A Critical Analysis of Brain Tumor MRI Segmentation and Classification Utilizing Machine Learning and Deep Learning Methods

Aruna V S, J. Vijayashree*
School of Computer Science and Engineering, Vellore Institute of Technology, Vellore
E-mail: aruna.vs2023@vitstudent.ac.in, vijayashree.j@vit.ac.in
*Corresponding author

Keywords: brain tumor, ischemic stroke, computer aided diagnostic, MRI, UNet, discrete wavelet transform (DWT) and residual network

Received: February 4, 2025

Brain Tumor (BT) result from uncontrolled cell growth and can be fatal if not treated. Classification and segmentation of data remain difficult despite many large-scale initiatives and encouraging results. Variations in the location, shape, and size of tumors make diagnosis difficult for doctors. This report provides a comprehensive literature analysis on magnetic resonance imaging (MRI) to aid researchers in detecting brain tumors (BT). The subject matter includes topics such as the anatomy of the Brain Tumor (BT), publicly available datasets, methods to improve the quality of images, dividing the tumor into distinct parts, extracting important characteristics, categorizing the tumor, using advanced Machine Learning (ML) techniques like Deep Learning (DL), transferring knowledge from one task to another, and employing fuzzy sets for analysis. The review provides an extensive overview of ML and DL methods for BT classification. With its capability to analyse vast amounts of data, DL has shown outstanding performance across various fields, particularly in biomedicine. This assessment offers comprehensive information about both 2-dimensional (2D) and 3-dimensional (3D) datasets and the methodologies used. The use and testing of MRI scans have been utilized to identify BT, resulting in positive outcomes. The goal of this study is to conduct a thorough and critical review of existing research on BT detection and classification using MRI.

Povzetek: Članek nudi poglobljen pregled naprednih metod strojnega in globokega učenja za MRI-segmentacijo in klasifikacijo možganskih tumorjev, ter izpostavi izzive, podatkovne nabore in smernice prihodnjega razvoja.

1 Introduction

As the brain and spinal cord facilitate dispersion, the Central Nervous System (CNS) conveys information about the body's senses and motor functions. The main brain components include the brain stem, cerebrum, and cerebellum. The typical male brain weighs 1.2-1.4 K and is 1260 cm3 and the female brain is 1130 cm3. Motor control, judgment, and problem-solving are all helped by the frontal lobe. The parietal lobe controls posture. The occipital governs vision, whereas the temporal handles memory and hearing. Neurons form a greyish cerebral cortex at the cerebrum's outermost section. The brain is bigger than the cerebellum. In neural-system animals, motor control organizes voluntary movements. Due to their changing size, stroke area and lesioning cannot detect tiny lesion patches. Human cerebellums are more developed than others. The cerebellum has three lobes: anterior, posterior, and flocculonodular. Each lobe is joined to the one behind it by the circular vermis. The greyish cortex and white matter (WM) core are thinner in the cerebellum than in the cerebrum. The anterior and posterior lobes collaborate to coordinate complex motor activities. The flocculonodular lobe keeps bodily processes in balance. The balancing act, breathing, and eye movement are regulated by peripheral and central nervous system bundles. From the brainstem to the spinal cord, thalamic nerves travel. All across the body, they were scattered. The medulla, pons, and midbrain comprise the brain stem. The midbrain facilitates ocular, muscular, auditory, and visual movements. While the pons regulates respiration, intra-brain communication, and sensations, the medulla oblongata regulates blood flow, swallowing, and sneezing [1]–[3].

The revolution in computer vision and ML has enabled the development of novel ideas and methods. Its applications in education, healthcare, and self-driving cars are just a few of the domains in which it has shown exceptional success [4]. Anomaly detection is one area where scientists have recently been focusing their attention on the biomedical applications of ML and Artificial Intelligence (AI) [5]. Automated, semi-automatic, or hybrid models that can accurately and quickly identify and segment tumors have been developed by researchers using DL [6] as a subfield of ML [7]. Since BT may vary greatly in size, form, location, and

appearance, radiologists still face challenges, even though early identification improves long-term survival and offers a more favourable prognosis. Doctors, patients, and researchers have already benefited from extensive work in this area. Although several Computer-Aided Diagnostic (CAD) systems have been developed to automatically detect and classify brain anomalies [8] and [9], there are still significant gaps in the field. Numerous evaluations have been published in this area, but none of them have addressed the shortcomings of the previous research or offered any noteworthy insights for the way forward. While deep models have the issue of gradient vanishing, hybrid models are not compatible with one another. The uniformity of data pre-processing is also deficient. To create a link across various methods, algorithms, and domains, DL optimization algorithms are needed. According to [7-9], the largest disadvantage of DL is its enormous annotated data need. The only way to find a solution to these issues is to conduct a comprehensive analysis of the methods already used and documented in the literature.

This study included current research articles that investigated the use of DL in identifying and categorizing BT from 2019 to 2024. The most recent cutting-edge studies on BT detection and categorization are what we want to review. The primary motivation for this study is our interest in working on a particular DL model known as multi-task learning. We also want to learn more about the weaknesses and vulnerabilities in current DL models so that we may better understand how to implement our suggested DL strategy.

The paper's structure is as follows: Section 1 serves as an introductory section of the study, while the second part 2 offers an in-depth elucidation of BT after the introduction. Section 3 of the report encompasses the literature review undertaken between 2019 and 2024. Topics covered include BT classifying and segmenting using various ML and DL methods, datasets, and performance indicators. Section 4 provides a comprehensive analysis of this work, including its drawbacks. Section 5 presents the study's conclusion, while Part 6 provides the sources used in this work.

Brain tumor

BT are categorized as either aggressive or slow-growing. A malignant tumor, characterized by its aggressive nature, can metastasize and spread from its initial place to many organs and tissues. In contrast, a benign tumor, which grows at a slower pace, does not invade or infiltrate the surrounding tissues [10]. As far as the World Health Organization (WHO) is concerned, BT falls within categories I–IV [11], [12]. Grades III and IV malignancies are characterized by their high aggressiveness and poor prognosis, while malignancies of grades I and II are thought to advance slower. The details about the grading of BT are outlined below [13], [14].

Grade I: Cancers are characterized by their sluggish growth and low tendency to spread. These may be effectively removed by surgical procedures and have been associated with enhanced duration of survival. Grade 1 pilocytic astrocytoma is one sort of this tumor.

Grade II: Cancers have a sluggish growth rate, but possess the capability to metastasize to other tissues and progress into tumors of a higher grade. Despite undergoing surgical intervention, several types of malignancies have the potential to reoccur. Oligodendroglioma is a kind of this type of tumor.

Grade III: Tumors can invade neighbouring tissues and have a faster growth rate compared to grade II tumors. For certain types of malignancies, surgical intervention alone is insufficient; postoperative chemotherapy or radiation treatment is offered. One particular instance of this kind of tumor is an anaplastic astrocytoma.

Grade IV: Tumors are characterized by their high aggressiveness and ability to spread rapidly. They could use blood vessels to expedite their development.

Ischemic stroke: An ischemic stroke occurs when the brain's blood flow is interrupted. An ischemic stroke occurs when the blood supply to a portion of the brain is cut off, depriving the tissue of enough oxygen and hastening the process of death. Context Globally, ischemic stroke is a leading cause of mortality and disability. Lesions related to strokes are classified as acute (0–24 hours), sub-acute (1–2 weeks), and chronic (since more than 2 weeks ago).

A BT results from the uncontrolled development of both living and dead brain cells. Disease size and severity affect BT survival. Primary and secondary BT differ in genesis. Secondary tumors travel to the brain from the outside, whereas primary brain malignancies start within [15]. The soft tissue composition of the three components and tumor exhibit varying contrasts under various physical conditions, rendering them crucial for the identification of BT in MRI imaging. T1, T2, T1-CE, and Fluid Attenuation Inversion Recovery (FLAIR) have frequently used MRI images [16], [17] as shown in Figure 1.

T1-weighted images: In T1-weighted imaging, cerebrospinal fluid (CSF) appears as a dark color, whereas MRI reveals adipose tissue.

T2-weighted images: Prolonged lengths of Repetition Time (TR) and Time to Echo (TE) result in increased brightness of CSF. These images depict the distribution of adipose tissue and hydration levels throughout the body.

T1-CE images: While the TE and TR are identical to those of T1 images, these images are taken after injecting a non-toxic gadolinium contrast agent. The agent can light imaging locations.

FLAIR images: By using extended TE and TR values, FLAIR images are acquired. By doing this action, any irregularities stay visible while the normal CSF appears black. This sequence is prone to sickness, which facilitates the differentiation of CSF and abnormalities [5], [18].

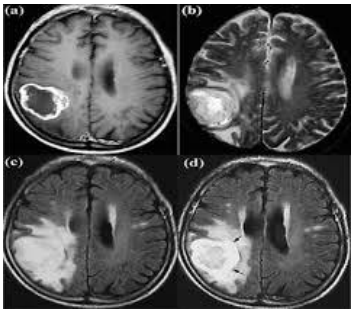


Figure 1: The imaging modalities used in this case are a. T1 Weighted, b. T2 Weighted, c. FLAIR, and d. FLAIR with contrast enhancement (https://bit.ly/33ab5og)

Brain imaging modalities

Medical imaging remains the most efficient approach for discovering, evaluating, and diagnosing gliomas and other medical conditions. Computerized Tomography (CT), positron emission tomography, MRI, functional magnetic resonance imaging, and Diffusion Tensor Imaging (DTI) are the imaging modalities most often used in clinical settings. While each of these methods has its own set of benefits, they may all be combined to create a more comprehensive view. Mathematical models may be used to predict the progression of BT over time by using many time points of various modalities. Growth prediction has special importance as it may provide valuable insights into the tumor's physiology, aid in measuring the tumor's aggressiveness, enhance treatment planning, and perhaps even forecast the patients' survival in the context of precision medicine [19].

The evaluation of medical data was performed by using ML and DL techniques, applying the following methods:

MRI: To acquire electromagnetic signals, magnetic resonance is used. These signals are produced by human organs, which helps us reconstruct further details on the anatomy of human organs. High-resolution MRIs include more structural features that are necessary for identifying lesions and diagnosing diseases.

CT-Scan: This technique uses digital geometry to create 3-D images from 2-D X-ray images.

Mammogram: Mammograms are used for early identification of anomalies in the body and the successful screening of breast cancer. Masses and calcifications are thought to be the most typical abnormalities that lead to breast cancer.

Electrocardiogram (ECG): This device measures electrical heart activity and helps identify cardiac issues in people [20].

Out of all these technologies, MRI is the most superior technology for researching the brain due to its ability to accurately identify tissues with a high level of spatial resolution. The same tissue area may be represented by many images using varying contrast visualizations. These several imaging types may provide very helpful information for the same area, assisting researchers in accurately studying brain disease. It is effective in tumor identification, as it is a non-invasive technique without producing harmful radiation [21].

2 Literature review

The increasing complexity of Cyber-Physical Reviewing various works in the area of DL and ML applications for medical imaging is the main objective of this research. Classification, detection, and segmentation are critical tasks in medical image processing. To train and complete specific DL tasks in medical applications, deep neural networks need large amounts of labelled data. Nonetheless, the medical field is lacking of annotated data points. Transfer Learning (TL) is one tactic to address this problem. Fine-tuning a pre-trained network and fixed feature extractors are two popular and widely utilized TL approaches. The classification method divides images into two or more classes using DL models. In the process of detecting anomalies in medical imaging, DL algorithms identify organs. The models attempt to separate medical image regions of interest for processing in the segmentation challenge [22]. DL can also extract complicated features that humans cannot see, allowing quantified image interpretation [23].

Fuzzy C means feature extraction methods, neural networks, and mathematical morphology were introduced for better performance [24], [25]. Also, an evolutionary algorithm is important in medical image analysis. Genetic algorithms are employed to identify glioma grades noninvasively. It is used to select and optimize features [26], suggested by utilizing MRI [27] and spontaneous categorization to improve accuracy. We find the best features in the dataset using Swarm Intelligent Redundancy Relevance (SIRR) [28]. In the paper [29], the enormous data collection is decreased in dimension using the evolutionary correlated gravitational search method, an excellent optimization strategy.

The research articles that discuss the segmentation and classification of BT MRI images using ML and DL approaches are thoroughly reviewed in this part and span the years 2018 through 2024. This section's format is as follows: We conduct a thorough assessment of the majority of articles that deal with the ML and DL techniques used to separate BT from MRI images, along with the datasets and performance metrics used in each. Most papers use ML and DL techniques to classify MRI images as BT. The classification is shown in Figure 2 and the table along with the datasets and performance metrics used in the examined research papers.

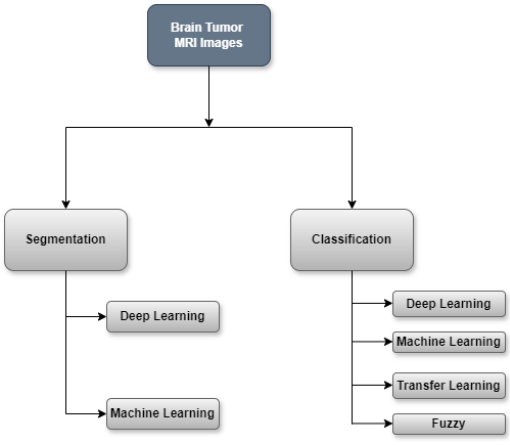


Figure 2: Different methods used in BT classification

2.1 Segmentation of brain tumor using different methods

Automated BT segmentation is essential for computer-aided glioma diagnosis. Automating these procedures could potentially enhance their effectiveness and efficiency. This study presents a hybrid method for segmenting BT. To segment MRI in the BRATS 2013 dataset, the hybrid model makes use of Deep Convolutional Neural Networks (DCNNs) and multimodal MRI data. The hybrid model, which included two-path Convolutional Neural Networks (CNNs) and three-path CNNs, surpassed cutting-edge methods in every significant performance criterion. Using a patch model, the suggested network segments the brain anatomy by classifying the core pixels of individual patches. They slice the 3D MRI into 2D spatial layers. It uses local and global data to predict output classes, considering label correlations between pixels. The training technique for the model is two-phase to ensure precise label distribution and manage inconsistent data [30]. BT exhibits significant variation in terms of their size, shape, and overall appearance, posing challenges in their identification. This study classifies input slices as either healthy or sick (tumor) using a DL network. In this work, a high-pass filter image fused with the input slices highlights the inhomogeneity field impact of the magnetic resonance slices. The fused slices also go through the median filtering procedure. It is easy to enhance the output slices by simply smoothing and emphasizing the borders of the input slices. Subsequently, the 4-connected seed-generating algorithm categorizes pixels with comparable intensity from the input slices. Then, refine the segmented slices into the Stacked Sparse Auto Encoder (SSAE) model using two layers of a recommended stacked sparse autoencoder. The author selected the model's hyperparameters after extensive testing. In the first layer, they use 200 hidden units, and in the second, 400. Evaluated the prediction of images with and without tumors on the softmax layer [31].

Due to the high labour and time requirements of manual segmentation, automatic segmentation is the superior method. When it comes to medical picture segmentation, the UNet is a popular and versatile architecture. For 2D and 3D medical picture segmentation, this paper suggests and applies 2DGA-UNet and 3DGA-UNet, respectively. The initial version uses TL and UNet architecture to improve the 2DGA-UNet framework's performance. Visual Geometry Group 16 (VGG-16), a collection of basic CNN, encodes the 2DGA-UNet network. In 3DGA-UNet, up-sampling operators replace pooling operators, adding layers to a large contractual network. Consequently, these layers can learn from a limited number of images and surpass current best practices with output resolution. These models evaluate their performance on five benchmark datasets as mentioned in Table 1. The findings demonstrate excellent performance across 14 medical image segmentation assessment factors. From the comparison of performance metrics given in Table 3, the GA-UNet performs better than conventional approaches [32]. This paper introduces a novel cloud-based 3D U-Net method for BT segmentation, which makes use of the BRATS dataset. In this research, we look into how the 3-D U-Net network deviates when max pooling layers and convolutional layer initial sequences are combined. Cloud computing has several benefits. This network's worldwide accessibility reduces computational expenses. It effectively trained the system using the many hyperparameters of the Adam optimization solver. We provide the first cloud-based method with an average dice score of 95% that achieves maximal accuracy. Then the dice score is calculated using the Sorensen similarity coefficient [33]. In another article, the author proposes a fully automated approach utilizing the BraTS2020 dataset and 2D U-net design for tumor localization separation from normal tissue. Each distinct MRI sequence serves as training data to retrieve (Region of Interest) ROIs, or tumor regions in brain MRIs. To save computational costs, the input image graphs are reduced in size to a single 128 by 128 images, and then normalized. We evaluate the model using all MRI sequences to determine which yields the highest performance [34].

