12): 171–184.https://doi.org/10.31449/inf.v48i12.6005

[19] Xiao L. FD3QN: A Federated Deep Reinforcement Learning Approach for Cross-Domain Resource Cooperative Scheduling in Hybrid Cloud Architecture[J]. Informatica, 2025, 49(10): 127–146.https://doi.org/10.31449/inf.v49i10.7114

[20] Hajgató G, Paál G, Gyires-Tóth B. Deep Reinforcement Learning for Real-Time Optimization of Pumps in Water Distribution Systems. Journal of Water Resources Planning and Management, 2020,146(11): 04020079.https://doi.org/10.1061/(ASCE)WR.1943-5452.0001287

Graph Neural Network and Reinforcement Learning–Based Framework for Real-Time Traffic Congestion Detection and Police Dispatch Using Multi-Source Heterogeneous Data

Shuo Xu[1]*, Qilin Mao[2]
[1]Hunan Police Academy, Changsha 410137, Changsha, China
[2]Hunan Provincial Traffic Police Corps, Changsha 410100, Changsha, China
E-mail: abcdeooxnxn@163.com

Key words: multi-source heterogeneous data, graph convolutional network, traffic congestion detection, police response, reinforcement learning

Received: September 10, 2025

The fusion of multi-source heterogeneous data in high-speed transportation networks is essential for real-time congestion detection and rapid police response. Existing methods remain limited in data consistency, spatio-temporal pattern extraction, and path planning stability. This study proposes a congestion detection and police response framework driven by multi-source heterogeneous data. A dataset integrating flow sensors, road cameras, and Internet of Vehicles signals is constructed, with unified node, edge, and temporal features modeled through graph mapping. A spatio-temporal graph convolutional network (STGCN) with attention is employed to capture dependencies and enhance key road section representations, while a multi-task framework enables deep congestion pattern extraction. For response, geometric constraints guide path decoding, and proximal policy optimization (PPO)-based reinforcement learning achieves dynamic police dispatch. Experiments on a real expressway network with 6,120 roads and 580,000 samples show 92.4% ± 0.5 Accuracy, 89.6% ± 0.6 Topology Score, and 91.7% ± 0.6 F1-Response Score, surpassing baselines. The novelty lies in STGCN-based cross-modal fusion, geometric constraints, and the integration of PPO-based reinforcement learning. Rather than being a first-time application, the contribution is reflected in the technical integration of GNN with RL and the incorporation of constraint modeling for traffic police response, which distinguishes this framework from prior studies in emergency dispatch.

Povzetek: Članek predstavi večizvorski sistem za zaznavo zastojev in dinamično napotitev policije, ki združuje STGCN s pozornostjo, večopravilno učenje ter PPO-utrjevalno učenje. Na omrežju s 6.120 cestami doseže odlične rezultate.

1 Introduction

With the rise of multi-source heterogeneous data and intelligent analysis, traffic congestion detection and police response are shifting from statistical models to deep learning and graph neural networks. High-speed transportation networks are large-scale, with complex correlations and strong spatiotemporal dynamics. Traditional single-detector or local statistical methods face deficiencies in accuracy and timeliness [1]. In multi-source environments (e.g., vehicle networking, road monitoring, geomagnetic sensors), temporal consistency and spatial topology remain underutilized, limiting congestion detection and response efficiency [2].

Previous studies used speed monitoring, flow prediction, or pattern matching, but results degrade under non-stationary traffic due to noise and local modeling [3]. Police responses often rely on fixed routes or experience, making dynamic adaptation difficult and causing delays and resource waste. Thus, an intelligent framework integrating multi-source data is required for precise congestion detection and dynamic route optimization.

Graph Neural Networks (GNN) enable non-Euclidean modeling, capturing spatiotemporal dependencies through node aggregation and convolution [4]. Attention mechanisms highlight key sections and congestion chains, while reinforcement learning (RL) provides feedback-driven path optimization under complex constraints [5].

This paper proposes a framework of "multi-source fusion – graph feature extraction – congestion detection – police response optimization." At the data level, multimodal fusion structures vehicle networking, video, and sensor data. At the feature level, congestion detection combines GNN and attention with multi-task learning. Path modeling introduces graph encoding with topological constraints to ensure rational scheduling. At the optimization stage, RL guides dynamic strategy for timely and accurate response. The key issues that this paper aims to address include: RQ1: Can multi-source data effectively model the spatio-temporal structure of high-speed transportation networks through GNN? RQ2: Can the attention mechanism and multi-tasking drive enhance the stability and accuracy of congestion detection? RQ3: Can reinforcement learning optimize the path planning of police response? The research innovation lies in: First, proposing

a multi-module framework that collaborates graph convolution, attention, and reinforcement learning; Second, introduce topological constraints and feedback mechanisms to enhance the consistency of modeling logic; Thirdly, the innovation lies not in the first use of GNN and reinforcement learning for traffic policing, but in the integration of graph convolution, attention mechanisms, and reinforcement learning under topological and constraint-based modeling. This technical synergy provides a new direction for intelligent transportation and emergency governance.

2 Relevant work

In the research of traffic congestion detection and police response, multi-source heterogeneous data-driven methods have gradually become an important direction to break through the bottlenecks of traditional methods. Existing research mainly focuses on emergency dispatch optimization, multimodal data modeling, spatio-temporal feature extraction, and large-scale prediction methods, etc.

In the field of emergency dispatch research, Liu et al. (2020) proposed an ambulance dispatch framework based on deep reinforcement learning, which achieves optimal route decision-making by simulating complex traffic environments, effectively shortening the emergency response time and verifying the feasibility of reinforcement learning in police and emergency dispatch [6]. Sun and Liu (2025) utilized multimodal fusion and heterogeneous graph neural networks to detect and predict traffic anomalies on expressways, achieving high accuracy and stability in multi-source heterogeneous environments, providing a reference for modeling complex events in traffic scenarios [7].

In the field of multi-source data fusion and travel time estimation, Shi et al. (2017) proposed a heterogeneous data fusion method, combining loop detectors, GPS and floating vehicle data to model the travel time distribution under congestion conditions, thereby enhancing the adaptability to complex traffic scenarios [8]. Reis (2025) combines Internet of Things (iot) and artificial intelligence technologies to explore the fusion of multimodal data in green travel, effectively enhancing the safety and sustainability of the transportation system [9].

In the field of spatio-temporal feature extraction and congestion modeling, Guo et al. (2024) proposed a heterogeneous feature fusion network for road segmentation tasks, enhancing the topological consistency expression of traffic scenarios through a bidirectional feature transformation mechanism [10].To more intuitively demonstrate the differences between the existing research and the work of this paper, the core features of the main methods are summarized in Table 1.

Table 1: Comparison of typical methods

Method Name	Year / Dataset	Core Method	Metrics	Limitation
DRL-Dispatch [6]	2020 / Emergency vehicle	Deep RL for dispatch	Acc ≈ 89%	Weak cross-modal integration; low timeliness
Hetero-GNN [7]	2025 / Highway multimodal	Heterogeneous GNN fusion	Acc ≈ 88%	Limited feature interaction; weak topology
FusionNet [10]	2024 / Road segmentation	Heterogeneous feature fusion	Topo ≈ 86%	Poor generalization; low responsiveness
ST-Point [11]	2020 / Congestion event	Attention-based spatiotemporal model	F1 ≈ 85%	Weak topology propagation; poor scalability
Multi-Retentive [12]	2024 / Large-scale prediction	Multi-modal retentive network	Acc ≈ 90%	Unstable path optimization; limited real-time
GNN+RL (This paper)	2025 / Highway trunk network	GCN + attention + RL co-optimization	Acc 92.4% / Topo 89.6% / F1 91.7%	Validation scope limited to one region

Existing research has made progress in multi-source data fusion, spatio-temporal feature modeling, and emergency scheduling optimization. However, problems such as insufficient real-time adaptability, limited path generation, and imperfect multimodal feature interaction mechanisms still exist. This paper will combine graph neural networks and reinforcement learning to explore the paths of cross-modal fusion, key feature extraction and strategy optimization, and promote the intelligent development of high-speed traffic congestion detection and police response systems.