An attention module named Attention Gate was developed and its efficacy for BT segmentation was the intended focus of this study. The Attention Gate Residual U-Net (AGResU-Net) architecture incorporates attention gates and residual modules into an original, simplistic U-Net design. It augments the skip connection with several attention gate units to focus on relevant feature data and ignore irrelevant or noisy feature replies. Not only does AGResU-Net collect large amounts of semantic data to improve feature learning, but it also manages data on small-scale BT. We do a comprehensive evaluation of attention gate units using three reliable MRI BT benchmarks. Evidence from experiments shows that AGResU-Net and AGU-Net perform better than their respective baselines, ResU-Net and U-Net [35]. A DWT and a Daubechies wavelet kernel are used throughout the fusion process to provide a more informative tumor region than a single, independent MRI signal. After fusion is complete, noise is removed using a Partial Differential Diffusion Filter (PDDF). A suggested CNN model is fed to the segmented tumor region using a global thresholding strategy to discriminate between tumor and non-tumor regions. To evaluate the suggested technique, five datasets that are publicly accessible are used. The results show that fused images perform better than isolated sequences on benchmark datasets [16]. Early diagnosis of BT may lead to patient survival. This study proposes to segment lesion symptoms properly using the Grab Cut technique and refine the TL model VGG-19 by extracting features that are sequentially combined with manually generated (shape and texture) data. Classifiers get a fused vector as a result of entropy's optimization of these characteristics for quick and precise classification. They evaluated the suggested method using three years' worth of multimodal BT segmentation BRATS challenge datasets from the top Medical Image Computing and Computer-Assisted Intervention (MICCAI) competitions. They test and train the suggested approach on benchmark datasets like BRATS 2015–17. The highest possible results on the BRATS 2015 exam are DSC = 0.9636 and accuracy =

0.9878. In 2016, we achieved a DSC of 0.9959 and an accuracy of 0.9963, while in 2017, we achieved a DSC of 0.9967 and an accuracy of 0.9980 [11].

Tumor segmentation and grading are commonly used for MRI and are crucial for diagnosing and formulating a treatment strategy. They developed a DL approach using CNNs built with U-net and TL, a fully connected classifier for grading, and a pre-trained VGG-16 convolution basis to meet the clinical need for tumor segmentation. We train and assess the segmentation and grading models using T1-precontrast, T1-postcontrast, and FLAIR MRI data from 110 patients diagnosed with Lower-Grade Glioma (LGG). The segmentation model has an average DSC of 0.84 and an accuracy of 0.92 in identifying tumors [36]. We talk about a way to use CNN to automatically tell the difference between BT in 3D BT Segmentation. Furthermore, we used whole-brain 3D imaging to compare predicted labelling in three dimensions with ground truth. The method used sagittal, coronal, and axial images to accurately establish the position and dimensions of the tumor, including its height, width, and depth. Tumor prediction outcomes via semantic segmentation with a DL network are promising. 91.718 was the average prediction ratio [37].

Table 1: Summary of methods and datasets used for brain lesion segmentation and classification

Ref.	Method	Dataset
[30]	-CNN with two paths -CNN with three paths - Hybrid CNN	BRATS 2013 Dataset (3D images)
[31]	- Lesion Enhancement Using High Pass Filter - Lesion Segmentation Using Seed Growing Method -Autoencoders (AE) -Sparse Autoencoder (SAE) - Stack Sparse Autoencoder	A few examples of BRATS datasets: 2012 (challenge and synthetic), 2013 (and 2013 Leaderboard), 2014, and 2015 datasets
[32]	- Two variants of GA-UNet: 2DGA-UNet and 3DGA-UNet - TL with VGG16 - encoder in 2DGA-UNet - Up sampling operators replacing pooling operators in 3DGA-UNet	- Brain lesion segmentation (MICCAI 2008 Multiple Sclerosis Challenge) - BRATS 2018 and BRATS 2019 - Lung segmentation - Segmentation of the liver (3D-IRCADb database)
[33]	Segmentation method: 3D U-Net	BraTS 2020 3D
[34]	U-Net	BraTS 2020
[35]	AGResU-Net model	BRATS 2018, 2019, and BRATS 2017
[16]	CNN, PDDF, Daubechies wavelet kernel, and DWT	BRATS Leader Board, BRATS 2012, 2013, 2015, and 2018
[11]	Grab cut technique, VGG 19, and manually created feature extraction using HOG (Histogram of Oriented Gradients) and LBP (Local Binary Pattern)	BRATS 2015, 2016, and 2017
[36]	A pre-trained VGG16, fully connected classifier for segmenting and U-Net	The Cancer Imaging Archive (TCIA)
[37]	Semantic segmentation - CNN	BRATS dataset 3D
[38]	3D fully CNN	BRATS 19 and 2018 datasets
[39]	3D U-Net	BT classification using MRI data from Kaggle
[40]	The Fully Resolution Convolutional Network (FrCN), the MSFO method, and the LBM	BRATS 18, BRATS 19, BRATS 20

The datasets used in the various kinds of research publications are shown in Table 2 below. Additionally, the size of the dataset, whether it was composed of 3D or 2D MRI images and the modalities of the datasets are discussed.

Table 2: Segmentation of BT MRI different datasets

Ref.	Dataset	Dataset Size	Modalities and Dimension
[30]	BRATS 2013 Dataset (3D images)	3D images: HGG - 20, LGG - 10; 2D slices - 155	MRI Modalities: T1, T2, T1c, T2FLAIR
[31]	BRATS Challenge and Leaderboard (2012-2015)	Each case has 155 slices; Cases: • 2012: LGG - 25, HGG – 25 • 2012 Synthetic: LGG - 25, HGG – 25 • 2013: LGG - 10, HGG - 20 • 2013 Leaderboard: LGG - 4, HGG – 21	---

		• 2014: LGG & HGG – 200 • 2015: LGG - 54, HGG - 220	
[32]	Brats'18 & Brats'19, Lung Segmentation, 3D-IRCADb-01	Brats'18: 285 cases (HGG - 210, LGG - 75) Brats'19: 285 cases (HGG - 210, LGG - 75) Lung Segmentation: Shenzhen dataset (336 normal, 326 abnormal CXR of TB), Chest X-ray (112,120 Frontal CXR), RSNA pneumonia detection challenge (30,000 exams) 3D-IRCADb-01: 3D CT scan (20 patients, 15 hepatic tumors)	Brats'18: Volumes from structural MRI (T1, T1Gd, T2, T2-FLAIR)
[33]	Brats 2020 (3D)	7 GB, 750 MRI scans of brain tumors Training: 484 labeled voxels Test: 266 label-free volumes	240 × 240 × 155 × 4 Modalities: FLAIR, T1w, T1gd, T2w
[34]	BraTS 2020	473 subjects, 5 images per subject	Dimension: 224 × 224 × 150 Modalities: FLAIR, T1, T1ce, T2, Seg
[35]	BraTS 2017, 2018, 2019	Brats 2017: Training - 285 cases (HGG - 210, LGG - 75), Validation - 46 cases Brats 2018: Training - 285 cases (HGG - 210, LGG - 75), Validation - 66 unlabeled cases Brats 2019: Training - 335 cases (HGG - 259, LGG - 76), Validation - 125 cases	MRI Modalities: T1, T2, T1c, T2FLAIR
[16]	BRATS Leaderboard (2012-2018)	Brats 2012: HGG - 10, LGG – 5 Brats 2013: HGG - 20, LGG – 10 Brats 2013 Leaderboard: HGG - 21, LGG – 4 Brats 2015: 384 cases (Training: HGG - 220, LGG - 54; Testing: 110 cases) Brats 2018: HGG - 191, LGG - 75	Flair, T1, (Flair + T1) MRI sequences
[11]	BRATS 2015, 2016, 2017	BRATS 2015: Training - (HGG - 220, LGG – 54), Testing - 110 cases BRATS 2016: Training - (HGG - 220, LGG - 54) BRATS 2017: Training - (HGG - 210, LGG - 72) High & Low-grade images: HGG - 130,200, LGG - 46,500 Normal & Glioma images: Glioma - 15,272, Non-glioma - 19,032	BRATS 2015 Challenge: 240 x 240 x 155 x 4 (155 slices in each scenario) Modalities: Flair, T1, T1c, T2
[36]	The Cancer Imaging Archive (TCIA)	3,929 images (Tumor: 1,373, Tumor-free: 2,556)	MRI sequences: T1 pre-contrast, FLAIR, T1 post-contrast Includes patient data (tumor grade, subtype, gender, age)
[37]	BraTS dataset	Training: 257 labeled images Testing: 5 distinct images	Dimensions: 240 × 240, 155 slices Modalities: T1, T1C, T2, FLAIR
[38]	BraTS 2019 & 2018 datasets	BraTS 2019: 462 MRI scans Training: HGG - 259, LGG – 76 Validation: 127 cases BraTS 2018: Validation - 67 MRI scans	Modalities: FLAIR, T1, T1c, T2
[39]	Kaggle dataset for BT classification	3,264 files (Training & Research set)	Classification: Pituitary, Meningioma, Glioma, Absence of tumor
[40]	Brats'18, Brats'19, Brats'20	Brats'18: HGG - 191, LGG - 75 Brats'19: 335 cases (HGG - 259, LGG - 76) Brats'20: 494 cases (Training - 369, Testing - 125)	---

The performance metrics for evaluating the MRI BT include the Dice score, sensitivity (SE), specificity (SP), accuracy (ACC), Jaccard Similarity Index (JSI), False Negative Rate (FNR), False Positive Rate (FPR), Area Under the Curve (AUC), Positive Predictive Value (PPV), Highest Density Intervals (Hdis), Variable Speed Drives (VSds), Lesion True Positive Rate (LTPR), Lesion False Positive Rate (LFPR), True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), and Precision (PREC). These metrics are presented in Table 3.

Table 3: Segmentation performance metrics in BT magnetic resonance imaging (BT MRI)

Ref	Modality	Dice	SE	SP	ACC	JSI	FNR	AUC	PPV	FPR
[30]	Complete	0.86	0.86	0.91	-	-	-	-	-	-
	Core	0.86	0.87	0.93	-	-	-	-	-	-
	Enhancing	0.88	0.90	0.94	-	-	-	-	-	-
[31]	2012	1.00	1.00	1.00	100	1.00	0.00	1.00	1.00	0.00
	2012 Synthetic	0.94	0.88	1.00	90	0.89	0.12	1.00	1.00	0.00
	2013	0.96	1.00	0.90	95	0.93	0.00	0.97	0.93	0.10
	2013 Leaderboard	1.00	1.00	1.00	100	1.00	0.00	1.00	1.00	0.00
	2014	0.98	0.98	0.96	97	0.97	0.02	0.99	0.98	0.04
	2015	0.96	0.93	1.00	95	0.93	0.07	0.96	1.00	0.00
[32]	BraTS'18	92.1	95.5	92.1	-	-	-	-	-	-
	BraTS'19	90.29	93.27	92.0	-	-	-	-	-	-
	MICCAI 2008	99.4	99.6	64.0	-	-	-	-	-	-
	Lung	98.35	97.7	95.1	98.4	-	-	-	-	-
	Liver	81.2	81.3	99.1	90.2	-	-	-	-	-
[33]	BraTS'22	0.95	-	-	-	-	-	-	-	-
[34]	FLAIR	91.23	98.53	99.14	98.95	93.9	-	-	-	-
	T1	93.86	98.97	99.68	99.41	-	-	-	-	-
	T1ce	85.67	98.72	98.52	98.68	-	-	-	-	-
	T2	79.32	98.49	98.37	98.25	-	-	-	-	-
[35]	BraTS'17	0.749	-	-	-	-	-	-	-	-
	BraTS'18	0.772	-	-	-	-	-	-	-	-
	BraTS'19	0.709	-	-	-	-	-	-	-	-
[16]	BraTS'18 (Flair)	0.96	-	-	-	-	-	-	-	-
	BraTS'15 (T1)	0.82	-	-	-	-	-	-	-	-
	BraTS'18 (T1C)	0.20	-	-	-	-	-	-	-	-
	BraTS'18 (T2)	1.00	-	-	-	-	-	-	-	-
	BraTS'18 (T1C+T2)	0.95	-	-	-	-	-	-	-	-

2.2 Deep learning

To efficiently integrate both local and global contextual information, the suggested design makes use of a deep neural network and a 3D convolutional layer, giving lower weights to each. This is achieved by the use of minuscule kernels. To overcome the data's intrinsic unpredictability, the author suggested a preprocessing method for MRI that makes use of adaptive contrast augmentation and intensity normalization. In addition, a data augmentation strategy was used to ensure that the robust 3D network was trained effectively [43]. To classify the medical images, we use a CNN feature combination with SVM. The Figshare public dataset, including MRI images of three distinct types of BT, is used to evaluate the fully automated methodology. CNN is designed to extract features from brain MRI scans. CNN characteristics are combined with a multiclass SVM for improved performance. The integrated system underwent testing and evaluation using a five-fold cross-validation procedure. With an accuracy rate for classification of 95.82%, the suggested model outperformed the most advanced method. The proposed approach is tested extensively on other brain MRI datasets to determine its efficacy [44]. An efficient DL method for multiclass brain cancer categorization is presented in this study. The Densenet201 Pre-Trained DL Model is adjusted and refined utilizing unbalanced data via deep transfer. Each tumor type is covered by the average pooling layer, which extracts the trained model's properties. Two feature selection methods are used since this layer's properties are unsuitable for exact categorization. Two methods are discussed: Entropy-Kurtosis-based High Feature Values (EKbHFV) and Metaheuristic-based Modified Genetic Algorithms (MGAs). Our threshold function improves Genetic Algorithm (GA) properties. Finally, a non-redundant serial-based technique combines and classifies EKbHFV and MGA characteristics by using a multiclass SVM cubic classifier. This method employed BRATS 2018 and BRATS 2019 without augmentation and obtained over 95% accuracy [45].

CNN multi-classifies BT for early detection. Three CNN models for classification tasks are shown. Grid

search optimization finds all CNN model hyperparameters automatically. This CNN multi-classification of BT MRI images tunes most hyper-parameters using the grid search optimizer [46]. A powerful DL strategy for BT classification utilizing many data sources is presented in this research. To begin, a nine-layer CNN model is trained to enhance contrast using a combination of hybrid division histogram equalization and ant colony optimization. Moth flame and differential evolution optimize characteristics from the second completely connected layer. Both algorithms' results are blended using matrix length and delivered to MC-SVM. In experiments, the recommended strategy obtained 99.06, 98.76, 98.18, and 94.6% accuracy on BRATS 2013, 2015, 2017, and 2018. Results indicate that greater tumor visualization before CNN model training leads to improved feature generation. Feature optimization affects processing time, whereas fusion results in increased accuracy and time [47]. To get three-dimensional brain lesions, this research expands 2D-CNNs to multimodal 3D-CNNs. It can resolve the wide neighbourhood of errors that the 2D-CNNs raw input demands while also extracting the modalities of the information differences more effectively. Next, to accelerate the network's convergence and reduce over-tipping, the true normalization layer is inserted in between the pooling and convolution layers. Finally, by using the weighted loss function, we were able to adjust the loss function and improve the feature learning in the lesion region [48].

This research divides BT into two categories using three BraTS datasets, with four 3D MRI sequences per subject in each dataset and uses two methods. As a service, they provide a hybrid model that combines 3D CNNs with long short-term memory (LSTM): TD-CNN-LSTM. On every tier of this paradigm, Time Distributed functions are embedded. The goal is to combine all four MRI scans of every patient into one input dataset since they all provide important information about the tumor. Hence, the model is fine-tuned for layer design and hyper-parameter selection via ablation studies. In the second part, they use each MRI sequence to build a 3D CNN model, which we then use to evaluate performance. Additionally, datasets undergo preprocessing for optimal performance. In tests, the TD-CNN-LSTM network achieved the best accuracy of 98.90%, surpassing 3D CNN [49]. The researchers introduced an innovative BT classification technique using DL, which operates across many grades. The methodology we employ consists of three main steps. Firstly, they utilized a CNN model to separate the tumor areas from the dataset. The second step is to improve the segmented data by adding more parameters to increase the sample size. Lastly, BT is divided into several categories by refining a pre-trained VGG-19 CNN model. By using DL and data augmentation, they enhanced the precision of the proposed technique. The findings of the trial demonstrate how well the suggested CNN-based CAD system works to support radiologists in making accurate decisions while grading multi-grade BT into four categories [50]. In this article the author uses a 3D-CNN to categorize brain MRI images into two predetermined categories. Furthermore, they suggest visualizing the 3D-CNN's behaviour using a Genetic Algorithm-based Brain Masking (GABM) method to get a new understanding of its operation. Two separate steps to the GABM method have been suggested. Firstly, the 3D-CNN is trained using brain MRI data. The second step is using a GA to identify brain regions in MRI data. The 3D-CNN primarily extracts crucial and discriminative characteristics from the knowing areas of the brain [51].

The research introduces a more advanced methodology for classifying brain tumors using MRI data. The suggested technique employed a Residual Network (ResNet), a DL architecture, to build the model. To expand the dataset and improve accuracy, we used rotating, shifting, zooming, vertical and horizontal flips, brightness, shearing, and ZCA whitening change [52]. The author introduced A CNN architecture, which then used a U-Net-based model to separate brain pictures and classify them into four distinct groups. The author used six standardized datasets to test and train the classification and segmentation models to determine how segmentation affects brain MRI tumor categorization. While testing on all six datasets, our one-of-a-kind deep-learning BT segmentation and classification model outperformed pre-trained methods [53]. By including adjacent tissues in the tumor area (ROI expansion), characteristics may become more apparent. This study utilizes pre-trained AlexNet (Alexnet, a previous deep CNN approach, significantly improves ImageNet classification [54]), ResNet-18, GoogLeNet, and ShuffleNet to extract deep characteristics from tumors and surrounding tissues. Deep features are crucial to tumor classification, but as the network gets more complicated, it may forfeit low-level information. Shallow neural networks handle fundamental data. After that, deep and superficial traits are blended to compensate for information loss [55].