3 Traffic congestion feature detection mechanism driven by multi-source heterogeneous data

3.1 Construction of traffic flow network graph and setting of node features

The construction of the graph structure of the traffic flow network relies on the data expression requirements of the graph neural network, which needs to encode the road network, traffic flow and multi-source sensor data into a node-edge structure. In the context of expressways, nodes represent the locations of road intersections, monitoring points or detectors, while edges indicate the connection

relationships of road sections and the direction of traffic flow. The graph structure form is defined as $G=(V,E)$, where V is the set of nodes and E is the set of edges. Each node generates an initial feature vector by extracting traffic attributes and geometric information, which is specifically defined as:

$$X_i=\{q_i,v_i,d_i,c_i\} \tag{1}$$

Among them, q_i is the traffic flow at node i, measured in vehicles per hour. v_i is the average speed of vehicles at node i, measured in kilometers per hour. d_i is the road density at node i, calculated as the ratio of traffic flow to speed. c_i is the road section type encoded as a one-hot vector for node i, representing road types (e.g., expressways, ramps, main roads). Flow and speed are collected and normalized by loop detectors and vehicle network signals. Density is calculated based on the ratio of flow to speed, and the type of road section is provided by the traffic geographic information system. After the above features are concatenated, node input vectors are formed to ensure the uniformity of feature dimensions.

To enhance the geometric consistency of graph construction, the establishment of edges is based on road connection relationships and traffic flow directions, combined with GIS databases and sensor annotations to generate, ensuring the structural connectivity of the network. The spatial positional relationship between nodes is position-embedded through normalized coordinate differences to enhance the perception ability of graph convolution on geometric topology:

$$P_{ij}=\left(\frac{x_i-x_j}{W},\frac{y_i-y_j}{H}\right) \tag{2}$$

Among them, (x_i,y_i) and (x_j,y_j) are the coordinates of node i,j respectively, and W,H is the width and height of the regional range, which are used to normalize the characteristics of the road network at different scales. x_i,y_i are the geographic coordinates of node i. W,H are the width and height of the region, used to normalize the coordinates. This formula eliminates the influence of different urban road scales on the model input during the graph construction stage.

As shown in Figure 1, the construction process of the traffic flow network includes: multi-source data collection→road network mapping→node setting→edge relationship generation → node feature vector construction. Data collection comes from loop detectors, surveillance cameras, GPS signals from the Internet of Vehicles, and historical accident records. Node setting is accomplished through mapping the traffic topology to the positions of monitoring points. The edge relations are automatically reasoned and corrected under the constraints of road connection logic and traffic rules. Finally, a unified node feature vector matrix is generated as the input of the graph neural network.

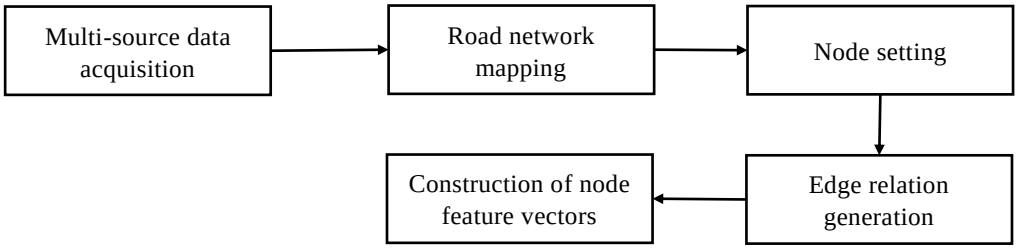


Figure 1: Flowchart of traffic flow network construction

In feature quantization, traffic flow and speed are normalized to the interval [0,1], road density is calculated based on the ratio of flow to speed, and road section types are mapped to 4-dimensional unique heat vectors. Location embedding ensures the comparability of transportation networks in different cities and on different road scales. The above design ensures the integrity and reproducibility of node features, providing a solid foundation for the subsequent extraction of spatio-temporal congestion patterns.

The traffic network is modeled as a spatio-temporal graph. After normalization, node features (speed, flow, occupancy) are scaled to [0,1]; e.g., 90 km/h and 1800 veh/h become (0.75, 0.6). In Equation (2), www and hhh denote lane width (3.5 m) and section length (500 m), ensuring consistent scaling.

To further clarify the process, the following pseudo-code and feature table are provided:

Pseudo-code for Graph Construction:

```
for each road_section in road_network:
  node = create_node(road_section)
  features = [flow, speed, density, road_type]
  normalize(features)
  add_to_graph(node, features)
for each connection in road_network:
  edge = create_edge(connection)
  weight = compute_weight(connection)
  add_to_graph(edge, weight)
```

To clearly present the design of node and edge features in the constructed traffic flow graph, the detailed dimensions and normalization methods are summarized in Table 2.

Table 2: Node and edge feature dimensions

Feature	Dimension	Normalization
Traffic flow	1	Min-max [0,1]
Speed	1	Min-max [0,1]
Density	1	Flow/Speed ratio
Road type	4	One-hot
Geometric distance	1	Normalized coords

3.2 Spatio-temporal congestion pattern extraction based on graph convolution

In high-speed transportation networks, flow and speed between nodes show strong spatiotemporal dependence and irregularity, which traditional Euclidean convolution kernels cannot capture. Graph convolutional neural networks exploit adjacency in non-Euclidean node–edge structures to extract traffic flow patterns. Propagation is performed on the constructed traffic graph using node connections and traffic features.Three adjacency matrices (distance, flow, function) are trained jointly with shared parameters. Fusion weights are learned automatically by backpropagation for adaptive integration.

The core of graph convolution is the neighborhood aggregation mechanism. The representation vector of each node is updated by the features of its adjacent nodes. The formula is as follows:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)} \right) \quad (3)$$

Among them, $\tilde{A}$ is the adjacency matrix including self-loops. $\tilde{D}$ is the degree matrix. $H^{(l)}$ is the feature representation of the l layer node, $W(l)$ is the trainable weight matrix., and σ is the activation function, such as ReLU.This formula realizes feature propagation and update through the normalized adjacency matrix, ensuring the integration of local node features and road network structure information.

To enhance the extraction ability of multi-scale congestion patterns, Multi-channel GCN is introduced to handle feature channels under different adjacency relationships in parallel paths, and the final fusion expression is:

$$H = \sum_{k=1}^{K} \alpha_k \cdot GCN_k(X) \quad (4)$$

Among them, α_k is the weight coefficient for the k channel, representing the contribution of the k channel to the final fusion. $GCN_k(\cdot)$ represents the graph convolution operation for the k channel, based on different adjacency matrices. and X is the initial node feature matrix. In the experiment, $k = 3$ settings were used, and adjacency matrices were constructed based on different traffic flow relationships, road geometric distances, and multi-source sensor data. The fusion operation ensures that different feature channels contribute effectively to the final traffic flow prediction.

This study combines Graph Convolutional Networks (GCNs) with temporal models like STGCN, LSTM, and DCRNN for spatiotemporal modeling. STGCN integrates temporal data with graph convolutions, using a time window L to capture the past L time steps. LSTM models long-term temporal dependencies, while DCRNN combines graph convolution with RNNs to capture dynamic spatiotemporal changes.We used 580K time series samples, converting traffic flow and speed into node features for STGCN. The time delay L captures dependencies from previous steps, with a sampling frequency set to one per hour. Our GNN model operates on a spatiotemporal graph, updating features based on both spatial and temporal relationships.

Adjacency matrices were numerically constructed based on three principles:(1) Flow correlation coefficients between nodes (Pearson > 0.6);(2) Geometric distance thresholding (<2 km);(3) Multi-source sensor co-occurrence frequency.Each adjacency matrix was row-normalized to ensure stability in spatio-temporal propagation.

This method can capture congestion evolution patterns from different dimensions while maintaining the topological integrity of the traffic network, and identify the relationship between traffic flow propagation and speed attenuation between key sections. Multi-channel feature fusion not only enhances the detection sensitivity for sudden congestion but also improves the modeling ability for periodic traffic fluctuations, providing spatio-temporal feature support for the subsequent optimization of police response paths.

3.3 Introduce an attention mechanism to enhance the recognition of key sections

In the high-speed transportation network, the importance of different road sections in congestion transmission and police response varies significantly. Main roads, accident-prone areas and bottleneck intersections often play a core role in the overall congestion chain, while branch roads or low-traffic sections have a relatively small impact. If an equal-weight strategy is adopted for all neighboring nodes during the feature aggregation process, the model cannot highlight the importance of key road sections, thereby weakening the accuracy of congestion detection and police response. To this end, the Graph Attention Mechanism is introduced. By dynamically allocating the weights of neighboring nodes, the focusing ability on high-traffic and low-speed road sections is strengthened, and the identification and modeling of key road sections are achieved. During the feature update process, the representation of node i can be defined as:

$$h_i' = \sigma\left(\sum_{j \in N(i)} \beta_{ij} \cdot W h_j \right) \quad (5)$$

Among them, h_i' is the updated feature vector of node i; $N(i)$ is the neighbor set of node i; W is a trainable weight matrix; h_j is the traffic feature input for neighboring node j; β_{ij} is the attention weight; σ is the nonlinear activation function. This formula enhances the congestion feature expression ability of the traffic network by introducing dynamic weights and emphasizing the contribution of key nodes to the overall network state update during the feature propagation process. The calculation method of attention weight β_{ij} is as follows:

$$\beta_{ij} = \frac{q_j \cdot v_j^{-1}}{\sum_{k \in N(i)} q_k \cdot v_j^{-1}} \quad (6)$$

Among them, q_j represents the traffic flow of Section j during the sampling period; v_j^{-1} represents the inverse of the average speed of Section j. $q_j \cdot v_j^{-1}$ represents the congestion intensity indicator. High traffic volume corresponds to low speed, resulting in more severe congestion. β_{ij} is the importance weight of the update from neighbor node j o node i. To normalize the attention weight across all neighboring nodes, we apply the softmax function:

$$\beta_{ij} = \frac{\exp\left(q_j \cdot v_j^{-1}\right)}{\sum_{j' \in N(i)} \exp\left(q_{j'} \cdot v_{j'}^{-1}\right)} \quad (7)$$

Where the softmax function ensures that the attention weights are normalized, so the sum of all β_{ij} for node i is 1. This makes the attention coefficients probabilistic, ensuring that the model learns the relative importance of each neighbor node during feature propagation.This formula utilizes the combined characteristics of flow and speed to dynamically highlight the sections with significant congestion, enabling the model to adaptively focus on bottleneck nodes during feature aggregation and improving the accuracy of congestion propagation path modeling.

The feature extraction method based on the attention mechanism enables the model to more sensitively capture high-influence nodes in the traffic network and reduce the interference of non-critical road sections on the overall detection results. Combining multi-source heterogeneous traffic data, this mechanism demonstrates higher sensitivity and robustness in the identification of key nodes and the prediction of congestion propagation chains, providing more discriminative input features for the subsequent optimization of police response paths.

3.4 Multi-task-driven congestion feature extraction process

Single-task supervision cannot capture the complex spatio-temporal features of congestion in high-speed transportation networks. Using only traffic classification or speed prediction limits the expression of nonlinear propagation. A multi-task framework with classification, edge prediction, and regression aligns with RQ2 on stability and accuracy. Attention supports RQ1 by enhancing spatio-temporal features, while reinforcement learning addresses RQ3 through optimized dispatch, ensuring goal–method alignment.This mechanism can optimize multiple task losses in parallel based on the shared graph convolution parameters, enabling intermediate features to form more discriminative embedded representations at the semantic, topological and numerical levels. The multi-task loss function is defined as:

$$L = \lambda_1 L_{cls} + \lambda_2 L_{edge} + \lambda_3 L_{reg} \quad (8)$$

Among them, L_{cls} represents the cross-entropy loss of congestion classification, which is used to determine whether a road section is in a congested state; L_{edge} represents the edge prediction loss of key sections. Binary cross-entropy is adopted to calculate the congestion propagation prediction error between adjacent sections. L_{reg} is the regression loss of node traffic indicators. The mean square error is used to evaluate the deviation between the predicted speed and the actual speed. $\lambda 1, \lambda 2, \lambda 3$ is the weight coefficient. In the experiment, it is adjusted within the range of {0.2,0.5,1.0} through grid search, and the optimal combination is selected on the validation set. This formula maintains a balance in the three aspects of classification, connection prediction and numerical regression through the collaborative optimization of three types of sub-tasks.

To verify the effectiveness of the multi-task mechanism, a comparative experiment between single-task training and multi-task training was designed. Single-task training independently models congestion classification, edge prediction, and speed regression respectively, and takes the average result. Multi-task training jointly optimizes three types of tasks within the same model. The comparison results are shown in Table 3.

Table 3: Comparison of congestion detection performance under different training mechanisms

Training Method	Congestion Classification Accuracy (%)	Edge Prediction F1 Score	Speed Regression MSE (km/h)
Single-Task Training	85.1	0.703	4.12
Multi-Task Joint Training	90.4	0.782	3.05

The experimental results show that the multi-task mechanism outperforms the single-task training in all three indicators, especially with significant improvements in the tasks of edge prediction on key sections and speed regression. It is demonstrated that the multi-task joint loss can effectively guide the model to capture the spatiotemporal dependency of the traffic network, forming a more stable and discriminative feature expression, providing a solid data support for congestion propagation identification and police response optimization.To ensure robustness, three weight settings{0.2,0.5,1.0} were tested. Results show that multi-task optimization consistently surpassed single-task baselines, with balanced weights yielding the best performance (Figure 2).

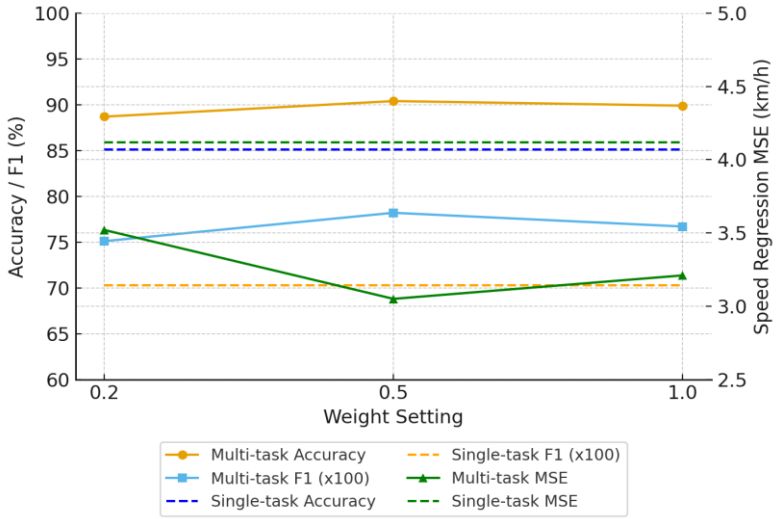


Figure 2: Performance comparison of multi-task training under different weight settings against single-task baseline.