They present an automated brain illness diagnosis model utilizing computer vision based on exemplars. The model produces profound characteristics by using exemplars. Choose a pre-trained DL model for generating features. The selected feature extractor is MobilNetV2. The automated brain sickness detection model consists of four stages: preprocessing, production of example deep features, Iterative Neighbourhood Component Analysis (INCA) feature selection, and SVM classification. During the processing phase, the original brain scans are scaled down to dimensions of 512 × 512 and then divided into two sets of exemplars, one with dimensions of 128 × 128 and the other with dimensions of 256 × 256. MobileNetV2 generates 1000 features from each example and scaled image. The INCA feature selector takes all of the extracted features that were created and chooses the best one. The SVM classifier uses the chosen feature as input [56]. This article presents a novel approach using a CNN, together with conventional classifiers and DL approaches, to accurately differentiate brain cancers from 2D MRI. To train the model efficiently, they have obtained a diverse collection of MRI scans that include several tumors with varying sizes, forms, locations, and levels of picture intensity. To verify the accuracy of this work, we have used the SVM classifier as well as different activation algorithms including sigmoid, softmax, and RMSProp.

We use the Python programming language, using "TensorFlow" and "Keras", to develop our recommended solution due to its efficiency and effectiveness. CNN achieved 99.74% accuracy in our work [57].

This study suggests automated categorization for fast, accurate diagnosis. This is believed to be the first two-stage deep CNN model capable of accurately classifying three distinct tumor kinds as well as typical brain cell. The best CNN models and classifiers were evaluated using a two-stage ensemble and classification method to ensure the model's correctness. The model's functionality is enhanced by a two-stage ensemble that incorporates Principal Component Analysis (PCA) and data augmentation. The two-stage ensemble boosts accuracy 4.31%. PCA reduces execution time by 6.71 and features by 18.71 [58]. For real-world use, models are inadequate as they rely on the most precise tumor model derived from slices. This research introduces a new 3 (Attention-Convolutional-LSTM) DL model for classifying BT using MRI data. All three ACL models integrate attention, convolutional, and Long Short-Term Memory (LSTM) structures into a single learning framework, using an end-to-end learning technique. Because of this, the ability to mirror the attributes was improved. Due of the three-dimensional nature of the model, the 3ACL model made direct use of three-dimensional MRIs without first converting them to two-dimensional data. Deep, very representative features are generated by the 3ACL model's fully linked layer. A set of features is sent into the SVM. The classification accuracy was further enhanced by including expected outcomes from all SVM slices into the weighted majority vote technique [59].

2.3 Machine learning

The approach proposed for BT detection produces efficient and improved results. To make the overall system more efficient, each of the various activities, such as pre-processing, segmentation, and feature extraction, among others, collaborate. The use of these pre-processing approaches allows for the achievement of favourable outcomes in segmentation, which in turn assists in the extraction of certain characteristics that are correct for classification. Attained a 95.85% accuracy rate with the use of a watershed technique combined with SVM classification [42]. The next article describes MRI brain cancer detection ML. Optimization selects traits to reduce duplication and boost relevance. Pre-processing NMF lowers brain MRI noise; Segmentation enhances tumor identification. GLCM and spatial Gray level dependence matrix (SGLDM) extract key attributes from MRI-smoothed images. Select meta-heuristic Harris Hawks Optimization (HHO) properties alone. The Knowledge-based Support Vector Machine (KSVM) classifier, which is based on ML, categorizes MRI images as either benign or malignant depending on certain factors [24].

ML-Based Back Propagation Neural Networks (MLBPNN) improve pathologists' danger location accuracy and reduce entomb onlooker variability in BT categorization. Acquisition, upgrading, division, extraction, image portrayal, characterization, and leadership are needed to prepare biopsy images for disease localization. This study analyses MLBPNN using infrared sensors. Subsystem reduction simplifies the neural distinguishing proof calculation. Fractal dimension technique uses multi-fractal detection to find the most relevant characteristics to simplify the process. The wireless infrared imaging sensor delivers tumor warming data to a specialist doctor for general health monitoring and ultrasound measurement management, particularly for elderly faraway patients [60].

2.4 Transfer learning

This paper [61] proposes a new approach for categorizing images of BT using a strategy that combines fine-tuning with TL. Utilizing a pre-trained CNN as a readily available tool for extracting features, without the need for additional training, to train a separate classification method (such as KNN, SVM, Boosted Trees, DTs, and Random Forest) is distinct from the suggested approach of TL with block-wise fine-tuning. We suggest using a TL method that makes use of a deep CNN model that has already been trained and then fine-tuning it block-wise. Using the T1-weighted contrast-enhanced MRI (CE-MRI) benchmark dataset, the suggested method is assessed. This study [62] introduces a novel tumor detection approach. The suggested approach consists of four steps: normalization, segmentation, classification, and fusion of the scores obtained from the Alex/Google DL model using MRI/CT modalities. With its score-based fusion method, the whole tumor area may be precisely segmented and classified.

2.5 Fuzzy

This study seeks to construct a brain cancer classifier using smart segmentation and classification. The recommended method incorporates data collection, preprocessing, tumor segmentation, and classification. Benchmark BT datasets are pre-processed. Here, median filtering and contrast enhancement are used. The Adaptive Fuzzy Deformable Fusion (AFDF) method integrates Fuzzy C-Means Clustering (FCM) with snake deformable approaches for segmentation. This scenario utilizes the updated Deer Hunting Optimization Algorithm (DHOA), commonly referred to as Adaptive Coefficient Vector-based DHOA, to improve crucial AFDF parameters. An ensemble classifier consisting of an autoencoder, SVM, and Deep Neural Network is suggested as a replacement for the fully connected layer in DL classification. ACV-DHOA optimizes CNN's hidden and convolutional layers [63]. This paper suggests a Neutrosophy-CNN hybrid approach. It classifies brain-image-segmented tumor regions as benign or malignant. They started by segmenting the MRI images using the Neutrosophic Set-Expert Maximum Fuzzy-Sure Entropy (NS-EMFSE) method. Segmented brain pictures were classified using CNN features using SVM and KNN classifiers. The experimental evaluation included doing a 5-fold cross-validation on a dataset consisting of 80 noncancerous tumors and 80 cancerous tumors. The research shows that CNN features surpassed many classifiers in the task of

classification [64]. As shown in Figure 3, the image that was chosen is used for the various classifications of the procedure. The procedure for classifying BT is further upon in Table 4, which includes several different approaches.

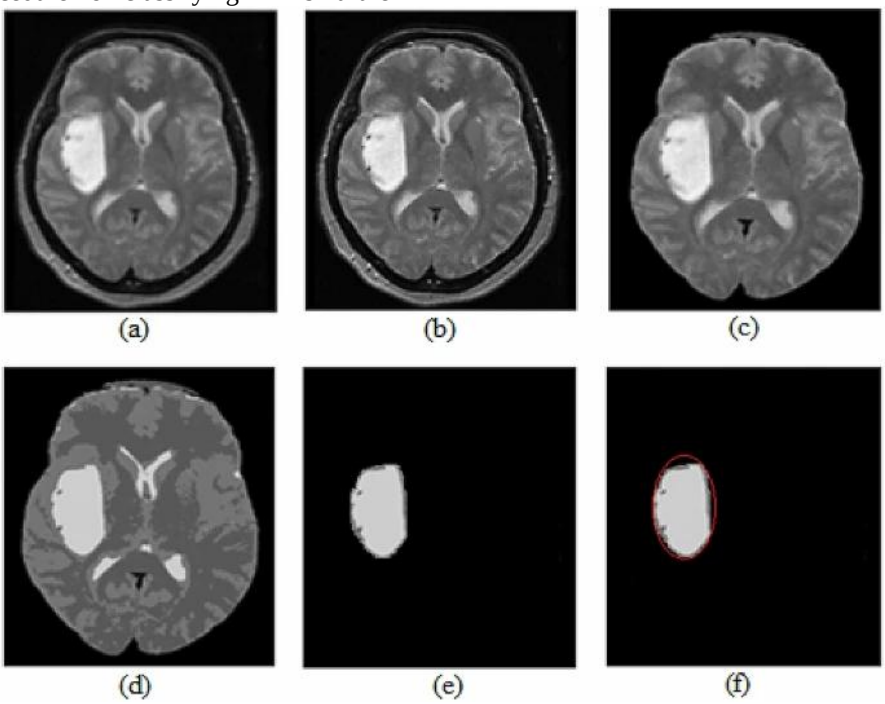


Figure 3: a) The Original Picture, b) Refined Picture c) Picture with Removed Skull, d) Image with Segmented Regions, e) Region of Tumor, f) Extracted Tumor Region [65]

Table 4: Classification of BT MRI Using Different Methods

S. No	Method	Dataset
[43]	3D CNN	BraTS 2018
[44]	CNN features with SVM	Dataset 1 (2018) - Figshare, Dataset 2 (2019) - Radiopaedia, Dataset 3 (2019) - Harvard University
[45]	Densenet201, EKbHFV, MGA	BRATS 2018, BRATS 2019
[46]	CNN with Fully Optimized Framework	RIDER, REMBRANDT, TCGA-LGG
[47]	DL Network	BraTS 2013, 2015, 2017, 2018
[48]	3D-CNN with multi-modal input	MICCAI BraTS 2018
[49]	Time Distributed-CNN-LSTM	BraTS 2018, 2019, 2020 (3D images)
[50]	Deep CNN with VGG-19 and data augmentation	Radiopaedia dataset, BT dataset
[51]	3D-CNN and GABM	ADNI (Alzheimer's Disease), ABIDE (Autism Brain Imaging Data Exchange)
[52]	ResNet	3,064 BT MRI images (2005–2010) from Nanfang Hospital (Guangzhou) & General Hospital (Tianjing Medical University)
[53]	VGG16, VGG19, EfficientNet B0, EfficientNet B7, ResNet152V2	Sets A, B, C, D
[55]	AlexNet, ResNet-18, GoogLeNet, ShuffleNet; SVM, k-NN classifiers	BT dataset, LGG-1p19q Deletion Dataset
[56]	MobileNetV2, Exemplar deep feature generator, INCA, SVM	Firat University Hospital Radiology Department dataset
[57]	CNN	BraTS 2020
[58]	Feature Extractors: Modified VGG-19, EfficientNet-B0, Inception-V3, ResNet-50, Xception; Classifiers: SVM, RF, KNN, AdaBoost, PCA	Sets 1, 2, 3, combined datasets
[59]	3ACL (Attention-Convolutional-LSTM) DL model	BraTS 2015 & 2018 benchmark datasets

[42]	SVM, KNN, Tree, Ensemble, Logistic Regression	Private dataset (512 images, Nishtar Hospital, Multan, Pakistan), Slice-based dataset (940 images)
[24]	Binomial Thresholding, GLCM, SGLDM, HHO, Kernel SVM	BraTS 2018, 2019, 2020
[60]	MLBPNN (Machine Learning-Based Back Propagation Neural Networks)	Database - Class I and Class II
[61]	Transfer Learning (TL) and Fine-Tuning (VGG19)	CE-MRI dataset
[63]	Optimized CNN with Ensemble Classification (OCNN-EC)	Kaggle dataset
[64]	NS-EMFSE-CNN, SVM, KNN	TCGA-GBM (Cancer Imaging Archive - TCIA)

3 Brain tumour MRI classification using diverse datasets

Magnetic Resonance Imaging (MRI) plays a pivotal role in the detection, classification, and analysis of brain tumors. Over the years, various datasets have been compiled to aid in the development of automated diagnostic systems. These datasets provide diverse imaging modalities, tumor classifications, and annotations that are crucial for training deep learning models. With the growing advancements in medical imaging and artificial intelligence, the availability of high-quality, labeled datasets has significantly improved brain tumor classification accuracy.

Diversity of brain tumor MRI datasets

Brain tumor MRI datasets vary significantly in terms of size, modalities, and tumor classifications. Some datasets focus on specific tumor types, while others provide a comprehensive collection of images spanning different tumor grades and subtypes.

Publicly available MRI datasets

Several large-scale publicly available datasets have been utilized in brain tumor classification research. The BraTS (Brain Tumor Segmentation) series, including BraTS 2013, 2015, 2017, 2018, 2019, and 2020, remains one of the most widely used datasets. These datasets include MRI scans from patients diagnosed with gliomas, categorized into high-grade gliomas (HGG) and low-grade gliomas (LGG). They contain multimodal imaging sequences, such as T1-weighted, T1 contrast-enhanced (T1ce), T2-weighted, and Fluid-Attenuated Inversion Recovery (FLAIR), which are essential for accurate tumor segmentation and classification. The dataset sizes vary across the years, with the latest versions containing hundreds of training and validation cases.

Other publicly available datasets, such as the Figshare BT dataset, Radiopaedia dataset, and Harvard Medical **dataset**, provide a broader spectrum of brain tumor MRI images. These datasets are particularly useful for multi-class classification tasks, where tumors such as gliomas, meningiomas, and pituitary tumors need to be distinguished. The inclusion of diverse imaging techniques across datasets enhances model generalizability.

Large-Scale MRI collections for brain tumor classification

Some datasets are designed for extensive classification tasks, containing tens of thousands of images. For example, certain datasets include over 110,000 images of gliomas spanning Grade II, III, and IV. Additionally, classification-specific datasets offer thousands of images categorized into multiple tumor types, such as glioblastoma, gliomas, meningiomas, pituitary tumors, and metastatic tumors.

Datasets with such large-scale MRI collections are invaluable for deep learning models, as they provide extensive training data that enables robust classification performance. The MICCAI BraTS 2018 dataset, for instance, contains MRI scans of 220 advanced gliomas and 54 LGG cases, captured across different modalities and resolutions. The Brats 2019 and 2020 datasets further expand this collection, providing 3D images that improve volumetric analysis and classification accuracy.

Multi-Class and multi-modal imaging in MRI classification

Advanced classification models benefit from datasets that offer multi-class labeling and multi-modal imaging. Some datasets, such as the Radiopaedia dataset and BT dataset, include detailed MRI scans categorized into different tumor grades and types. The Cancer Genome Atlas Glioblastoma Multiforme (TCGA-GBM) dataset, available in the TCIA repository, provides T1-Gadolinium (Gd) contrast-enhanced MRI sequences, further aiding in accurate tumor segmentation.

Additionally, datasets with multi-planar MRI images, such as the CE-MRI dataset, offer imaging along axial, coronal, and sagittal planes. These datasets have a 512 × 512-pixel resolution, enabling high-quality tumor identification. The availability of both 2D and 3D datasets allows researchers to experiment with different deep learning architectures, such as convolutional neural networks (CNNs) and 3D-CNN models.

3.1 Challenges in brain tumor MRI classification

Despite the availability of high-quality datasets, brain tumor MRI classification presents several challenges. Variability in tumor shapes, sizes, and locations makes classification difficult, particularly for deep learning models that rely on spatial consistency. Some datasets contain limited annotated data, requiring augmentation techniques to improve training efficiency.

Additionally, the presence of **class imbalance** in some datasets affects model performance. For instance, some datasets have significantly more glioma images than meningiomas or pituitary tumors, leading to biased predictions. Addressing these challenges requires techniques such as data augmentation, transfer learning, and advanced feature extraction.

Role of data augmentation in improving classification accuracy

Data augmentation is an essential technique in MRI-based brain tumor classification. Some datasets, such as the BT dataset, expand their dataset size through augmentation methods, increasing the total number of MRI slices from 3,064 to 91,920. This approach improves the robustness of deep learning models, ensuring better generalization to unseen tumor cases.

Augmentation techniques such as flipping, rotation, contrast adjustment, and Gaussian noise addition enhance dataset diversity. When applied to models trained on smaller datasets, these techniques significantly improve classification accuracy, sensitivity, and specificity.

Performance benchmarks in brain tumor classification

Various classification models have been evaluated using these MRI datasets. Deep learning architectures, such as CNNs, ResNet, and multi-class SVMs, have demonstrated high classification accuracy across different datasets. For instance, classification models trained on Brats 2018 and 2019 datasets achieved accuracy rates of 99.7% for HGG tumors and 98.8% for LGG tumors. Similarly, models trained on the Figshare dataset achieved accuracies exceeding 95% for meningioma, glioma, and pituitary tumor classification.

Other benchmark models, such as Naïve Bayes, K-Nearest Neighbors (KNN), Decision Trees (DT), and Multi-Class SVM, have also been evaluated on datasets like Brats 2013, 2015, 2017, and 2018. The Multi-Class SVM consistently outperformed other models, with accuracy rates above 98% in most cases.

Future directions in brain tumor MRI classification

The future of brain tumor MRI classification lies in the integration of deep learning with advanced medical imaging techniques. The use of transformer-based architectures, hybrid deep learning models, and attention mechanisms has the potential to further enhance classification accuracy. Additionally, federated learning approaches can leverage multi-center datasets while preserving patient privacy, enabling large-scale medical AI applications.

Further research is also needed to improve the interpretability of deep learning models in medical diagnostics. Explainable AI (XAI) techniques can help radiologists understand model decisions, fostering greater adoption of AI-assisted diagnostic tools in clinical settings.

3.2 3D Dataset of brain tumor MRI images

Early tumor detection has been greatly aided in recent years by 3D MRI scans. In this [34] study, The best method for BT segmentation using MRI data is CNN-based 2D U-net segmentation. This method finds regions of interest (ROIs) and trains on all MRI sequences. Normalizing and rescaling input images into 128 × 128 images reduces computing costs and maximizes efficiency. We use 2D layers to combine data. Using the BraTS 2020 brain MRI dataset, this model is trained to determine which of the four MRI sequences achieves the greatest segmentation performance based on MRI image ground truth. The greatest DSC score (93.9%) is achieved using U-Net model T1 MRI sequence training. This study [66] presents 3D Deep dResU-Net, a new method for enhancing BT segmentation accuracy using MRI data. To enhance learning, the U-Net model made use of identity mapping in the encoder, which allowed for the retention of local feature responses and their transmission to the decoder via skip connections. By addressing the issue of the fading gradient, the suggested technique hopes to enhance the training process. Using the BraTS 2020 benchmark dataset, the proposed model was assessed. To ensure the design's resilience, 50 randomly chosen patients from the BraTS 2021 dataset were employed for cross-validation. These cross-validation findings show that the model works well with the external dataset.

To automatically identify the ET, WT, and TC areas in 3D MRI BT images, the author presents a competent approach for segmentation. Positive results were obtained from the studies performed on the BraTS 2020 dataset. By including VNet, the AGSE-VNet model is enhanced. There are total of nine blocks, with five serving as encoders and four as decoders. Each decoder has an Attention Guild Filter block, whereas every encoder has an extrusion and excitation block. With consistent input/output ratios, our model can accommodate such a design without changing the network structure's size discrepancy. The SE module evaluates the model and grants the network access to global data, allowing it to choose relevant data from the enhancement channel. Then, to swiftly boost the model's performance, it uses the Attention Guild Filter block's attention method to capture dependencies. Also added was a new loss function. Change the values of the weights for the unneeded masks in the Categorical_Dice function; for example, make the background region's weight 0.1 and the tumor area of interest's weight 1. Take into consideration the difference between voxels in the background and those in the front [67]. The research study [68] presents a computerized hardware design for segmenting 3D/2D

MRI images, intending to assist professionals in identifying variations in brain tissue for diagnostic purposes. To suggest several solutions to the problems related to the fitness function, position, and velocity, we used a metaheuristic approach based on Particle Swarm Optimization (PSO). The objective is to provide a real-time automated system for segmenting MRI images that increases variables like precision, sensitivity, specificity, execution time, and resource consumption. It demonstrates an 87.97% DSC similarity and executes in a mere 4.57ms using BraTS 2013 brain MRI images. This work [69] utilizes a unique learning strategy to integrate three BT segmentation models that are mutually trained. Mutual Ensemble Learning (MEL) differs from traditional ensemble learning by sharing information across member networks using a novel loss function. Each network is trained using the whole training dataset to ensure comprehensive coverage of strong local minima. Every MEL member network surpasses the performance of the baseline single model, and the ensemble is more resilient. Our MEL has been validated by a thorough examination of two recent BT segmentation datasets. Table 7 provides an elaborate depiction of the 3D MRI BT dataset.

4 Discussion

Through the literature review of many research publications on BT Segmentation and Classification utilizing deep learning (DL) and machine learning (ML) approaches, we could gain a comprehensive grasp of BT diagnosis. The techniques used for brain tissue segmentation include the Hybrid approach using deep convolutional neural networks (DCNNs), UNet architecture, 2DGA-UNet, 3DGA-UNet, VGG-16, and multimodal MRI data. Utilising threshold segmentation, watershed approach, and classifiers for the extraction of BT. Integration of form and texture data to create a merged vector for categorization. The objective of the present research endeavours is to develop a novel architecture, named Attention Gate Residual U-Net (AGResU-Net), for the segmentation of brain tumors (BTs). The proposed architecture aims to improve feature learning and effectively process small-scale BT data by integrating attention gates with residual modules inside a U-Net framework. Achieving precise segmentation of BT regions necessitates the use of many techniques, such as CNN models with a global thresholding strategy, the Daubechies wavelet kernel, and the Partial Differential Diffusion Filter (PDDF). A deep learning approach using convolutional neural networks (CNNs), U-net, transfer learning (TL), and VGG-16 for tumor segmentation and grading enables us to get high accuracy in identifying tumors in Lower-Grade Glioma patients.

The following methods are discussed for BT classification: MLBPNN, infrared sensors, fractal dimension technique, wireless infrared imaging sensor, CNN, grid search optimization, DL strategy, fine-tuning with Temporal Learning, MobilNetV2, INCA feature selection, SVM classification, and CNN with conventional classifiers. The papers provide deep learning techniques for classifying brain tumors using convolutional neural network (CNN) models. The primary objective is to enhance the effectiveness of BT detection via segmentation, classification, and feature extraction. The proposed approach integrates intelligent segmentation, classification, and ensemble classifiers to accurately identify tumors. Non-negative Matrix Factorization (NMF) and Gray Level Co-occurrence Matrix (GLCM) are mathematical methods used to decrease noise and extract pertinent characteristics from MRI images, thereby assisting in precise tumor detection. Furthermore, the work investigates the use of Machine Learning-Based Back Propagation Neural Networks (MLBPNN) to boost diagnostic precision and decrease inconsistency in BT classification, by using infrared sensors for improved picture analysis. The manuscript presents a new transfer learning (TL) technique that utilizes a pre-trained Convolutional Neural Network (CNN) with block-wise fine-tuning to efficiently extract features and classify brain tumours. The Adaptive Fuzzy Deformable Fusion (AFDF) approach achieves improved segmentation accuracy by integrating Fuzzy C-Means Clustering with snake deformable models, which are optimized using the Deer Hunting Optimization Algorithm (DHOA). A novel hybrid Neutrosophy-CNN method is introduced, which combines an ensemble classifier with autoencoder, SVM, and Deep Neural Network to accurately categorize segmented tumor tissues as either benign or malignant.

Subsequently, this article examines the progress made in the early identification of cancers enabled by 3D MRI scans, underscoring the need for precise segmentation techniques for brain tumors detected from MRI data. A novel model, 3D Deep dResU-Net, is presented with the aim of improving the precision of segmentation. To mitigate the problem of fading gradient during training, the encoder utilizes identity mapping to preserve local feature responses. Additionally, the work explores a computerized hardware design for real-time segmentation of 3D/2D MRI images. This design incorporates a Particle Swarm Optimization (PSO) technique to improve accuracy, sensitivity, and processing speed. A novel learning technique named Mutual Ensemble Learning is presented, which combines three BT segmentation models that exchange information across different networks. This methodology guarantees thorough inclusion of local minima and enhances the overall performance.

Some limitations exist in the prior study publications, including despite its success in identifying core and improving tumor regions, the model struggles with low-intensity edema. Due to the ineffective accounting of edema in the model, this shortcoming might result in reduced accuracy in total tumor segmentation. The research uses a two-stage training technique to tackle the data imbalance problem however, this problem might still affect the model's generalizability to new data [30]. The difficulty of creating a functional neural network with fewer computational resources is recognized in the research. Improving the model's segmentation outcomes depends on its capacity to handle more datasets or more complicated pictures, both of which this constraint might

impede [38]. One limitation is that it can't handle dark images, which causes the pre-processing contrast to rise [42]. The reliability of regional forecasts made by TC and ET remains inconsistent [67]. Rather than vast valleys, deep networks may converge into small fissures [69].

.

5 Conclusion

Computer-aided approaches for tumor diagnosis using MRI data may be developed via three processes: BT detection, segmentation, and classification. These approaches provide superior precision, reduced levels of interference, and quicker processing times as compared to manual procedures. Consequently, much research has been conducted on the applications of DL and standard ML technologies. This study investigated several techniques for diagnosing brain MRI images. Additionally, a presentation was given on the comparison of datasets and performance indicators between DL and traditional ML. The overview includes a compilation of three-dimensional datasets that were acquired from a variety of research journals, as well as the approaches that were used to correlate with those datasets. The performance measurements that they have provided evidence of the best accuracy rate. Following that comes the discussion section, which provides a concise summary of the whole work and outlines the constraints of the research. Although significant advancements have been made in tumor diagnosis using MRI-based automated approaches, there are several areas for future research. One key direction is enhancing model generalization across diverse MRI datasets by incorporating transfer learning and domain adaptation techniques. Additionally, integrating multi-modal imaging data, such as combining MRI with PET or CT scans, may improve diagnostic accuracy and robustness. Further research could explore the interpretability and explainability of deep learning models to increase clinical trust and acceptance. The development of lightweight, real-time models suitable for deployment in low-resource clinical settings is another promising avenue. Finally, improving data augmentation and preprocessing techniques to handle class imbalance and noise in medical images can further refine model performance and reliability.

References

[1] J. Amin, M. Sharif, A. Haldorai, M. Yasmin, and R. S. Nayak, “Brain tumor detection and classification using machine learning: a comprehensive survey,” *Complex Intell. Syst.*, vol. 8, no. 4, pp. 3161–3183, 2022, doi: 10.1007/s40747-021-00563-y.

[2] W. Ayadi, W. Elhamzi, I. Charfi, and M. Atri, “Deep CNN for Brain Tumor Classification,” *Neural Process. Lett.*, vol. 53, no. 1, pp. 671–700, 2021, doi: 10.1007/s11063-020-10398-2.

[3] M. I. Sharif, J. P. Li, J. Amin, and A. Sharif, “An improved framework for brain tumor analysis using MRI based on YOLOv2 and convolutional neural network,” *Complex Intell. Syst.*, vol. 7, no. 4, pp. 2023–2036, 2021, doi: 10.1007/s40747-021-00310-3.

[4] M. K. Abd-Ellah, A. I. Awad, A. A. M. Khalaf, and H. F. A. Hamed, “A review on brain tumor diagnosis from MRI images: Practical implications, key achievements, and lessons learned,” *Magn. Reson. Imaging*, vol. 61, no. August 2018, pp. 300–318, 2019, doi: 10.1016/j.mri.2019.05.028.

[5] M. Nazir, S. Shakil, and K. Khurshid, “Role of deep learning in brain tumor detection and classification (2015 to 2020): A review,” *Comput. Med. Imaging Graph.*, vol. 91, no. April, 2021, doi: 10.1016/j.compmedimag.2021.101940.

[6] A. Maier, C. Syben, T. Lasser, and C. Riess, “A gentle introduction to deep learning in medical image processing,” *Z. Med. Phys.*, vol. 29, no. 2, pp. 86–101, 2019, doi: 10.1016/j.zemedi.2018.12.003.

[7] K. R. Bhatele and S. S. Bhadauria, “Brain structural disorders detection and classification approaches: a review,” *Artif. Intell. Rev.*, vol. 53, no. 5, pp. 3349–3401, 2020, doi: 10.1007/s10462-019-09766-9.

[8] M. I. Sharif, J. P. Li, J. Naz, and I. Rashid, “A comprehensive review on multi-organs tumor detection based on machine learning,” *Pattern Recognit. Lett.*, vol. 131, pp. 30–37, 2020, doi: 10.1016/j.patrec.2019.12.006.

[9] S. S. Begum and D. R. Lakshmi, “Combining optimal wavelet statistical texture and recurrent neural network for tumour detection and classification over MRI,” *Multimed. Tools Appl.*, vol. 79, no. 19–20, pp. 14009–14030, 2020, doi: 10.1007/s11042-020-08643-w.

[10] J. Amin, M. Sharif, M. Raza, T. Saba, R. Sial, and S. A. Shad, “Brain tumor detection: a long short-term memory (LSTM)-based learning model,” *Neural Comput. Appl.*, vol. 32, no. 20, pp. 15965–15973, 2020, doi: 10.1007/s00521-019-04650-7.

[11] T. Saba *et al.*, “Brain tumor detection using fusion of hand crafted and deep learning features,” *Cogn. Syst. Res.*, vol. 59, pp. 221–230, 2020, doi: 10.1016/j.cogsys.2019.09.007.

[12] J. Amin, M. Sharif, M. Yasmin, and S. L. Fernandes, “A distinctive approach in brain tumor detection and classification using MRI,” *Pattern Recognit. Lett.*, vol. 139, pp. 118–127, 2020, doi: 10.1016/j.patrec.2017.10.036.

[13] M. Sharif, J. Amin, M. W. Nisar, M. A. Anjum, N. Muhammad, and S. Ali Shad, “A unified patch-based method for brain tumor detection using features fusion,” *Cogn. Syst. Res.*, vol. 59, pp. 273–286, 2020, doi: 10.1016/j.cogsys.2019.10.001.

[14] M. I. Sharif, J. P. Li, M. A. Khan, and M. A. Saleem, “Active deep neural network features selection for segmentation and recognition of brain tumors using MRI images,” *Pattern Recognit. Lett.*, vol. 129, pp. 181–189, 2020, doi: 10.1016/j.patrec.2019.11.019.

[15] M. Sharif, J. Amin, M. Raza, M. Yasmin, and S. C. Satapathy, “An integrated design of particle swarm optimization (PSO) with fusion of features for detection of brain tumor,” *Pattern Recognit. Lett.*, vol. 129, pp. 150–157, 2020, doi: 10.1016/j.patrec.2019.11.017.

[16] J. Amin, M. Sharif, N. Gul, M. Yasmin, and S. A. Shad, "Brain tumor classification based on DWT fusion of MRI sequences using convolutional neural network," *Pattern Recognit. Lett.*, vol. 129, pp. 115–122, 2020, doi: 10.1016/j.patrec.2019.11.016.

[17] J. Chang *et al.*, "A mix-pooling CNN architecture with FCRF for brain tumor segmentation," *J. Vis. Commun. Image Represent.*, vol. 58, pp. 316–322, 2019, doi: 10.1016/j.jvcir.2018.11.047.

[18] K. R. M. Fernando and C. P. Tsokos, "Deep and statistical learning in biomedical imaging: State of the art in 3D MRI brain tumor segmentation," *Inf. Fusion*, vol. 92, no. August 2022, pp. 450–465, 2023, doi: 10.1016/j.inffus.2022.12.013.

[19] A. Elazab *et al.*, "GP-GAN: Brain tumor growth prediction using stacked 3D generative adversarial networks from longitudinal MR Images," *Neural Networks*, vol. 132, pp. 321–332, 2020, doi: 10.1016/j.neunet.2020.09.004.

[20] M. Mazzei, "A Change Detection with Machine Learning Approach for Medical Image Analysis," *Lect. Notes Electr. Eng.*, vol. 810 LNEE, no. December, pp. 203–229, 2023, doi: 10.1007/978-981-16-6775-6_18.

[21] W. Ayadi, W. Elhamzi, I. Charfi, and M. Atri, "A hybrid feature extraction approach for brain MRI classification based on Bag-of-words," *Biomed. Signal Process. Control*, vol. 48, pp. 144–152, 2019, doi: 10.1016/j.bspc.2018.10.010.

[22] M. Puttagunta and S. Ravi, "Medical image analysis based on deep learning approach," *Multimed. Tools Appl.*, vol. 80, no. 16, pp. 24365–24398, 2021, doi: 10.1007/s11042-021-10707-4.

[23] K. R. M. Fernando and C. P. Tsokos, "Deep and statistical learning in biomedical imaging: State of the art in 3D MRI brain tumor segmentation," *Inf. Fusion*, vol. 92, no. November 2022, pp. 450–465, 2023, doi: 10.1016/j.inffus.2022.12.013.

[24] C. Sundar and R. K. Karunakara, "Efficient Detection and Classification of Brain Tumor using Kernel based SVM for MRI," *Multimed. Tools Appl.*, pp. 7393–7417, 2022, doi: 10.1007/s11042-021-11821-z.

[25] J. Amin, M. Sharif, M. Yasmin, T. Saba, and M. Raza, "Use of machine intelligence to conduct analysis of human brain data for detection of abnormalities in its cognitive functions," *Multimed. Tools Appl.*, vol. 79, no. 15–16, pp. 10955–10973, 2020, doi: 10.1007/s11042-019-7324-y.

[26] J. Jayashree and S. A. Kumar, "Linear Discriminant Analysis Based Genetic Algorithm with Generalized Regression Neural Network – A Hybrid Expert System for Diagnosis of Diabetes," *Program. Comput. Softw.*, vol. 44, no. 6, pp. 417–427, 2018, doi: 10.1134/S0361768818060063.

[27] A. Kabir Anaraki, M. Ayati, and F. Kazemi, "Magnetic resonance imaging-based brain tumor grades classification and grading via convolutional neural networks and genetic algorithms," *Biocybern. Biomed. Eng.*, vol. 39, no. 1, pp. 63–74, 2019, doi: 10.1016/j.bbe.2018.10.004.

[28] J. Jayashree and S. Ananda Kumar, "Hybrid swarm intelligent redundancy relevance (RR) with convolution trained compositional pattern neural network expert system for diagnosis of diabetes," *Health Technol. (Berl).*, vol. 10, no. 1, pp. 197–206, 2020, doi: 10.1007/s12553-018-00291-3.

[29] J. Jayashree and S. Ananda Kumar, "Evolutionary correlated gravitational search algorithm (ECGS) with genetic optimized Hopfield neural network (GHNN) – A hybrid expert system for diagnosis of diabetes," *Meas. J. Int. Meas. Confed.*, vol. 145, pp. 551–558, 2019, doi: 10.1016/j.measurement.2018.12.083.

[30] S. Sajid, S. Hussain, and A. Sarwar, "Brain Tumor Detection and Segmentation in MR Images Using Deep Learning," *Arab. J. Sci. Eng.*, vol. 44, no. 11, pp. 9249–9261, 2019, doi: 10.1007/s13369-019-03967-8.