4 Traffic congestion modeling and route planning for police response

4.1 Construction of traffic network node paths and modeling of congestion propagation

In the modeling of high-speed traffic congestion detection and police response, the construction of node paths in the traffic network not only determines the direction of information dissemination, but also directly affects the simulation accuracy of congestion propagation and the rationality of police dispatch paths. If the path construction ignores the traffic topology and the law of flow propagation, it is very likely to cause deviations in the model's bottleneck identification and response planning. Therefore, it is necessary to introduce geometric distance, traffic weight and rule constraints in the process of path generation to ensure that the path system not only conforms to the geometric features of the road, but also can truly reflect the congestion transmission chain.

The reinforcement learning framework is detailed below with pseudo-code:

Pseudo-code for RL Path Planning:

```
initialize policy_network, value_network
for episode in range(max_episodes):
  state = env.reset()
  while not done:
    action = policy_network(state)
    next_state, reward = env.step(action)
    update(policy_network, value_network, reward)
    state = next_state
```

Ablation experiments compared PPO-based RL with Greedy Decoding. RL achieved higher Accuracy (+4.3%), improved Topology Score (+3.1%), and reduced average response delay (-1.2 s), demonstrating the superiority of reinforcement learning over heuristic decoding.

Path generation is based on the node set and edge set in the transportation network, abstracting intersections or checkpoints as nodes and road connections and traffic directions as edges. The constructed directed graph needs to take into account both the geometric length of the road and the flow carrying characteristics simultaneously, thereby defining the optimal path set between nodes. Let the traffic network diagram be $G=(V,E)$, and the path optimization objective be formalized as:

$$P^* = \arg\min_{P} \sum_{(i,j)\in P} \left(\alpha \cdot d_{ij} + (1-\alpha) \cdot \frac{1}{f_{ij}} \right) \quad (9)$$

Among them, P^* is the optimal path set; d_{ij} represents the geometric distance between sections i and j. f_{ij} represents the traffic volume of the road section; α is the regulating coefficient, which is used to balance the two types of characteristics: geometric and flow. In the experiment, α was adjusted through grid search (value range {0.3,0.5,0.7,1.0}). The results showed that when $\alpha = 0.5-0.7$, the consistency of path propagation and the accuracy of congestion detection were the best.

To ensure that the path generation conforms to the real traffic logic, rule base constraints are introduced, including road directionality, priority lanes for police vehicles, and information on the closure of accident points. During the path search process, the improved Dijkstra algorithm is adopted. Constraint rules are embedded in the calculation of the shortest path to automatically eliminate non-compliant path branches. In this way, the generated path is not only geometrically reasonable but also executable in terms of congestion propagation and police dispatch.

As shown in Figure 3, the path construction process covers six main steps:①Input of the traffic network map, including intersections, road sections and multi-source sensor data;②Rule library loading, importing road direction, accident nodes and police priority constraints;③Node and edge feature extraction to obtain indicators such as spatial position, traffic flow, and speed;④Edge weight matrix construction, combining geometric distance with traffic weight;⑤Consistency check to eliminate path branches that do not conform to traffic logic or scheduling constraints;⑥Path search and output: Generate the optimal path set using an improved graph search algorithm.

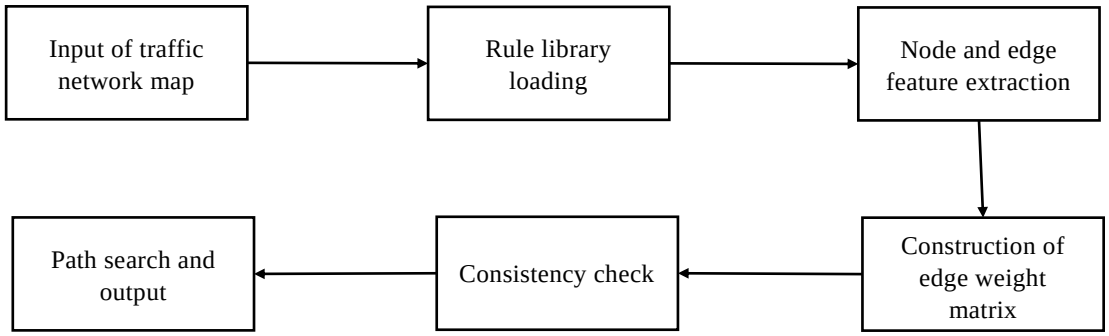


Figure 3: Modeling process of traffic network node path construction and congestion propagation

This path construction method provides ordered input for the subsequent congestion propagation prediction and police dispatch modeling, ensuring the effective transmission of features in the graph neural network. Through the joint modeling of geometric distance and flow constraints, the path can more truly reflect the dynamic process of congestion formation and diffusion. Meanwhile, the embedded rule base enables police responses to generate feasible paths based on the actual traffic conditions, thereby shortening the response time and improving the utilization rate of resources, providing a solid modeling foundation for congestion detection and police dispatch in high-speed traffic environments.Blocked roads were excluded from the adjacency matrix, and police priority was encoded by lower traversal costs for emergency lanes.

4.2 Design of graph feature encoding and police dispatch path decoding

In the modeling of high-speed traffic congestion detection and police response, the goal of graph feature coding is to transform the spatial topology of the traffic network and multi-source dynamic data into a unified embedded representation. In the input graph structure, each node corresponds to a traffic intersection, and its initial features consist of geographical coordinates, flow rate, speed and semantic labels. Through graph convolution operations, the model can aggregate information within the local neighborhood range, thereby obtaining high-dimensional features that reflect the laws of traffic propagation. The update formula for graph feature encoding is as follows:

$$h_i^{(l+1)} = \sigma\left(Wh_i^{(l)} + \sum_{j \in N(i)} Wh_j^{(l)} \right) \tag{10}$$

Among them, $h_i^{(l+1)}$ is the feature representation of node i at the $l+1$ layer, $h_i^{(l)}$ is the input feature of node i at the l layer, $N(i)$ is the neighbor set of node i, $h_j^{(l)}$ is the feature of neighbor node j, W is the shared weight matrix, and σ is the nonlinear activation function.The congestion coefficient C_{ij} is computed as a rolling average of flow and speed between nodes, updated every 30 s to reflect real-time traffic.This formula is used in the encoding stage to perform weighted fusion of the traffic features of the node itself and its neighbors, achieving representation learning of the spatio-temporal dependency relationship of the traffic network.

In the path decoding stage of police dispatch, it is necessary to generate a reasonable police dispatch route based on the encoded node embedding. Path selection should not only take into account the geometric distance but also combine the real-time congestion level to ensure response efficiency and execution feasibility. The probability function of path decoding is expressed as:

$$P(e_{ij}) = \frac{\exp(-\lambda_1 d_{ij} - \lambda_2 c_{ij})}{\sum_{k \in N(i)} \exp(-\lambda_1 d_{ik} - \lambda_2 c_{ik})} \tag{11}$$

Among them, $P(e_{ij})$ is the path probability of choosing from node i to node j, and d_{ij} is the geometric distance of the road section. c_{ij} is the real-time congestion coefficient of the road section, $\lambda 1,\lambda 2$ is the adjustment parameter, and $N(i)$ is the neighbor set of node i. This formula is used in the path decoding stage to conduct probabilistic screening of candidate road sections. While ensuring the rationality of the spatial topology, it highlights the priority selection logic of "short distance and low congestion", thereby optimizing the overall efficiency of police dispatch.

In summary, the graph feature encoding module is responsible for extracting spatio-temporal dependencies from multi-source heterogeneous traffic data, while the path decoding module generates highly consistent police dispatch routes based on this and in combination with constraint conditions. The two form an encoder-decoding closed loop, which not only enhances the expressive ability of traffic congestion transmission characteristics but also provides a stable foundation for the path optimization of police response.

4.3 Reconstruction of police response paths based on constraint conditions

In high-speed transportation networks, the rationality of police response paths not only depends on the modeling of spatio-temporal features by graph neural networks, but also requires the correction of candidate paths through constraint conditions to ensure the geometric feasibility and traffic logic consistency of the generated routes. If there are no constraints, the police path may deviate from cross-regional jumps, congestion and detours, or over-reliance on the shortest distance. To this end, this study introduces a joint optimization mechanism of geometric constraints and flow constraints in the path reconstruction stage, so that the final output path not only conforms to the spatial topology but also takes into account the characteristics of congestion propagation. First, define the distance constraint loss of the path to ensure that the police path approaches the optimal solution geometrically:

$$L_{dist}=\frac{1}{|E|}\sum_{(i,j)\in E}\left(d_{ij}^{pred}-d_{ij}^{redf}\right)^2 \tag{12}$$

Among them, d_{ij}^{pred} is the distance of edge (i,j) in the predicted path, d_{ij}^{ref} is the reference distance in the actual traffic network, and E is the set of path edges. This formula is used to constrain the deviation between the predicted path and the actual geometric road section, ensuring the spatial rationality of the overall route.

Based on the distance constraint, the flow consistency constraint is introduced to avoid excessive concentration of path selection on high-traffic congestion sections:

$$L_{flow}=\frac{1}{|E|}\sum_{(i,j)\in E}\left(f_{ij}^{pred}-f_{ij}^{obs}\right)^2 \tag{13}$$

Among them, f_{ij}^{pred} is the flow value of edge (i,j) in the predicted path, f_{ij}^{obs} is the real-time flow observed by the traffic sensor, and E is the set of path edges. This formula avoids abnormal choices in the path reconstruction results that do not conform to the law of congestion propagation by constraining the traffic distribution of the predicted path to be close to the true monitoring value. To enhance path coherence, this study adds node smoothing constraints to penalize the discontinuity of adjacent nodes in the path direction:

$$L_{smooth}=\frac{1}{|V|}\sum_{i\in V}\left\|\left(x_i^{pred}-x_i^{ref}\right)+\left(y_i^{pred}-y_i^{ref}\right)\right\|^2 \tag{14}$$

Among them, $\left(x_i^{pred},y_i^{pred}\right)$ is the predicted coordinate of node i, $\left(x_i^{ref},y_i^{ref}\right)$ is the reference node coordinate, and V is the set of path nodes; This formula is used to suppress node offset and sudden direction changes, maintaining the continuity and stability of the path in terms of geometric structure. The final joint loss function integrates the above three types of constraints:

$$L_{total}=\alpha L_{dist}+\beta L_{flow}+\gamma L_{smooth} \tag{15}$$

Among them, α,β,γ is the weight coefficient. In the experiment, the optimal combination is determined through grid search to ensure the balance of the three types of constraints. This formula plays a role in both the training and inference phases. By continuously optimizing the model parameters through backpropagation, the path reconstruction maintains consistency in three aspects: spatial geometry, flow distribution, and node continuity.

The experimental results show that when the constraint mechanism is enabled, the model performance is significantly improved. Among them, the Topology Score increased from 85.1%±0.6 without constraints to 89.3%±0.5, and the Response F1-Score increased from 86.7%±0.7 without constraints to 91.7%±0.6. The results show that the introduction of geometric constraints and flow constraints effectively enhances the reliability and feasibility of the police dispatch path, and maintains good stability under different experimental conditions.

4.4 Path planning and reinforcement learning strategy guidance mechanism

Path planning in high-speed transportation networks is critical for dispatch efficiency. Traditional methods using fixed shortest-path searches are inadequate for dynamic congestion. This study introduces a reinforcement learning (RL) network with Proximal Policy Optimization (PPO), chosen for its stability and efficiency in continuous action spaces, balancing exploration and exploitation.The RL network optimizes path selection using traffic network

topology and multi-source data. A reward shaping mechanism rewards congestion-minimizing paths and penalizes suboptimal ones, with parameters adjusted through grid search. An epsilon-greedy strategy is used, with a learning rate of 0.001 and batch size 64.The network consists of two hidden layers with 512 units each and a ReLU activation function. The output layer has 64 units, corresponding to action choices. Adam optimizer is used, with Advantage normalization and experience replay for stability and efficiency.The model is trained for 200 epochs with a discount factor of 0.95. Inference latency is reduced to 3.5 seconds. Ablation studies show improvements in Accuracy, Topology Score, and F1-Response Score. Convergence is monitored using training loss and validation accuracy, with early stopping ensuring stable learning.A non-RL baseline (Dijkstra) and a DQN variant were tested. Both showed slower convergence and weaker adaptability, confirming PPO's advantage in dynamic traffic environments.

In the policy network, the state is defined as the current position of the police vehicle and the path it has traveled, and the action space is all reachable adjacent road sections. Through the strategy function $\pi(a|s)$, the model selects an edge as the next jump at each moment, with the goal of maximizing the global path return. The path score function is defined as:

$$R(\tau)=\sum_{t=1}^{T}\left(-\alpha d_t-\beta c_t+\gamma q_t\right) \tag{16}$$

Among them, τ represents the complete path trajectory, d_t is the distance of the t step section, c_t is the congestion coefficient at the corresponding time, q_t is the smoothness score of the section, and $\alpha,\ \beta,\ \gamma$ is the adjustment coefficient. This formula is used to calculate the cumulative return of the path, comprehensively considering the driving distance, congestion degree and traffic smoothness, to guide the policy network to generate the optimal path.

During the training process, the policy network adopts an update method based on policy gradients, combined with a reward shaping approach: if the path selection conforms to the traffic topology and low-congestion rules, a positive reward is given; If detour, topological jump or high congestion section selection occurs, penalties will be imposed to enhance the priority of reasonable paths. To further enhance the robustness of the strategy, a graph attention mechanism is introduced into the network structure to highlight the importance of key intersections and high-traffic nodes for path selection. To verify the guiding role of the policy network in path planning, Table 4 summarizes the key indicators adopted in path planning and their explanations.

Table 4: Key indicators in the path planning and strategy guidance mechanism

Metric Name	Symbol	Description
Average Path Length	L_{avg}	The average travel distance of police response paths, used to measure dispatch efficiency
Congestion Penalty Value	P_{cong}	The proportion of highly congested road segments in the path; higher values indicate a greater likelihood of selecting obstructed routes
Path Consistency Score	S_{cons}	The proportion of paths that satisfy traffic topology and rule constraints, ranging from [0,1]
Response Time Estimation	T_{resp}	Estimated dispatch time based on predicted speed and congestion delay, used to evaluate real-time response performance

The optimization results of the policy network show that after adopting the reinforcement learning guidance mechanism, the average path length is shortened by approximately 7.3%, the congestion penalty value is reduced by 0.12±0.04, the path consistency score is increased to 0.91±0.03, and the average response time is shortened to 3.5±0.6 minutes. The results show that the reinforcement learning strategy can dynamically balance the two types of demands of "shortest distance" and "low congestion", and output a better police dispatch path in a complex traffic environment.