[31] J. Amin *et al.*, "Brain Tumor Detection by Using Stacked Autoencoders in Deep Learning," *J. Med. Syst.*, vol. 44, no. 2, 2020, doi: 10.1007/s10916-019-1483-2.

[32] A. Kaur, L. Kaur, and A. Singh, *GA-UNet: UNet-based framework for segmentation of 2D and 3D medical images applicable on heterogeneous datasets*, vol. 33, no. 21. Springer London, 2021. doi: 10.1007/s00521-021-06134-z.

[33] Z. Shaukat, Q. ul A. Farooq, S. Tu, C. Xiao, and S. Ali, "A state-of-the-art technique to perform cloud-based semantic segmentation using deep learning 3D U-Net architecture," *BMC Bioinformatics*, vol. 23, no. 1, pp. 1–21, 2022, doi: 10.1186/s12859-022-04794-9.

[34] S. Montaha, S. Azam, A. K. M. Rakibul Haque Rafid, M. Z. Hasan, and A. Karim, "Brain Tumor Segmentation from 3D MRI Scans Using U-Net," *SN Comput. Sci.*, vol. 4, no. 4, pp. 1–10, 2023, doi: 10.1007/s42979-023-01854-6.

[35] J. Zhang, Z. Jiang, J. Dong, Y. Hou, and B. Liu, "Attention Gate ResU-Net for Automatic MRI Brain Tumor Segmentation," *IEEE Access*, vol. 8, pp. 58533–58545, 2020, doi: 10.1109/ACCESS.2020.2983075.

[36] M. A. Naser and M. J. Deen, "Brain tumor segmentation and grading of lower-grade glioma using deep learning in MRI images," *Comput. Biol. Med.*, vol. 121, no. February, 2020, doi: 10.1016/j.compbiomed.2020.103758.

[37] G. Karayegen and M. F. Aksahin, "Brain tumor prediction on MR images with semantic segmentation by using deep learning network and 3D imaging of tumor region," *Biomed. Signal Process. Control*, vol. 66, no. January, 2021, doi: 10.1016/j.bspc.2021.102458.

[38] J. Sun, Y. Peng, Y. Guo, and D. Li, "Segmentation of the multimodal brain tumor image used the multi-pathway architecture method based on 3D FCN," *Neurocomputing*, vol. 423, pp. 34–45, 2021, doi: 10.1016/j.neucom.2020.10.031.

[39] P. Agrawal, N. Katal, and N. Hooda, "Segmentation and classification of brain tumor using 3D-UNet

deep neural networks," *Int. J. Cogn. Comput. Eng.*, vol. 3, no. August 2021, pp. 199–210, 2022, doi: 10.1016/j.ijcce.2022.11.001.

[40] V. V. S. Sasank and S. Venkateswarlu, "An automatic tumour growth prediction-based segmentation using full resolution convolutional network for brain tumour," *Biomed. Signal Process. Control*, vol. 71, no. April 2021, 2022, doi: 10.1016/j.bspc.2021.103090.

[41] J. Liu *et al.*, "IOUC-3DSFCNN: Segmentation of Brain Tumors via IOU Constraint 3D Symmetric Full Convolution Network with Multimodal Auto-context," *Sci. Rep.*, vol. 10, no. 1, pp. 1–15, 2020, doi: 10.1038/s41598-020-63242-x.

[42] H. Habib, R. Amin, B. Ahmed, and A. Hannan, "Hybrid algorithms for brain tumor segmentation, classification and feature extraction," *J. Ambient Intell. Humaniz. Comput.*, vol. 13, no. 5, pp. 2763–2784, 2022, doi: 10.1007/s12652-021-03544-8.

[43] H. Mzoughi *et al.*, "Deep Multi-Scale 3D Convolutional Neural Network (CNN) for MRI Gliomas Brain Tumor Classification," *J. Digit. Imaging*, vol. 33, no. 4, pp. 903–915, 2020, doi: 10.1007/s10278-020-00347-9.

[44] S. Deepak and P. M. Ameer, "Automated Categorization of Brain Tumor from MRI Using CNN features and SVM," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 8, pp. 8357–8369, 2021, doi: 10.1007/s12652-020-02568-w.

[45] M. I. Sharif, M. A. Khan, M. Alhussein, K. Aurangzeb, and M. Raza, "A decision support system for multimodal brain tumor classification using deep learning," *Complex Intell. Syst.*, vol. 8, no. 4, pp. 3007–3020, 2022, doi: 10.1007/s40747-021-00321-0.

[46] E. Irmak, "Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework," *Iran. J. Sci. Technol. - Trans. Electr. Eng.*, vol. 45, no. 3, pp. 1015–1036, 2021, doi: 10.1007/s40998-021-00426-9.

[47] M. I. Sharif, J. P. Li, M. A. Khan, S. Kadry, and U. Tariq, "M3BTCNet: multi model brain tumor classification using metaheuristic deep neural network features optimization," *Neural Comput. Appl.*, vol. 36, no. 1, pp. 95–110, 2024, doi: 10.1007/s00521-022-07204-6.

[48] M. Li, L. Kuang, S. Xu, and Z. Sha, "Brain Tumor Detection Based on Multimodal Information Fusion and Convolutional Neural Network," *IEEE Access*, vol. 7, pp. 180134–180146, 2019, doi: 10.1109/ACCESS.2019.2958370.

[49] S. Montaha, S. Azam, A. K. M. R. H. Rafid, M. Z. Hasan, A. Karim, and A. Islam, "TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study," *IEEE Access*, vol. 10, pp. 60039–60059, 2022, doi: 10.1109/ACCESS.2022.3179577.

[50] M. Sajjad, S. Khan, K. Muhammad, W. Wu, A. Ullah, and S. W. Baik, "Multi-grade brain tumor classification using deep CNN with extensive data augmentation," *J. Comput. Sci.*, vol. 30, pp. 174–182, 2019, doi: 10.1016/j.jocs.2018.12.003.

[51] H. Shahamat and M. Saniee Abadeh, "Brain MRI analysis using a deep learning based evolutionary approach," *Neural Networks*, vol. 126, pp. 218–234, 2020, doi: 10.1016/j.neunet.2020.03.017.

[52] S. Ali, A. Ismael, A. Mohammed, and H. Hefny, "Arti fi cial Intelligence In Medicine An enhanced deep learning approach for brain cancer MRI images classi fi cation using residual networks," vol. 102, no. December 2019, 2020.

[53] A. Akter *et al.*, "Robust clinical applicable CNN and U-Net based algorithm for MRI classification and segmentation for brain tumor," *Expert Syst. Appl.*, vol. 238, no. October 2023, 2024, doi: 10.1016/j.eswa.2023.122347.

[54] J. Vaishnavi, S. Ravi, and A. Anbarasi, "An efficient adaptive histogram-based segmentation and extraction model for the classification of severities on diabetic retinopathy," *Multimed. Tools Appl.*, vol. 79, no. 41–42, pp. 30439–30452, 2020, doi: 10.1007/s11042-020-09288-5.

[55] C. Öksüz, O. Urhan, and M. K. Güllü, "Brain tumor classification using the fused features extracted from expanded tumor region," *Biomed. Signal Process. Control*, vol. 72, no. September 2020, 2022, doi: 10.1016/j.bspc.2021.103356.

[56] A. Kursad Poyraz, S. Dogan, E. Akbal, and T. Tuncer, "Automated brain disease classification using exemplar deep features," *Biomed. Signal Process. Control*, vol. 73, no. December 2021, 2022, doi: 10.1016/j.bspc.2021.103448.

[57] A. Chattopadhyay and M. Maitra, "MRI-based brain tumour image detection using CNN based deep learning method," *Smart Agric. Technol.*, vol. 2, no. 4, 2022, doi: 10.1016/j.neuri.2022.100060.

[58] N. F. Aurna, M. A. Yousuf, K. A. Taher, A. K. M. Azad, and M. A. Moni, "A classification of MRI brain tumor based on two stage feature level ensemble of deep CNN models," *Comput. Biol. Med.*, vol. 146, no. April, 2022, doi: 10.1016/j.compbiomed.2022.105539.

[59] F. Demir, Y. Akbulut, B. Taşcı, and K. Demir, "Improving brain tumor classification performance with an effective approach based on new deep learning model named 3ACL from 3D MRI data," *Biomed. Signal Process. Control*, vol. 81, no. October 2022, 2023, doi: 10.1016/j.bspc.2022.104424.

[60] P. M. Shakeel, T. E. E. L. Tobely, H. Al-feel, G. Manogaran, and S. Baskar, "Neural Network Based Brain Tumor Detection Using Wireless Infrared Imaging Sensor," *IEEE Access*, vol. 7, pp. 5577–5588, 2019, doi: 10.1109/ACCESS.2018.2883957.

[61] Z. N. K. Swati *et al.*, "Brain tumor classification for MR images using transfer learning and fine-tuning," *Comput. Med. Imaging Graph.*, vol. 75, pp. 34–46, 2019, doi: 10.1016/j.compmedimag.2019.05.001.

[62] J. Amin, M. Sharif, M. Yasmin, T. Saba, M. A. Anjum, and S. L. Fernandes, "A New Approach for

Brain Tumor Segmentation and Classification Based on Score Level Fusion Using Transfer Learning," *J. Med. Syst.*, vol. 43, no. 11, 2019, doi: 10.1007/s10916-019-1453-8.

[63] M. Y. B. Murthy, A. Koteswararao, and M. S. Babu, "Adaptive fuzzy deformable fusion and optimized CNN with ensemble classification for automated brain tumor diagnosis," *Biomed. Eng. Lett.*, vol. 12, no. 1, pp. 37–58, 2022, doi: 10.1007/s13534-021-00209-5.

[64] F. Özyurt, E. Sert, E. Avci, and E. Dogantekin, "Brain tumor detection based on Convolutional Neural Network with neutrosophic expert maximum fuzzy sure entropy," vol. 147, 2019.

[65] C. Narmatha, S. M. Eljack, A. A. R. M. Tuka, S. Manimurugan, and M. Mustafa, "A hybrid fuzzy brain-storm optimization algorithm for the classification of brain tumor MRI images," *J. Ambient Intell. Humaniz. Comput.*, no. 0123456789, 2020, doi: 10.1007/s12652-020-02470-5.

[66] R. Raza, U. Ijaz Bajwa, Y. Mehmood, M. Waqas Anwar, and M. Hassan Jamal, "dResU-Net: 3D deep residual U-Net based brain tumor segmentation from multimodal MRI," *Biomed. Signal Process. Control*, vol. 79, no. P1, p. 103861, 2023, doi: 10.1016/j.bspc.2022.103861.

[67] X. Guan *et al.*, "3D AGSE-VNet: an automatic brain tumor MRI data segmentation framework," *BMC Med. Imaging*, vol. 22, no. 1, pp. 1–18, 2022, doi: 10.1186/s12880-021-00728-8.

[68] F. Hamdaoui and A. Sakly, "Automatic diagnostic system for segmentation of 3D/2D brain MRI images based on a hardware architecture," *Microprocess. Microsyst.*, vol. 98, no. February, 2023, doi: 10.1016/j.micpro.2023.104814.

[69] J. Hu, X. Gu, and X. Gu, "Mutual ensemble learning for brain tumor segmentation," *Neurocomputing*, vol. 504, pp. 68–81, 2022, doi: 10.1016/j.neucom.2022.06.058.

Tourism Consumer Sentiment Analysis Using a Multi-layer Memory Network Combining Temporal Convolutional and BiLSTM Architectures

Kun Yang[1]*, Wei Yang[2]
[1]School of Tourism Management, Wuhan Business University, Wuhan, 430000, China
[2]Department of Social Sciences, Quanzhou Medical College, Quanzhou, 362000, China
E-mail: Y7yangk@163.com
*Corresponding author

Keywords: multilayer memory networks, text sentiment analysis, tourism, bi-directional long-short term memory, temporal convolutional networks

Received: January 22, 2025

Traditional sentiment analysis methods for tourism review texts often suffer from a strong dependence on vocabulary quality and insufficient ability to capture contextual temporal features, making it difficult to accurately identify sentiment tendencies in complex contexts. To address this issue, this study proposes a multi-layer memory network algorithm that integrates temporal convolutional networks (TCN) and bidirectional long short-term memory (BiLSTM) to improve the depth of feature extraction and the accuracy of emotion classification. A dataset of 30,000 manually labeled tourism review texts was constructed, with sentiment labels categorized into positive, neutral, and negative. Experimental results showed that the proposed model achieved an accuracy of 95.78%, an F1 score of 95.98%, and a training time of only 40.18 seconds in sentiment classification tasks. Ablation studies demonstrated that the model exhibited clear advantages in structural integration and regularization mechanisms, with stronger stability. Sentiment orientation analysis showed that positive reviews account for 74.8% on average, and the model effectively identified multi-emotional expressions, indicating strong practicality and generalization ability. In summary, the proposed model achieves excellent performance in both feature extraction and sentiment classification, and achieves high generalization and training efficiency. Its application to tourism consumer review analysis can provide reliable technical support for the precise management of tourist attractions and user feedback analysis, while also offering new insights for the advancement of sentiment analysis methods.

Povzetek: Članek predstavi novo večplastno spominsko arhitekturo, ki združuje TCN in BiLSTM za učinkovito razpoznavanje čustev v turističnih ocenah z izboljšano kontekstualno zaznavo in računsko učinkovitostjo.

1 Introduction

The booming development of the tourism industry provides consumers with more and more choices in terms of tourism products and tourism services, and consumers often refer to the evaluations and feedbacks of other consumers before making purchase decisions [1]. Therefore, tourism consumer evaluation text has become an important source of information for tourism enterprises to understand consumer needs, improve services, and formulate marketing strategies. The text sentiment analysis (TSA) technology can automatically recognize the emotional tendency of the text and categorize it, thus helping tourism enterprises to quickly understand consumer feedback and make corresponding adjustments [2].

Currently, there are more categories of TSA techniques, and common methods include machine learning, deep learning, etc., in addition to traditional rule-based methods, and methods based on sentiment dictionaries. Among these algorithms, deep learning techniques have a wide range of applications in TSA [3]. Tasks such as text understanding and sentiment analysis that require long-range dependency modeling often benefit from optimized memory network structures. Multi-layer memory networks (MMNs) are an extension of traditional recurrent neural networks, such as long short-term memory (LSTM), by introducing stacked layers to improve the model's ability to capture long-range dependencies [4]. Although these models have strong expressiveness, they often suffer from long training times, high data requirements, and complex hyper-parameter tuning processes [5].

To address these limitations, this study proposes a lightweight MMN architecture that combines a temporal convolutional network (TCN) with a bidirectional long short-term memory (BiLSTM) network. TCN provides strong parallel computational capabilities and effectively extracts local sequential features, while BiLSTM captures global contextual dependencies and improves semantic

understanding of complex emotional expressions. The integration of both networks not only mitigates problems such as gradient vanishing, but also improves both model depth and training efficiency. In this research, the TCN-BiLSTM model is applied to the sentiment analysis of tourism consumer reviews, with the aim of improving the classification accuracy and computational efficiency in TSA tasks, and providing a high-performance technical solution for practical applications.

There are four primary sections to the research. An outline of the state of domestic and international research on TSA techniques is provided in the first section. The development of the study's algorithm model constitutes the second section. The third part is the validation of the performance of the algorithm. The fourth section summarizes the research work.

2 Related works

With the continuous development of tourism, tourism enterprises also pay more and more attention to consumer feedback. Consumer reviews are one of the important reference indicators for tourism enterprises to adjust their management strategies. Another crucial area of study in the fields of data mining and natural language processing is the TSA of tourist consumer evaluation [6]. Tourism evaluation TSA aims to extract the emotional tendency as well as the expression of intention in the text by analyzing the text, and MMNs have a high degree of adaptability in this application [7]. The commonly used algorithms for MMNs include LSTM, attention mechanisms, etc., which have many applications in various fields. Most of the researchers have improved MMNs, including optimization of different algorithms as well as combinations. To improve the accuracy of sentiment classification, Ramaswamy S L and Chinnappan J combined convolutional neural network (CNN) algorithm and LSTM algorithm to empirically investigate various ensemble models. It was also compared with other traditional sentiment classification methods to evaluate the accuracy of CNN-LSTM algorithm for sentiment analysis. According to the findings, the CNN-LSTM algorithm performs sentiment analysis with good accuracy [8]. To improve the accuracy and stability of stream flow prediction, Wan W et al. proposed a model combining two-layer TCN and LSTM algorithms while incorporating a multi-head attention mechanism and verifying the model generalization. The experimental results indicated that this algorithm had less error in traffic prediction and had high accuracy and stability [9]. Li C et al. combined TCN with LSTM for the estimation of the health state as well as the remaining lifetime of the battery. The results of simulation experiments indicated that the TCN-LSTM algorithm had a high degree of fit, accurate prediction results, and robustness [10]. He et al. proposed the SSAE-TCN-BiLSTM model in an attempt to capture the temporal nuances of network traffic, which reduced the feature dimensionality by SSAE and performed feature extraction with TCN and BiLSTM algorithms. The results indicated that this algorithm model was computationally efficient, converged quickly, and had a large improvement in accuracy [11]. For accurate groundwater level prediction, Mohammad E et al. proposed a model combining self-attentive TCN and LSTM algorithms to solve the problem of vanishing gradient and overcome the limitations of LSTM. The results showed that this algorithm could effectively capture the long and short-term dependencies in the time series, with a small overall error and good algorithm performance [12].