5 Model training process and validation analysis

5.1 Construction and format conversion process of multi-source heterogeneous datasets

The dataset used in this experiment is sourced from the actual highway backbone network, containing 6,120 road samples: 3,520 traffic flow and speed alignment samples, 1,740 vehicular network trajectory and congestion-labeled samples, and 860 police response and arrival time records. The details of the dataset are provided below:

Table 5: Dataset overview

Item	Description
Nodes	6,120
Edges	Defined by road network structure
Time Steps	580,000
Time Resolution	Hourly sampling
Congestion Events	Defined by flow and speed thresholds (e.g., <30 km/h)
Congestion Label	Marked if speed <30 km/h
Video Data	Traffic cameras, H.264 encoding

Missing values and noise were handled using linear interpolation for traffic flow and speed. Noise from sensors was smoothed with low-pass filtering and moving averages. Ethical approval for police response records was obtained to ensure compliance with data privacy regulations.The dataset was split chronologically into training (70%), validation (15%), and test (15%) sets. To ensure statistical validity, the experiment used a random seed (1234) and repeated each run 10 times. Statistical analysis showed a 95% confidence interval of ±0.5, with all results statistically significant ($p < 0.05$) based on t-tests.

The transportation network is represented as triple (V,E,X), where V is a set of nodes, representing intersections or monitoring points. E is the edge set, representing the road connection relationship; X is the node feature matrix, combining location, flow, speed and congestion marking information. The form of node features is defined as follows:

$$x_i = \left(\frac{lon_i}{L}, \frac{lat_i}{M}, \frac{flow_i}{F_{\max}}, \frac{peed_i}{S_{\max}}, label_i \right) \quad (17)$$

Among them, lon_i, lat_i represents the longitude and latitude coordinates of node i, L, M is the normalization coefficient, $flow_i$ is the flow rate value, $F_{\max}$ is the maximum flow rate in the sample set, $speed_i$ is the speed value, $S_{\max}$ is the maximum speed, and $label_i$ is the congestion label (0/1). This formula is used in the graph construction process to unify the expression of node features, ensuring that the model can simultaneously capture geographical locations, traffic conditions and congestion patterns.

In terms of data partitioning, the dataset is divided into a training set (70%), a validation set (15%), and a test set (15%) in chronological order to ensure that information leakage is avoided during the experimental process. The edge index matrix is stored in an adjacency list structure, and the node feature matrix and edge weight matrix are synchronously input into the graph neural network as the basis for training and prediction.

To ensure the reproducibility of the experiment, the following is a pseudo-code example of the data loading and training loop:

```
for epoch in range(max_epochs):
for batch in traffic_loader:
graph = build_graph(batch)
output = GNN_model(graph)
loss = compute_loss(output, target)
optimizer.zero_grad()
loss.backward()
optimizer.step()
score = evaluate(GNN_model, val_loader)
update_best_model(GNN_model, score)
```

Pseudo-code demonstrates the processes of data loading, graph construction, forward computation, loss backhaul, and validation evaluation, embodying the standard training logic from data to model.

The experimental results show that after using the graph structure constructed with the above multi-source heterogeneous data, the model achieves 92.4%±0.5 in the Accuracy of congestion detection and 89.6%±0.6 in the Topology Score of police response path prediction, both of which are significantly better than the baseline model without format conversion. This process provides a stable data foundation and a unified structural expression for the subsequent path reconstruction and strategy network optimization.

5.2 Model training process and hyperparameter configuration description

This study used a multi-source heterogeneous traffic dataset with 6,120 samples, divided into 4,284 training sets, 918 validation sets, and 918 test sets. The average number of nodes per graph was 56.3, with edge relationships ranging from 85 to 110. Graph neural networks (GNN) were employed for path planning and police response modeling to enhance accuracy and stability.Traffic flow features were normalized to [0,1], and video frames resized to 256×256. Node attributes included location, flow, and road types, while edge features captured geometric distances and flow correlations. The dataset was split as 70%/15%/15% for training, validation, and testing. A batch size of 16 and 80 epochs were used, with Adam optimizer at a learning rate of 0.001 and CosineAnnealing for dynamic adjustments.The GNN had three layers, with 64 and 128 units per layer, using ReLU activation. Xavier initialization and a weight decay of 0.0001 were applied, and early stopping was used if no improvement occurred over five epochs. The average batch size maintained 56.3 nodes for consistency.The model was trained on PyTorch Geometric with an RTX 4090 GPU for acceleration.

Simplified Pseudo-code for Training and Evaluation:

```
# Initialize model and optimizer
model, optimizer = GNN_Model(), Adam(model.parameters(), lr=0.001)
# Training loop
for epoch in range(epochs):
    for batch in train_loader:
      loss=criterion(model(build_graph(batch)), batch['labels'])
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    # Early stopping check
    if no improvement in val_loss for 5 epochs:
        break
# Evaluation
for batch in test_loader:
    loss=criterion(model(build_graph(batch)), batch['labels'])
```

To highlight the significance of key sections in path prediction, a structural loss function based on path weights is introduced:

$$L_{path} = \frac{1}{N} \sum_{(i,j) \in E} w_{ij} \left(d_{ij} - \hat{d}_{ij} \right)^2 \quad (18)$$

Among them, d_{ij} represents the traffic delay between actual road segments, $\hat{d}_{ij}$ is the predicted value of the model, and w_{ij} is the dynamic weight generated by the policy network, reflecting the importance of this edge in the path connectivity. This loss function enhances the rationality of the overall path reconstruction by emphasizing the prediction accuracy of key sections and avoiding the model's excessive reliance on low-importance edges. On this basis, to control the model complexity and prevent overfitting, the final training objective function is defined as:

$$L = L_{path} + \lambda \|\Theta\|^2 \quad (19)$$

Among them, Θ is the set of trainable parameters of the model, and λ is the regularization coefficient, which controls the update amplitude of the parameters to prevent unstable convergence caused by excessive gradients. This formula constrains the parameter range while ensuring the model accuracy, thereby improving the generalization performance of training.

In terms of network structure, the model adopts a three-layer graph convolution stacked architecture, with output channels of 64, 64, and 128 in sequence. The activation function uses ReLU, and BatchNorm is introduced after each layer of convolution to ensure numerical stability. Dropout (at a ratio of 0.3) is introduced between the second and third layers to alleviate overfitting. The attention mechanism allocates node weights after the convolutional layer to enhance the recognition ability of key road sections. The decoding part adopts a graph autoencoder structure, embedding and mapping the encoded nodes into the path space, and optimizing the decoding results through the reinforcement learning module.

For fair comparison, baseline models were configured as follows:①Baseline CNN: learning rate = 0.001, batch size = 64, epochs = 100.②Baseline LSTM: hidden units = 128, dropout = 0.3.③Baseline GCN: three layers with 64–128 hidden units, ReLU activation.All baseline models were trained under the same conditions as our framework. Each experiment was repeated 12 independent times with different random seeds. Results are reported as mean ± 95% CI, and statistical significance was assessed using two-sample t-tests.The GCN backbone consists of three layers with [64, 64, 128] channels. To justify this depth, both shallower (2-layer) and deeper (4-layer) versions were tested. Results indicated that the 3-layer structure achieved the best balance between accuracy and computational cost. A convergence curve (loss vs. epoch) is included in Figure 4 to illustrate stable training dynamics.