Numerous studies have been conducted on TSA by researchers both domestically and overseas. To interpret the emotions and opinions in the text, Rengarjan et al. adopted the CNN-RNN algorithm model and utilized AGTO for complex feature extraction and selection. Experimental results indicated that this method could effectively perform text feature extraction and perform sentiment analysis, improving feature recognition [13]. To extract video pop-up information, Zhou X and Li W proposed a local CNN-based pop-up text sub-recognition algorithm, which reconstructed the data into a data lookup table and formatted the content for output. The results indicated that the text classification and recognition of this method was better and advanced [14]. Wang L et al. proposed a multi-feature text data enhancement model, M-DA, in order to learn text information more deeply. The study reconstructed and analyzed text sequences and captured long-term dependencies between sequences with BiLSTM networks. The outcomes revealed that the modeling framework constructed in this way could effectively extract text features, improve TSA, and increase accuracy [15]. By suggesting an NBLex-based method to assess sentiment in Kannada-English code-switched text, Chundi et al. increased TSA's efficiency and decreased the amount of time needed to identify the sentiment. The results of the study indicated that this method was effective in improving the efficiency of TSA and did not require training samples, reducing the cost of TSA [16]. To analyze the comments on social media, Taneja K et al. used SA techniques for language processing. The study proposed an unsupervised learning method using a transformer architecture for performing SA on a dataset. The results of the study revealed that this method had high F1 and AUC scores and was feasible to be used for product recommendation and thus influencing business decisions [17]. Table 1 presents a structured comparison of existing sentiment analysis methods based on different algorithms, application domains, data sets, performance results, and attention limitations. This comparison highlights the research gap that the TCN-BiLSTM model proposed in this study addresses in the context of travel sentiment analysis.

Table 1: Summary of related sentiment analysis methods

Study	Method	Domain	Dataset	Performance	Limitations
Ramaswamy & Chinnappan [8]	CNN-LSTM	Sentiment analysis	General review texts	Good accuracy for sentiment classificationt	Limited modeling of long-term dependenciest
Wan et al. [9]	TCN+LSTM+Multi-head Attention	Streamflow prediction	Time series data	High accuracy, low prediction error, good stability	Not applied in text sentiment domain
Li C et al. [10]	TCN+LSTM	Streamflow prediction	Battery lifecycle data	High fit, accurate and robust predictions	Non-textual domain, no NLP validation
He Z et al. [11]	SSAE-TCN-BiLSTM	Network traffic analysis	Time series intrusion data	High computation efficiency, fast convergence	Not designed for textual sentiment
Mohammad E et al.[12]	Self-attention TCN+LSTM	Groundwater level prediction	Environmental time series	Models short- and long-term dependencies, low error	Not applicable to sentiment analysis
Rengarjan et al. [13]	CNN-RNN+AGTO	Literary sentiment analysis	Novel “Immortals of Meluha”	Effective feature extraction and sentiment recognition	Limited generalization beyond literary texts
Zhou X & Li W [14]	Local CNN	Video popup text classification	Popup texts from video	Better classification performance	Not focused on emotion polarity
Wang L et al. [15]	M-DA+BiLSTM	Chinese TSA	Multi-feature Chinese datasets	Improved accuracy through feature enhancement	Model complexity increases computation load
Chundi R et al. [16]	NBLex (Lexicon-based)	Code-switched sentiment analysis	Kannada-English mixed corpus	Efficient, cost-effective, no training data needed	Dependent on lexicon quality, low adaptability
Taneja K et al. [17]	Transformer (unsupervised)	Social media review analysis	E-commerce reviews	High F1 and AUC, usable in recommendation	Complex architecture, weaker interpretability

In summary, MMNs exhibit strong performance in feature classification and analysis tasks, and show high applicability in tourism sentiment analysis. However, existing TSA methods often face challenges such as limited modeling depth and inefficient training. This study proposes a novel MMN architecture that integrates TCN with BiLSTM. Rather than replacing LSTM or attention mechanisms, the proposed model builds on their ability to capture long-term dependencies while leveraging TCN to improve local feature extraction and parallel processing efficiency. As a result, the TCN-BiLSTM framework provides a lighter and more efficient solution for sentiment classification. Given its relatively limited application in the tourism domain, the model demonstrates both novelty and feasibility. Moreover, it is expected to provide new methodological insights for TSA research.

3 TSA approach based on multi-layer memory networks

3.1 Algorithmic modeling of multi-layer memory networks

To address the challenges of complex emotional expressions, diverse sentence structures, and frequent sentiment shifts in tourism review texts, this study proposes a TSA method based on a MMN. Based on the hypothesis that local features should be extracted in parallel, while global semantics require bidirectional modeling, the method integrates TCN with BiLSTM, providing improved contextual awareness and sentiment classification accuracy compared to existing approaches. When dealing with noisy or ambiguous sentiment expressions, the model leverages BiLSTM to capture contextual dependencies from both directions, improving robustness under uncertain emotional content. Compared to Transformer-based models, the proposed method achieves a better trade-off between computational efficiency and classification accuracy without requiring pre-training, making it suitable for real-time applications or resource-constrained environments.

MMNs is a network architecture that adds memory mechanisms to traditional neural networks, which is suitable for tasks requiring long-term memory and reasoning ability, such as question-answer systems in natural language processing, text understanding, and so on. Commonly used algorithms for MMNs include recurrent neural networks, LSTM, and attention mechanisms [18]. However, TSA needs to analyze and identify the sentiment features (SFs) contained in the text, and then classify them. Based on this consideration, this study takes BiLSTM network as the basis, combines TCN, constitutes MMNs algorithmic model, and carries out sentiment analysis on the text of tourism consumer evaluation. BiLSTM provides bi-directional modeling capabilities, allowing it to capture both past and future contextual dependencies. This makes it particularly suitable for analyzing complex tourism reviews that contain emotional shifts, negations, or mixed sentiments, enabling more accurate extraction and identification of key SFs.

To analyze the sentiment of the text, it is first necessary to vectorize the text content. Text vectorization is the process of transforming text data into digital signals, and commonly used algorithms include BERT dynamic word vector language model, Glove, Word2vec, and so on. Among them, Word2vec can capture the complex relationship between words and includes two architectures, CBOW and Skip-gram [19-20]. The computational process of CBOW is shown in Equation (1).

$$p(W \mid context) = p(w_t \mid w_{t-c}, w_{t-c+1}, \cdots, w_{t-1}, w_{t+1}, w_{t+c+1}, \cdots, w_{t+c}) \tag{1}$$

$$p(context \mid w_t) = p(w_{t-c}, w_{t-c+1}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+c+1}, \cdots, w_{t+c} \mid w_t) \tag{2}$$

$$p(w_{t+n} \mid w_t) = soft\max(v_{w_t}^T v_{t+n}^{'}) = \frac{\exp(v_{w_t}^T v_{t+n}^{'})}{\sum_{w' \in v} \exp(v_{w_t}^T v_{w'}^{'})} \tag{3}$$

In Equation (1), W represents the vocabulary. w_t represents the target word of the current time step t. c represents the context window size. w_{t-c} through w_{t+c} are the context terms. *context* context refers to the sequence of words composed of c words on both sides of the target word, which is used to predict the conditional probability of the current word. As demonstrated in Equation (2), Skip-gram predicts the context by the target word, whereas CBOW primarily predicts the target word by the provided context.

In Equation (2), the Skip-gram can make a judgment on the probability of the occurrence of the glossary content in the text before and after the word based on the provided target word, as shown in Equation (3).

In Equation (3), v_w represents the vector representation of the input word in the input embedding matrix. $v_w{}'$ represents the vector representation of the prediction target word in the output embedding matrix. n denotes the proximity between the specified word and the current input word. The training sample w is contained in the vocabulary list v. The Skip-gram model can represent the relationship between different words with fewer word vector dimensions, and at the same time, it can take into account the contextual information to effectively extract the semantic information in the text. Then, the text data is processed by convolutional network. The convolutional kernel of the CNN can extract the data features of the text within the convolutional layer (CL), as shown in Equation (4).

$$x_i^k = f(\sum_{j \in N_i} x_j^{k-1} * w_{j_i}^k + b_i^k) \tag{4}$$

In Equation (4), x_i^k and x_j^{k-1} represent the output of the i neuron in layer k and the j neuron in layer $k-1$, respectively. $w_{j_i}^k$ is the weight parameter connecting these two neurons, b_i^k represents the bias term, and $f(\cdot)$ is the activation function. N_i represents the index set of the neurons of the previous layer connected by the i-th neuron in the k layer. $j \in N_i$ represents the traversal and weighted summation of all the neurons of the previous layer that have connections with this neuron. After the CL computes the features, the next step is the pooling layer (PL) to fuse the features. The PL simultaneously collects the best features and minimizes feature dimensions to prevent overfitting. The formula for processing text data by PL is shown in Equation (5).

$$x_i^k = f(\beta_i^k D(x_i^{k-1}) + b_i^k) \tag{5}$$

In Equation (5), $D(x_i^{k-1})$ represents the pooling operation, and according to the task needs, the maximum pooling method or average pooling method can be selected to process the text information. After the data is processed by the CL and PL, the fully connected layer will combine the local information extracted from them to map the features into the sample labeling space, highly purify the features, and input them to the classifier. The TCN algorithm used in the study has an additional feature compared to the normal CNN, which is to deform the latter so that it can handle temporal problems. This improvement can give the CNN the ability to perform text sequence prediction. In this case, the causal convolution enables the algorithm to have a causal property, which is calculated as shown in Equation (6).

$$P(x) = \prod_{t=1}^{T} P(x_t \mid x_1, x_2, \cdots, x_{t-1}) \tag{6}$$

In Equation (6), x_t represents the current input value. T represents the total length of the input sequence. That is, the number of terms in the text or word sequence, which is used to control the modeling range of the model in the time dimension. It follows that the current information can only be predicted by the previous input values. Since the feature extraction is limited to a certain range, the study adds dilation convolution to compensate for this limitation as shown in Equation (7).

$$f_{k_d} = (d-1) \times (f_{k-1}) + f_k \tag{7}$$

In Equation (7), d is the number of voids. Dilation convolution is equivalent to inserting d zeros into the convolutional network, and the text sequence is sampled and computed every certain d. The addition of dilation convolution increases the number of network layers. However, to prevent gradient vanishing due to excessive depth, residual links are necessary, as illustrated in Equation (8).

$$R = x + F(x) \tag{8}$$

Equation (8) is the process of linear transformation of the inputs to each layer. The TCN network algorithm proposed in the study is shown in Equation (9).

$$\begin{cases} y_i = Conv(W_i \cdot F_j + b_i) \\ \{Y_0, Y_1, \cdots, Y_t\} = Hi - norm(\{y_0, y_1, \cdots, y_t\}) \\ \{T_0, T_1, \cdots, T_t\} = \mathrm{Re}\, Lu(\{Y_0, Y_1, \cdots, Y_t\}) \end{cases} \tag{9}$$

In Equation (9), y_i represents the state values. W_i is the word matrix. F_j represents the convolution kernel of the j th layer. b_i is the bias matrix. The final resulting text sequence features are then $\{T_0, T_1, \cdots, T_t\}$. In order to make the overall algorithm able to learn the before and after associations, the study uses BiLSTM algorithm to optimize the TSA process. Its network structure is shown in Figure 1.

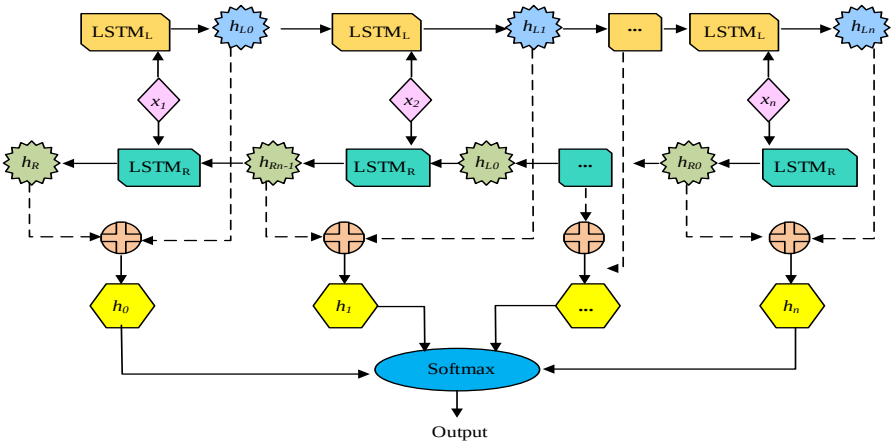


Figure 1: Schematic diagram of BiLSTM network structure

In Figure 1, BiLSTM constitutes an acyclic graph, which is able to obtain the output value (OV) through the text information before and after, and has higher stability. Splicing the text features in both directions can extract the dependencies of the text sequences and obtain the SFs. The text sequence feature in the forward direction is $\vec{H} = \{h_{L1}, h_{L2}, \cdots, h_{Lt}\}$, and the text sequence in the reverse direction is $\overleftarrow{H} = \{h_{R1}, h_{R2}, \cdots, h_{Rt}\}$. The vector of splicing the two directions is shown in Equation (10).

$$H = \{\vec{H}, \overleftarrow{H}\} = \{h_1, h_2, \cdots, h_t\} \tag{10}$$

The text features captured by BiLSTM are further fused as shown in Equation (11).

$$H(x) = Temp(x) + Bi(x) \tag{11}$$

In Equation (11), $Temp(x)$ is the OV of TCN. $Bi(x)$ is the OV of the BiLSTM network layer. The combination of the two is the final output. Summarizing the algorithms used in the study, the MMNs algorithm model is constructed as shown in Figure 2.

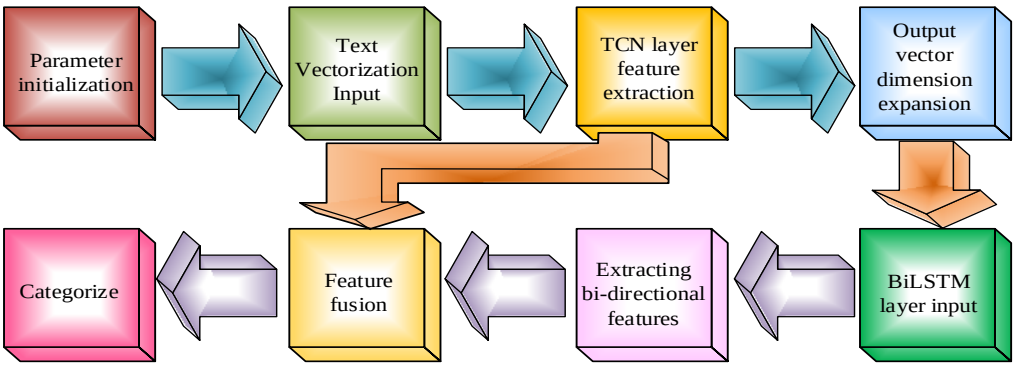


Figure 2: Model of TCN-BiLSTM multi-layer memory network algorithm

In Figure 2, the first step of the proposed TCN-BiLSTM MMNs algorithmic process lies in the initialization of the text data. Moreover, then the data is transmitted to the input layer of the TCN network, which outputs the vector information. Next, the data is used for final classification and TSA after the features are further extracted and fused using the BiLSTM method.

3.2 TSA method based on TCN-BiLSTM algorithm

TSA is a kind of computer technology to recognize and analyze the expression of emotion in text. It mainly includes four aspects: extraction of sentiment information, identification of subjectivity and objectivity of text, determination of sentiment tendency, and measurement of sentiment intensity. The analysis methods include methods based on deep learning as well as methods based on emotion features and emotion lexicon [21-22]. Figure 3 displays the details.