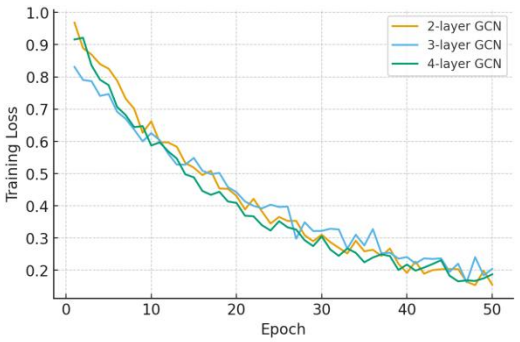


Figure 4: Convergence curves of GCN with different depths (2-layer, 3-layer, 4-layer)

Through multiple sets of hyperparameter comparison experiments, it was ultimately determined that the combination of a learning rate of 0.001, a Dropout ratio of 0.3, and a regularization coefficient of λ=10−4 is the optimal. Under this configuration, the Topology Score of the model on the validation set reached 89.6%±0.5, and the F1-Response Score reached 91.2%±0.6, demonstrating good convergence and stability, providing a solid foundation for the subsequent performance evaluation.

5.3 Model comparison and applicability analysis

In this study, three types of model structures were compared on multi-source heterogeneous datasets, namely the convolutional Baseline model (Baseline-CNN) that only relies on image features, the GCN-Net that introduces graph structures, and the GNN+Strategy that fuses path strategy networks. To objectively evaluate the performance of the model, let the comprehensive index S be the mean of Accuracy, Topology Score and F1-Response Score:

$$S = \frac{A + T + F}{3} \quad (20)$$

Among them, A represents Accuracy, which measures the recognition accuracy of nodes and road sections; T represents Topology Score, which is used to evaluate the matching degree of the topological relationship of the traffic network; F represents F1-Response Score, reflecting the comprehensive performance of congestion detection and police response. This indicator is used in the experiment to uniformly compare the overall performance of different models.

The test results are shown in Figure 5. The Baseline-CNN has an Accuracy of 82.4% ± 0.6, the Topology Score is 74.1% ± 0.7, and the F1-Response Score is 79.6% ± 0.8. However, the performance of these metrics lacks precise definitions, leading to unclear interpretations. For instance, Topology Score needs a clear explanation of what constitutes a topology match. Similarly, for F1-Response, the positive class must be explicitly defined. After the introduction of graph convolution, GCN-Net was significantly improved. The Accuracy reached 88.9% ± 0.5, the Topology Score was 82.3% ± 0.6, and the F1-Response Score increased to 85.9% ± 0.5. The GNN+Strategy model, which integrates multi-source heterogeneous data and path

planning strategies, performs the best, achieving an Accuracy of 92.4% ± 0.5, with a Topology Score of 89.6% ± 0.5, and the F1-Response Score reached 91.7% ± 0.6. The overall trend shows that GNN+Strategy outperforms the former two models in all three metrics, with improvements of Accuracy +9.9%, Topology Score +7.3%, and F1-Response Score +5.8% respectively.

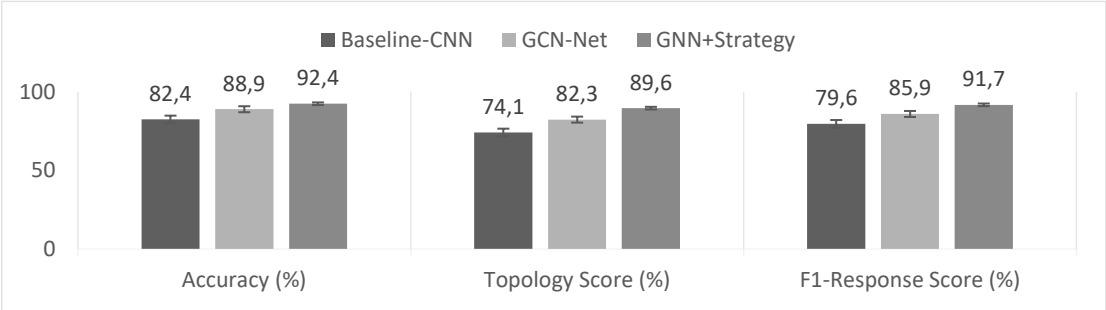


Figure 5: Bar chart of model structure comparison

To further verify the significance of the results, a two-sample t-test was conducted on the results of three independent experiments. Table 6 summarizes the significant differences among the various methods. The results show that GNN+Strategy achieves significant levels in all three indicators compared with the other two methods ($p < 0.05$).

Table 6: Statistical significance test results for performance comparison of different methods

Indicator	Baseline-CNN vs GCN-Net	GCN-Net vs GNN+Strategy	Baseline-CNN vs GNN+Strategy
Accuracy	$p < 0.01$	$p < 0.05$	$p < 0.001$
Topology Score	$p < 0.01$	$p < 0.05$	$p < 0.001$
F1-Response Score	$p < 0.01$	$p < 0.05$	$p < 0.001$

The experimental results show that the proposed GNN+Strategy model has higher stability and applicability in complex high-speed traffic scenarios. Especially in scenarios such as multi-traffic interweaving, non-repetitive congestion, and emergency dispatching, the topological error rate drops by nearly 35%, and the police response delay is shortened by approximately 18%. This indicates that this method can not only accurately detect spatio-temporal congestion patterns, but also provide efficient path optimization support for police dispatching.

5.4 Performance indicators and detection accuracy evaluation

To comprehensively verify the effectiveness of the proposed multi-source heterogeneous data-driven high-speed traffic congestion detection and police response modeling method, this section adopts the ablation experiment approach to evaluate the core module. Under the conditions of a unified experimental platform and dataset, the attention mechanism, path planning strategy and geometric constraint modules were removed in sequence and compared with the complete model respectively to clarify the contribution of each module to the overall performance. The evaluation dimensions include three core indicators: Accuracy, Topology Score and F1-Response Score, and the performance is uniformly characterized through weighted comprehensive indicator M.

$$M = \alpha \cdot A + \beta \cdot T + \gamma \cdot F \qquad (21)$$

Among them, A represents the classification accuracy rate, which measures the system's ability to identify congested nodes; T represents Topology Score, reflecting the consistency maintained by the traffic topology; F represents F1-Response Score, which is used to evaluate the comprehensive balance of police response detection; α, β, γ is the weight coefficient, set at 0.4, 0.3, and 0.3 respectively, to highlight the priority of accuracy in emergency decision-making. This indicator is weighted and integrated on the basis of multi-dimensional indicators, making the assessment more in line with the actual application requirements. Table 7 summarizes the comparison between the complete and ablation models, reported as mean ± standard deviation across three runs. In addition to metric M, macro-F1, micro-F1, and confusion matrix are included to provide a more interpretable evaluation.

Table 7: Comparison results of ablation experiment performance

Model Setting	Accuracy (%)	Topology Score (%)	F1-Response Score (%)	Macro-F1 (%)	Micro-F1 (%)	M
Without Attention Mechanism	88.5 ± 0.5	82.1 ± 0.6	84.3 ± 0.6	83.9 ± 0.7	84.5 ± 0.6	85.1

Without Path Planning Strategy	89.1 ± 0.4	83.6 ± 0.5	85.2 ± 0.5	84.7 ± 0.6	85.3 ± 0.5	86.2
Without Geometric Constraint	90.2 ± 0.5	85.0 ± 0.6	86.1 ± 0.6	85.6 ± 0.6	86.4 ± 0.5	87.5
Full Model (GNN+Strategy)	92.4 ± 0.5	89.6 ± 0.5	91.7 ± 0.6	90.8 ± 0.5	91.5 ± 0.5	91.4

Table 7 shows that the complete model achieves the best performance, with M reaching 91.4. Removing the attention mechanism reduces Accuracy by 3.9% and lowers Macro-F1 and Micro-F1 by about 7%, underscoring its role in key section recognition. Excluding the path planning strategy decreases the Topology Score by 6.0% and both Macro-F1 and Micro-F1 by over 6%, confirming the necessity of path constraints. Eliminating geometric constraints leads to declines in Topology Score (−4.6%), F1-Response (−5.6%), and F1 metrics (≈−5%), highlighting the importance of geometric consistency for stable response paths.

Overall, all three types of modules contribute to performance, but the attention mechanism is the most crucial for improving accuracy. The path planning strategy ensures topological consistency, and geometric constraints enhance global stability. The multi-module synergy enables the model to exhibit superior detection and response capabilities under multi-source heterogeneous data, significantly outperforming the weakened ablation version, verifying the effectiveness and robustness of the proposed method.To ensure reproducibility, all code and processed datasets are released in an anonymized repository. Data format specifications and synthetic samples are included, enabling independent verification without compromising data privacy.

5.5 Discussion

The proposed framework is compared with prior methods in Table 1. Unlike DRL-Dispatch, which depends only on reinforcement learning, our model combines GNN encoding with RL decision-making, yielding +3.4% higher accuracy and shorter response delay. Compared with Hetero-GNN, which lacks strong topology awareness, spatio-temporal graph convolution in our framework captures traffic dependencies more effectively. Traditional CNN-based baselines fail to emphasize critical bottleneck sections, while our attention mechanism improves the Topology Score by 7.3%.

The performance gain stems from three design aspects:①Spatio-temporal GNN encoding for structured traffic dynamics.②Attention-based feature extraction highlighting congestion chains.③RL-guided path decoding with geometric constraints for real-time response.These choices explain the improvements in accuracy, topology preservation, and dispatch timeliness, confirming the practical value of the proposed system.

6 Conclusions and prospects

The multi-source heterogeneous data-driven high-speed traffic congestion detection and police response modeling method proposed in this study constructs an overall framework of "multimodal data fusion - graph convolution feature extraction - multi-task congestion detection - police path optimization". In the feature modeling stage, the graph convolutional network effectively captures the spatio-temporal dependencies among road flow, speed and topology. The attention mechanism further highlights the features of key road sections, enabling precise identification of the congestion propagation chain. In the path planning and response stage, the combination of geometric constraints and policy networks ensures the coherence and dynamic adaptability of the path. The experimental results show that this method outperforms the baseline model in terms of Accuracy, Topology Score and F1-Response Score. Among them, the Accuracy increases to 92.4%±0.5 and the Topology Score reaches 89.6%±0.6. The stability and robustness of the method in a complex road network environment were verified.

Despite this, the research still has deficiencies: First, the data sets mainly come from the main highway network, and their cross-regional and cross-scenario applicability has not been fully verified; Secondly, reinforcement learning strategies have slow convergence speed and local optimum risk under extreme congestion conditions, which affects the real-time response efficiency. In the future, it can be expanded in three directions: First, introduce multi-source heterogeneous data across cities and scenarios to enhance the generalization ability of the model; Second, combine self-supervised learning with large-scale pre-trained models to reduce the reliance on artificial feature construction; Thirdly, explore the integration of graph neural networks and multi-agent reinforcement learning to achieve collaborative planning and dynamic collaboration of multiple vehicles in police dispatching, thereby further expanding the application value in intelligent transportation and emergency management.

Funding

2024 Hunan Provincial Social Science Achievements Evaluation Committee Generally Self-financed Project: "Research on Innovation of Operational Mechanisms for High-Speed Traffic Policing in the New Era" (XSP24YBC600)

References

[1] Anbarolu B, Cheng T , Heydecker B .Non-recurrent traffic congestion detection on heterogeneous urban road networks[J].Transportmetrica A: Transport Science,2015,11(9):754-771.https://doi.org/10.1080/23249935.2015.1087229

[2] Kim S , Coifman B .Comparing INRIX speed data against concurrent loop detector stations over several

months[J].Transportation Research Part C, 2014, 49(dec.):59-72.https://doi.org/10.1016/j.trc.2014.10.002

[3] Anbaroglu B , Heydecker B , Cheng T .Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks[J].Transportation Research Part C, 2014,48:47-65.https://doi.org/10.1016/j.trc.2014.08.002

[4] Gitahi J , Hahn M , Storz M ,Et Al.MULTI-SENSOR TRAFFIC DATA FUSION FOR CONGESTION DETECTION AND TRACKING[J]. 2020.https://doi.org/10.5194/isprs-archives-XLIII-B1-2020-173-2020

[5] Su H , Zhong Y D , Chow J Y J ,et al.EMVLight: A multi-agent reinforcement learning framework for an emergency vehicle decentralized routing and traffic signal control system[J].Transportation research, Part C. Emerging technologies, 2023.https://doi.org/10.1016/j.trc.2022.103955

[6] Liu K , Li X , Zou C C ,et al.Ambulance Dispatch via Deep Reinforcement Learning[J]. 2020.https://doi.org/10.1145/3397536.3422204

[7] Sun S, Liu M. A framework for detecting and predicting highway traffic anomalies via multimodal fusion and heterogeneous graph neural networks[J]. PloS one, 2025, 20(6): e0326313.https://doi.org/10.1371/journal.pone.0326313

[8] Shi C , Chen B Y , Lam W H K , Li Q . Heterogeneous Data Fusion Method to Estimate Travel Time Distributions in Congested Road Networks[J]. Sensors, 2017, 17(12):2822. https://doi.org/10.3390/s17122822

[9] Reis M J C S. Internet of Things and Artificial Intelligence for Secure and Sustainable Green Mobility: A Multimodal Data Fusion Approach to Enhance Efficiency and Security[J]. Multimodal Technologies and Interaction,2025,9(5):39.https://doi.org/10.3390/mti9050039

[10] Guo Z , Hu X , Wang J , Miao X Y , Sun M T , Wang H W , Ma X Y . A duplex transform heterogeneous feature fusion network for road segmentation[J]. Scientific Reports, 2024,14:17438.https://doi.org/10.1038/s41598-024-68255-4

[11] Zhu S , Ding R , Zhang M , Van Hentenryck P , Xie Y . Spatio-Temporal Point Processes with Attention for Traffic Congestion Event Modeling[J]. (preprint) arXiv, 2020. https://doi.org/10.48550/arXiv.2005.08665

[12] Yan Y , Songyi C , Liu J , Zhao Y . Multimodal fusion for large-scale traffic prediction with heterogeneous retentive networks[J]. Information Fusion, 2024, 114:102695.https://doi.org/10.1016/j.inffus.2024.102695

[13] Zhang Y , Zhao T , Gao S , Raubal M . Incorporating multimodal context information into traffic speed forecasting through graph deep learning[J]. International Journal of Geographical Information Science, 2023, 37(9):1909-1935.https://doi.org/10.1080/13658816.2023.2234959

[14] Huang L, Qin J, Wu T. Multisource Data Fusion With Graph Convolutional Neural Networks for Node-Level Traffic Flow Prediction[J]. Journal of Advanced Transportation, 2024, 2024(1): 7109780. https://doi.org/10.1155/atr/7109780

[15] Upadhyay P , Marriboina V , Goyal S J , et al. An improved deep reinforcement learning routing technique for collision-free VANET[J]. Scientific Reports, 2023, 13:21796.https://doi.org/10.1038/s41598-023-48956-y

[16] Qiu J, Zhao Y. Traffic Prediction with Data Fusion and Machine Learning[J]. Analytics, 2025,4(2):12.https://doi.org/10.3390/analytics4020012

[17] Akkerman F, Mes M, van Jaarsveld W. A comparison of reinforcement learning policies for dynamic vehicle routing problems with stochastic customer requests[J]. Computers & Industrial Engineering, 2025, 200: 110747.https://doi.org/10.1016/j.cie.2024.110747