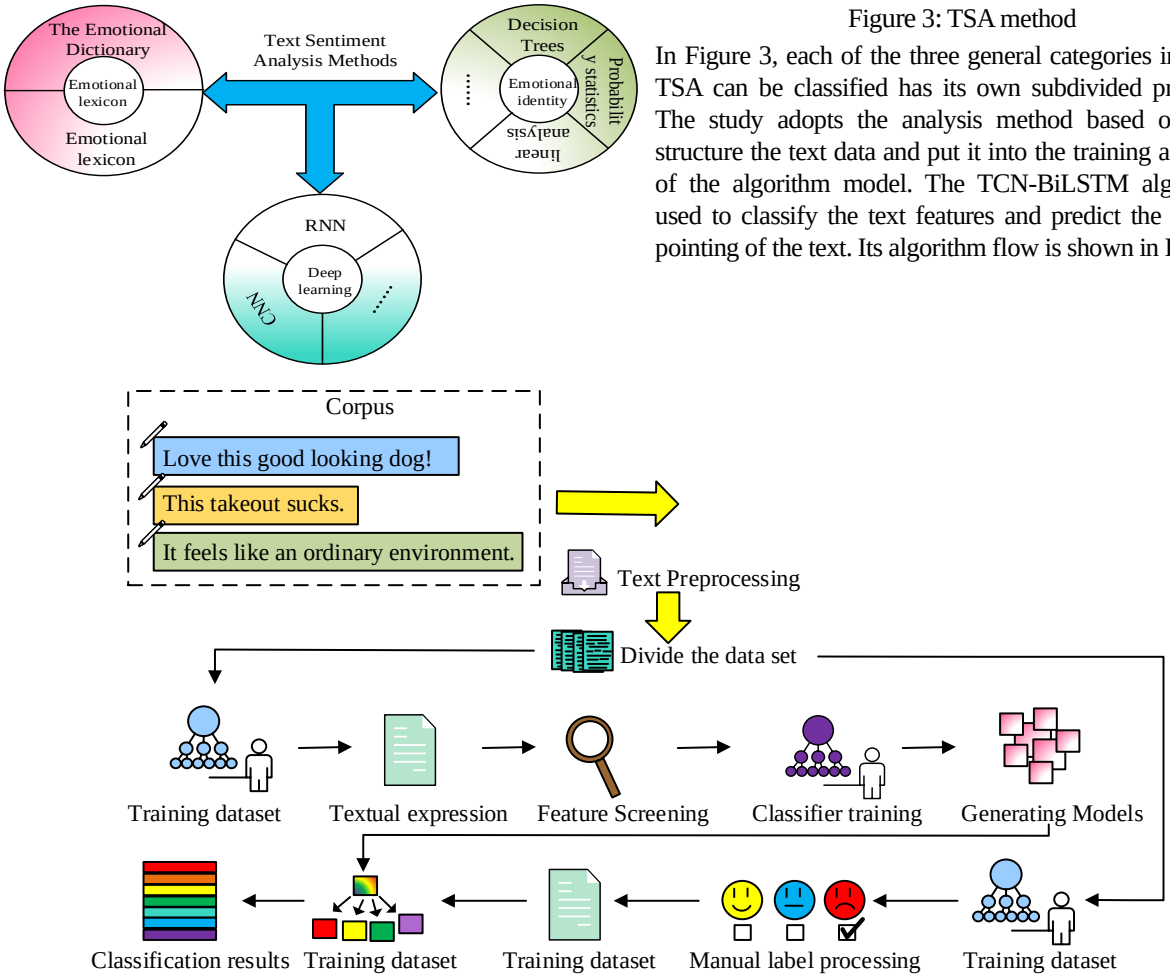


Figure 3: TSA method

In Figure 3, each of the three general categories into which TSA can be classified has its own subdivided procedures. The study adopts the analysis method based on SFs to structure the text data and put it into the training and testing of the algorithm model. The TCN-BiLSTM algorithm is used to classify the text features and predict the sentiment pointing of the text. Its algorithm flow is shown in Figure 4.

Figure 4: Flow of text analysis based on sentiment features

In Figure 4, the first step of text analysis based on SFs lies in preprocessing the text information and filtering and processing the features of the text by dividing the two datasets, training and testing. Finally, the classification model is generated to achieve the effect of classifying the textual SFs. For this purpose, the text needs to be preprocessed to represent the text in a computerized way, and the features are screened and classified. Among them, there are many methods commonly used for the screening of text features, such as the mutual information method. The mutual information approach may sort the frequency of texts based on the size of the MI value and measure the text's emotional inclination. Equation (12) displays the calculating formula.

$$MI(w_i, c_j) = \sum_{w \in W} \sum_{c \in C} P(w_i, c_j) \log \frac{P(w_i, c_j)}{P(w_i)P(c_j)} \tag{12}$$

In Equation (12), C represents the category of the document. W denotes all words in the dataset. P represents the probability. w_i is the feature item. c_j represents the text category of the item. $MI(w_i, c_j)$ then represents the mutual information value between the feature item and the category. Low-frequency features are more informative while high-frequency features are less informative. Another method used is information gain, which evaluates the discriminative power of textual features based on their information gain with respect to the target sentiment labels. By ranking features according to their computed information gain scores, features with low scores-indicating limited contribution to classification-are removed, while more informative features are retained for subsequent modeling. Equation (13) illustrates the computation procedure.

$$IG(t_j) = -\sum_{i=1}^{n} P(c_i) \log_2^{P(c_i)} + P(t_j) \sum_{i=1}^{n} P(c_i \mid t_j) \log_2^{P(c_i \mid t_j)} + P(\bar{t}_j) \sum_{i=1}^{n} P(c_i \mid \bar{t}_j) \log_2^{P(c_i \mid \bar{t}_j)} \tag{13}$$

In Equation (13), c_i is the text type and the feature term is t_j. The probability that t_j exists in c_i is represented by $P(c_i \mid t_j)$ and $P(c_i \mid \bar{t}_j)$, respectively.

Through the information gain method, the important feature terms can be extracted quickly, and it has a wider application in dealing with problems that use the same set of features. It

is worth noting that in the feature selection stage, this study employs mutual information and information gain methods, which are primarily used to preliminarily filter and rank the high-dimensional term features in the original review texts. These two methods are commonly used supervised feature evaluation techniques in text classification, capable of measuring the correlation between individual features (terms) and the target labels (sentiment categories). Therefore, it can effectively reduce input dimensionality, eliminating redundant information, and improving training efficiency. Considering that the focus of this study is on constructing a high-performance sentiment classification model architecture, the purpose of feature selection is to improve modeling efficiency and generalization ability, rather than model interpretability. Therefore, feature evaluation methods that focus on interpretability, such as SHapley Additive exPlanations (SHAP) and local interpretable model-agnostic explanations (LIME), are not introduced in this study.

To effectively classify the extracted features in the TCN-BiLSTM algorithm model, the study uses Softmax function for classification as shown in Equation (14).

$$P(y=k \mid x)=\frac{\exp(w_k^T x+b_k)}{\sum_{k=1}^{k}\exp(w_k^T x+b_k)} \tag{14}$$

In Equation (14), the number of categories is denoted by k. w is the weight. b is the bias. The larger the calculation result is, the greater the possibility that the feature item belongs to the category. Through this method, text features can be effectively classified, which is conducive to further TSA. Combining the TCN-BiLSTM algorithm and the computational flow of TSA, the proposed MMNs algorithm can be summarized as Figure 5. The proposed MMNs algorithm can be summarized as Figure 5.

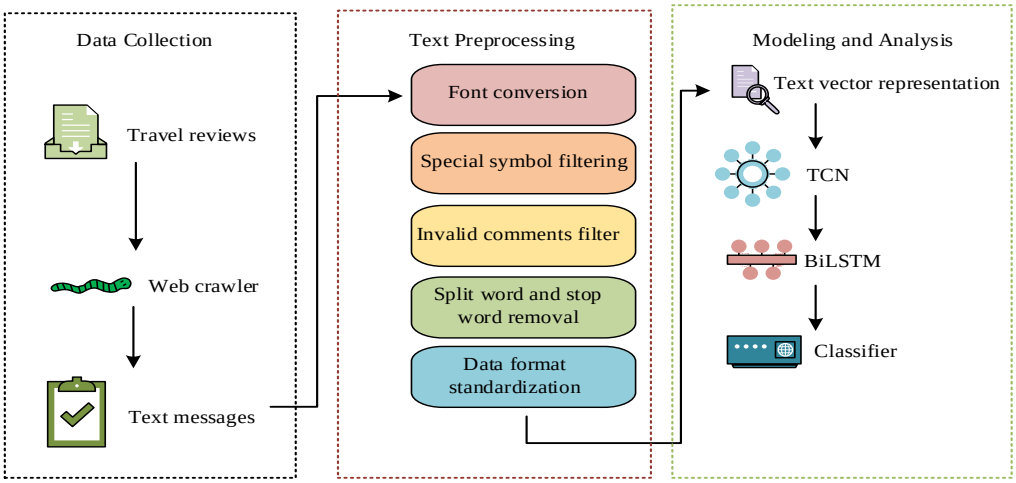


Figure 5: TSA model based on TCN-BiLSTM

The study's suggested MMNs algorithm model is broken down into three sections in Figure 5: text processing, data collecting, and memory network-based text analysis. The evaluation text of tourism consumers is extracted and converted into standard text data information by formatting the text information. Moreover, it is introduced into the TCN-BiLSTM MMNs model to represent the text as vectors. Then the vector information is output from the TCN network, and the text features are extracted and analyzed by the BiLSTM network, and finally output to the classifier to realize the classification of text SFs.

4 Tourism consumer evaluation TSA results

4.1 Validation of modeling effects of multi-layer memory networks algorithms

The study initially creates an experimental platform to confirm the overall performance of the algorithm model before examining its impact on the TSA of tourism assessment in order to confirm the impact of the suggested algorithm on the sentiment analysis of consumer evaluation. The dataset used in this study is collected using a Python-based web scraping tool from major tourism platforms such as Ctrip and Mafengwo, focusing on user reviews of popular tourist attractions in Hunan Province. To ensure data quality, preprocessing steps includes HTML tag removal, noise filtering, Chinese word segmentation, stopword removal, and word vector transformation. A total of 30,000 review samples are collected and sentiment labeled using a combination of keyword-based rules and manual correction. Of these, 15,000 are labeled positive, 4,000 are labeled neutral, and 11,000 are labeled negative. To reduce the impact of class imbalance, the sample proportions in the training set are adjusted accordingly, while the test set retained the original distribution. Given the subjective nature of user reviews, the consistency of manual annotations is verified through cross-validation to increase label reliability. Table 2 displays the specifications of the experimental platform.

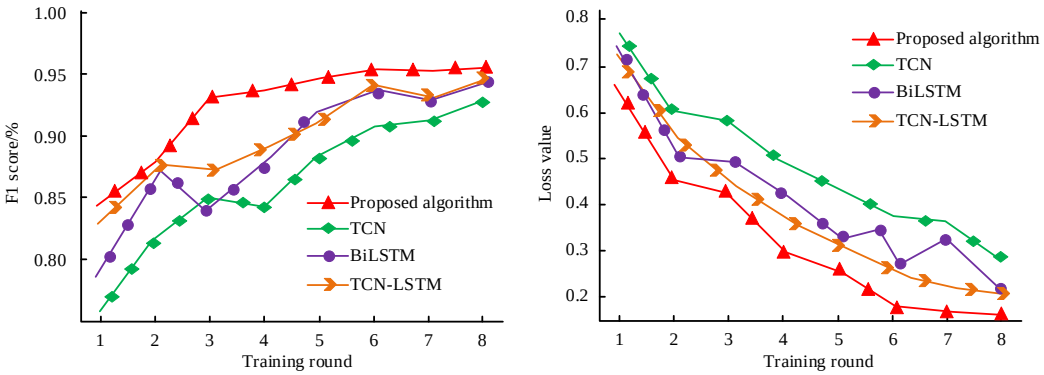
Table 2: Parameter settings of the experimental platform

Factors	Setting
Operating system	Windows 10
Processor	IntelXeon(R) E5
Graphics card	RTX3090
Host memory	64GB
Experimental frame	TensorFlow
Learning rate	0.001
Dropout value	0.5
Word vector size	210
Number of hidden layer units	128
BatchSize	128
Epoch	20

In Table 2, during the experiment, the model is run on Windows 10 operating system, the hardware platform includes Intel Xeon E5 series processor, RTX 3090 graphics card, and 64GB host memory, and the deep learning framework adopts TensorFlow. To ensure the performance stability and convergence efficiency of the model, the main hyper-parameters are experimentally pre-tuned in this study. The learning rate is initially set to 0.001 and tested in the range {0.0005, 0.001, 0.002}, and 0.001 is finally selected as the optimal value of the model for this task. The dropout rate is set to 0.5 to avoid overfitting. The word vector dimension is set to 210, chosen after comparing the performance in {128, 210, 300}. The number of hidden layer units is 128, the batch size is 128, and the number of training rounds (EPOchs) is 20. The above parameters show stable performance in multi-group combined tests, taking into account training efficiency and model effect. Considering that this study focuses on model structure design and comparative evaluation, the selection of hyper-parameters follows the principle of combining performance first with empirical verification.

The performance of the algorithms suggested in the study is confirmed using the experimental parameter settings mentioned above, and the trends of F1 and loss values of several methods are contrasted. Figure 6 displays the findings.

In Figure 6(a), although all the algorithmic models are able to stabilize the F1 scores as the training rounds increases, the TCN-BiLSTM algorithm proposed in the study is able to achieve the results faster. According to the change trend, it can be realized that the TCN-BiLSTM algorithm, besides being able to achieve a higher F1 score in a shorter time, also has stronger stability, faster convergence and higher accuracy. Figure 6(b) shows that the loss value of the proposed algorithm is the lowest when the iterations reaches 8, which indicates that the proposed algorithm has a higher fit and feature learning ability. Table 3 shows the average performance indicators of each model in the emotion classification task of travel reviews.



(a)Trend of F1 scores for different algorithms (b)Trend of loss value for different algorithms

Figure 6: Trend of F1 score and loss value of different algorithms

Table 3: Performance data for each algorithm model

Algorithm	Accuracy/%	Recall rate/%	F1 score/%	Accuracy 95% CI	Recall 95% CIt	F1 95% CI
TCN-BiLSTM	95.46±0.17	95.34±0.19	95.25±0.21	[95.25, 95.67]	[95.12, 95.56]	[95.04, 95.46]
TCN	92.36±0.22	94.26±0.23	94.89±0.24	[92.09, 92.63]	[94.00, 94.52]	[94.61, 95.17]
BiLSTM	93.28±0.19	94.17±0.21	94.41±0.23	[93.07, 93.49]	[93.96, 94.38]	[94.16, 94.66]
TCN-BiLSTM (without Dropout)	94.63±0.21	94.51±0.22	94.47±0.23	[94.40, 94.86]	[94.28, 94.74]	[94.24, 94.70]

TCN-BiLSTM (without BatchNorm)	94.81±0.19	94.33±0.20	94.59±0.22	[94.61, 95.01]	[94.13, 94.53]	[94.37, 94.81]
TCN-LSTM	94.59±0.18	93.29±0.22	94.65±0.20	[94.39, 94.79]	[93.03, 93.55]	[94.45, 94.85]

Table 3 presents a comparison of the model performance under different structural configurations and regularization settings. The full TCN-BiLSTM model achieves the best overall results, with an accuracy of 95.46%, recall of 95.34%, and an F1 score of 95.25%. In contrast, the TCN-only model achieves a recall of 94.26% but has a lower accuracy of 92.36%, indicating strong local feature extraction but weaker classification precision. The BiLSTM-only model improves accuracy to 93.28% and achieves an F1 score of 94.41%, showing better performance in modeling global context, but with limited ability to capture short-term dependencies.

From a structural integration perspective, the TCN-BiLSTM model effectively combines local and global representations, resulting in superior performance in sentiment classification tasks. Further analysis on regularization shows that removing the dropout layer reduces the F1 score to 94.47%, while removing the batch normalization yields an F1 score of 94.59%. Both results are lower than the full model, suggesting that regularization plays a key role in improving stability and generalization. Overall, the ablation study confirms the effectiveness of each module and supports the rationality of the proposed model design.

The performance of TCN-LSTM is relatively balanced, the accuracy rate is 94.59%, and the F1 score is 94.65%. However, due to the unidirectional LSTM structure, there is still a problem of insufficient understanding of bidirectional semantics. Compared with TCN-LSTM, TCN-BiLSTM performs better in emotion classification tasks, mainly due to the bidirectional modeling ability of BiLSTM, which can simultaneously capture the dependency between the preceding and following sentences. However, the LSTM in TCN-LSTM is a one-way structure that can only be modeled forward, and its comprehension ability is limited for texts with emotional transition, contrast, or summary sentence patterns. For example, when confronted with comments such as "the scenic area environment is good, but there are too many people and too much noise," TCN-LSTM is susceptible to being dominated by the positive emotions present in the first paragraph and disregarding the negative tendency exhibited in the second half of the sentence. This results in an emotional judgment bias. In addition, in texts containing negative constructs or mood reversals, such as "The hotel was supposed to be good, but the service attitude let me down," the CN-LSTM often struggled to effectively identify the dominance of the subsequent emotional intensity. Therefore, in terms of structural design, TCN-BiLSTM is more suitable for dealing with common complex sentences, multi-emotional sentences, and semantic contrast sentences in tourism texts. Moreover, it has stronger discriminative ability and adaptability. In addition, the training efficiency of the comparison algorithms is shown in Figure 7.

In Figure 7, the average training time (ATT) of the TCN-BiLSTM algorithm model is 43.8 seconds. The ATT for the TCN and BiLSTM algorithms, on the other hand, is 45.32 seconds and 48.29 seconds, respectively. The ATT for TCN-LSTM is 45.12 seconds. In comparison, TCN-BiLSTM has higher algorithmic performance.

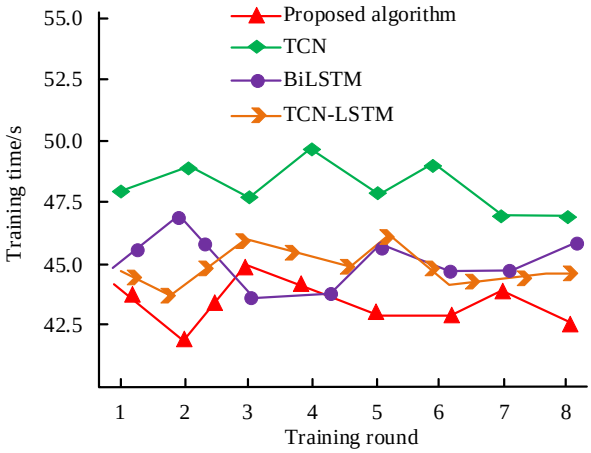


Figure 7: Comparison of model training efficiency of different algorithms

This is because the TCN-BiLSTM algorithm can utilize the CL of TCN to achieve parallel computing, which has a higher utilization of computational resources, thus increasing the speed of the model to process data. From the above data, it can be concluded that the TCN-BiLSTM algorithm model shows higher training efficiency and training accuracy on the training data, and better performance on text data processing, which can be applied to tourism evaluation TSA. Next, the study will verify the effect of this algorithm for TSA.

The pre-trained language models based on the Transformer architecture, such as BERT, RoBERTa, and XLNet, have made significant progress in TSA tasks. Moreover, these are widely considered to be the current mainstream SOTA models. To further verify the effectiveness of the proposed TCN-BiLSTM model, this study selects three representative models, BERT-base, RoBERTa-base, and XLNet-base. It also conducts comparative experiments on the same tourism consumer review dataset to evaluate their classification performance and training efficiency. The results are shown in Table 4.

Table 4: Performance comparison between the proposed model and the Transformer model

Algorithm	Accuracy/%	Recall rate/%	F1 score/%	Average training time (s)
TCN-BiLSTM	95.78	95.62	95.98	40.18
BERT-base	94.80	94.63	94.62	89.14
RoBERTa-base	95.10	94.81	94.08	93.46
XLNet-base	94.48	94.72	94.13	97.02

In Table 4, although Transformer-based models perform well in sentiment classification tasks, their training times are generally longer and they require more computational resources. For example, the training time of XLNet-base reaches 97.02 seconds, while RoBERTa-base and BERT-base require 93.46 seconds and 89.14 seconds, respectively. In contrast, the proposed TCN-BiLSTM model achieves higher accuracy (95.78%) and F1 score (95.98%) with a significantly lower training time of only 40.18 seconds, demonstrating superior computational efficiency and practical value.

Furthermore, Transformer models typically rely on large pre-trained corpora and complex hyper-parameter tuning, which pose higher barriers to domain adaptation. In comparison, the MMN architecture proposed in this study requires no pre-training and can achieve excellent classification performance on medium-sized tourism review datasets. This makes it particularly suitable for resource-constrained application environments, such as mobile tourism applications and real-time review monitoring systems on online travel platforms. Thus, the TCN-BiLSTM model achieves a well-balanced trade-off between performance and computational cost, offering strong potential for real-world applications.

4.2 TSA validation

Sentiment analysis of tourism review text typically involves three aspects: sentiment polarity classification, sentiment intensity analysis, and sentiment dimension identification. Sentiment polarity refers to the overall emotional orientation expressed in the text-positive, negative, or neutral-and serves as a key indicator of the consumer's overall tourism experience. Sentiment intensity reflects the degree or strength of the emotion conveyed, while sentiment dimension refers to the specific aspects being evaluated, such as service quality, scenic environment, or price experience. The study uses Python technology for web crawling to obtain reviews of famous tourist attractions in Hunan, and analyzes the evaluation text sentiment with TCN-BiLSTM model. The results are shown in Figure 8.

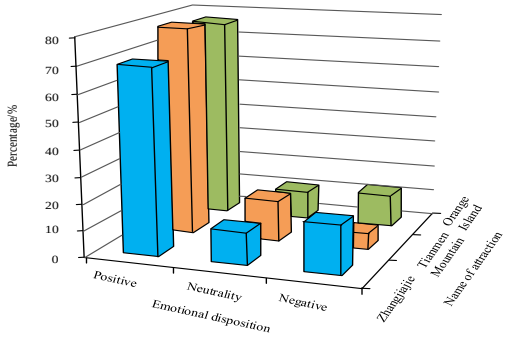
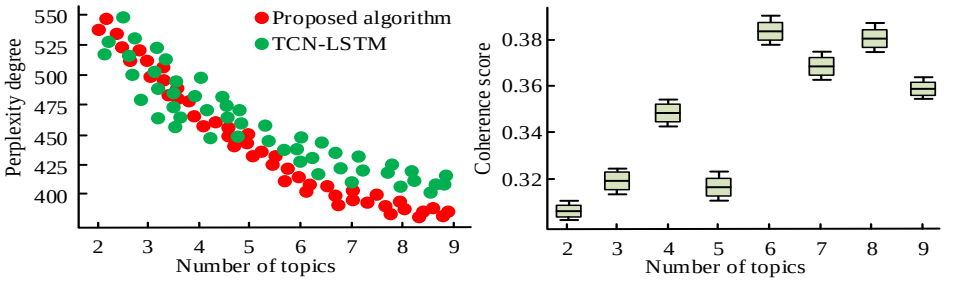


Figure 8: Sentiment analysis results of tourism text

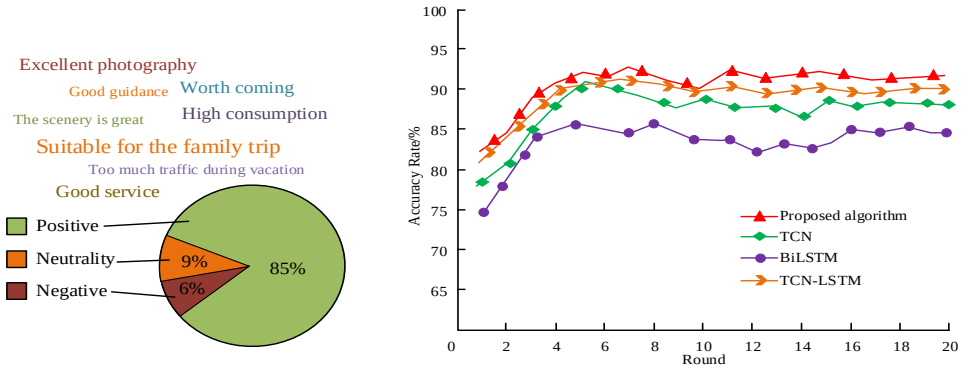
In Figure 8, the ratings of the three famous tourist attractions are generally dominated by positive ratings. The average proportion of positive evaluations is 74.8%, the average proportion of neutral evaluations is 12.9%, and the average proportion of negative evaluations is 12.3%. The sentiment tendency of the evaluations may be accurately analyzed by the TCN-BiLSTM algorithm model, it can be concluded. This is because the TCN algorithm can effectively extract local as well as global features of the time series to capture the emotional changes of the text, while the BiLSTM algorithm is able to do bi-directional information processing, which is able to more comprehensively understand the emotional expression of tourism evaluation. According to TSA results, the analysis effect is superior and the TCN-BiLSTM model has a reasonable level of validity. The extraction effect is further verified, as shown in Figure 9.

In Figure 9(a), the text theme extraction effect is mainly indicated by the perplexity degree. The lower the perplexity degree, the more accurate the effect of theme extraction. A thorough comparison of the text theme extraction effects of various algorithms reveals that the TCN-BiLSTM model has the best text theme extraction impact when the number of themes approaches nine. This is because the model has the lowest perplexity. However, if the number of themes is too high, the model is prone to overfitting. Figure 9(b) illustrates the relationship between the number of topics and topic coherence. The coherence score reaches its peak when the number of topics is 6, suggesting that six is the optimal number of latent topics in the review dataset. Summarizing the Hunan tourism consumer reviews, the proportion of their evaluation sentiment categories and the algorithm fitting effect are shown in Figure 10.



(a)Plot of number of topics against perplexity (b)Plot of number of themes against coherence

Figure 9: Topic number selection based on perplexity and coherence analysis



(a)Emotional cloud maps and statistics (b)Algorithm accuracy comparison

Figure 10: Sentiment analysis results and algorithm accuracy

In Figure 10(a), the high-frequency words calculated by the TCN-BiLSTM model constitute a sentiment cloud map, and its judgments of positive, negative, and neutral for evaluations are summarized as a fan statistic map. Combined with the model accuracy in Figure 10(b) for a comprehensive judgment, the TCN-BiLSTM model proposed in the study has a certain degree of accuracy in sentiment analysis, and can effectively differentiate between tourism evaluations of different sentiments. On the final data derived, the OV of the TCN-BiLSTM model is close to the real value. The tourism evaluation algorithm built on the basis of TCN-BiLSTM can effectively analyze the sentiment trend of tourism evaluation, which facilitates the efficient and accurate management of tourist attractions.

5 Discussion

The experimental results demonstrated that the proposed TCN-BiLSTM model outperformed traditional deep learning architectures in sentiment classification tasks for tourism review texts. This performance gain was mainly attributed to the combination of TCN's ability to extract local and hierarchical temporal features and BiLSTM's strength in modeling bidirectional contextual dependencies. This fusion enabled the model to better capture emotional shifts, polarity reversals, and nuanced sentiments that were commonly found in tourism-related user-generated content.

Despite its advantages, the model had certain limitations. First, while the TCN-BiLSTM architecture was more efficient than transformer-based models, it still required relatively long training times compared to simpler models such as standard CNN or LSTM. Second, the model was less effective when dealing with long texts where dependencies span multiple paragraphs, or when sentiment was implicit or highly contextual. In addition, scalability to very large datasets or multilingual scenarios remained an open challenge.

With regard to domain applicability, the TCN-BiLSTM model exhibited potential for transfer to other sentiment analysis tasks, including product reviews, social media comments, and service feedback, provided that the text manifested analogous sequential and emotional characteristics. However, domain adaptation and retraining would be necessary to maintain accuracy.

Compared to the most advanced transformer-based models such as BERT, RoBERTa and XLNet, the TCN-BiLSTM model achieves higher classification accuracy with significantly lower training costs and computational requirements. This makes it particularly suitable for real-time, resource-constrained applications such as mobile travel applications or online review monitoring systems. However, for tasks involving complex sentiment structures or fine-grained sentiment classification, Transformer models may still offer advantages due to their stronger global attention mechanisms and pre-trained knowledge representations.

6 Conclusion

To enhance the accuracy of tourism evaluation TSA and to

realize the sustainable development of tourism consumption program, the study proposed a multi-layer neural network model based on TCN-BiLSTM algorithm. This model mainly processed the input data information by TCN algorithm and then outputs the vector information. The features were extracted by BiLSTM algorithm, and feature fusion and classification were performed, and finally TSA was performed. The outcomes of the study indicated that the TCN-BiLSTM algorithm had a test accuracy of 95.46% and an F1 score of 95.25%. Compared with other traditional algorithms, its accuracy and F1 score were improved by 2.05% and 0.6%, respectively. In terms of the efficiency of model training, the ATT of TCN-BiLSTM was 43.8 s, which was more efficient than the traditional TCN and BiLSTM algorithms with an average improvement of 2.44 s. The ATT of TCN-BiLSTM was 43.8 s, which was more efficient than the traditional TCN and BiLSTM algorithms. Meanwhile, the TCN-BiLSTM algorithm also had a lower loss rate with higher fit and feature learning ability. The TCN-BiLSTM algorithm also had the lowest perplexity when analyzing the topics of tourism evaluation, which further proved the superiority of the TCN-BiLSTM algorithm in the analysis of tourism evaluation. Meanwhile, after the actual analysis of the tourism evaluation text, it can be concluded that the TCN-BiLSTM algorithm could effectively distinguish the emotional tendency of the text with high accuracy. In summary, TCN-BiLSTM has good performance in sentiment analysis of tourism evaluation text. With its high accuracy rate and high algorithm training efficiency, it has good performance in practical applications. Nevertheless, further enhancements can be made to this model. In subsequent studies, the incorporation of an attention mechanism may be a valuable avenue for exploring the dynamic allocation of attention to data of varying importance, with the aim of further optimising the model's performance.

However, this study specifically targets sentiment analysis in the tourism domain, and the dataset and model design are tailored to the linguistic and emotional characteristics of tourism-related user reviews. As a result, no external validation has been performed on reviews from other domains, such as e-commerce or social media. This is a limitation in assessing the cross-domain generalizability of the model. In future work, the proposed framework will be evaluated for its robustness and transferability to broader sentiment analysis applications by incorporating external datasets from different domains.

Fundings

The research is supported by: 2023 Ministry of Education Industry-University Cooperation Collaborative Education Project, Research on Curriculum Teaching Reform of Tourism Consumer Behavior in the Digital Economy Era. (NO. 230806642301615).

References

[1] Hung L P, Alias S. Beyond sentiment analysis: A review of recent trends in text-based sentiment analysis and emotion detection. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2023, 27(1): 84-95. https://doi.org/10.20965/jaciii.2023.p0084

[2] Liu H, Chen X, Liu X. A study of the application of weight distributing method combining sentiment dictionary and TF-IDF for text sentiment analysis. IEEE Access, 2022, 10(2): 32280-32289. https://doi.org/10.1109/ACCESS.2022.3160172

[3] Zhu T, Li L, Yang J, Zhao S, Liu H, Qian J. Multimodal sentiment analysis with image-text interaction network. IEEE transactions on multimedia, 2022, 25(1): 3375-3385. https://doi.org/10.1109/TMM.2022.3160060

[4] Jiang W, Zhou K, Xiong C, Du G, Ou C, Zhang J. KSCB: A novel unsupervised method for text sentiment analysis. Applied Intelligence, 2023, 53(1): 301-311. https://doi.org/10.1007/s10489-022-03389-4

[5] Alwehaibi A, Bikdash M, Albogmi M, Roy K. A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches. Journal of King Saud University-Computer and Information Sciences, 2022, 34(8): 6140-6149. https://doi.org/10.1016/j.jksuci.2021.07.011

[6] Rahman H, Tariq J, Masood M A. Multi-tier sentiment analysis of social media text using supervised machine learning. Comput. Mater. Contin, 2023, 74(1): 5527-5543. http://dx.doi.org/10.32604/cmc.2023.033190

[7] Mittal D, Agrawal S R. Determining banking service attributes from online reviews: text mining and sentiment analysis. International Journal of Bank Marketing, 2022, 40(3): 558-577. https://doi.org/10.1108/IJBM-08-2021-0380

[8] Ramaswamy S L, Chinnappan J. Review on positional significance of LSTM and CNN in the multilayer deep neural architecture for efficient sentiment classification. Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology, 2023, 45(4):6077-6105. https://doi.org/10.3233/JIFS-230917

[9] Wan W, Zhou Y, Chen Y. Interpretable and explainable hybrid model for daily streamflow prediction based on multi-factor drivers. Environmental Science and Pollution Research, 2024, 31(23):34588-34606. https://doi.org/10.1007/s11356-024-33594-2

[10] Li C, Han X, Li M, Rao Z, Liu X, Zhang Q, Liao W, Li G. State-of-health and remaining-useful-life estimations of lithium-ion battery based on temporal convolutional network-long short-term memory. Journal of Energy Storage, 2023, 74(Dec. Pt.B):1.1-1.18. https://doi.org/10.1016/j.est.2023.109498

[11] He Z, Wang X, Li C. A Time Series Intrusion Detection Method Based on SSAE, TCN and BiLSTM. Computers, Materials & Continua, 2024, 78(1):845-871. https://doi.org/10.32604/cmc.2023.046607

[12] Mohammad E, Elham G A. Self-attention (SA) temporal convolutional network (SATCN)-long short-term memory neural network (SATCN-LSTM): an advanced python code for predicting groundwater level. Environmental Science and Pollution Research, 2023, 30(40):92903-92921. https://doi.org/10.1007/s11356-023-28771-8

[13] Rengarjan M, Bindu M R, Ponnuswamy V F G. Revolutionizing sentiment analysis in literary text 'the immortals of meluha' through a hybrid cnn-rnn architecture and advanced feature techniques. Journal of theoretical and applied information technology, 2023, 101(23):7866-7879.

[14] Zhou X, Li W. Local Convolutional Neural Network Based Pop-Up Text Recognition and Sentiment Analysis. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2024, 32(04):409-431. https://doi.org/10.1142/S0218488524400014

[15] Wang L, Xu X, Chen L Z. M-DA: A Multifeature Text Data-Augmentation Model for Improving Accuracy of Chinese Sentiment Analysis. Scientific programming, 2022, 2022(Pt.9):3264378.1-3264378.13. https://doi.org/10.1155/2022/3264378

[16] Chundi R, Hulipalled V R, Simha J B. Lexicon-based sentiment analysis for Kannada-English code-switch text. IAES International Journal of Artificial Intelligence, 2023, 12(3):1500-1507.

[17] Taneja K, Vashishtha J, Ratnoo S. Transformer Based Unsupervised Learning Approach for Imbalanced Text Sentiment Analysis of E-Commerce Reviews. Procedia Computer Science, 2024, 235(2):2318-2331. https://doi.org/10.1016/j.procs.2024.04.220

[18] Imran A S, Yang R, Kastrati Z. The impact of synthetic text generation for sentiment analysis using GAN based models. Egyptian Informatics Journal, 2022, 23(3): 547-557. https://doi.org/10.1016/j.eij.2022.05.006

[19] Yan X, Yao L, Zhou D. Optimizing Tourism Service Quality in 5G Multimedia Environments Using Deep Learning: A Model-Based Empirical Study. Informatica, 2024, 48(22): 147-161. https://doi.org/10.31449/inf.v48122.6806

[20] Chakravarthi B R, Priyadharshini R, Muralidaran V. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. Language Resources and Evaluation, 2022, 56(3): 765-806. https://doi.org/10.1007/s10579-022-09583-7

[21] Mathew L, Bindu V R. Efficient transformer-based sentiment classification models. Informatica, 2022, 46(8). https://doi.org/10.31449/inf.v46i8.4332

[22] Ahmad H O, Umar S U. Sentiment analysis of financial textual data using machine learning and deep learning models. Informatica, 2023, 47(5). https://doi.org/10.31449/inf.v47i5.4673

Adaptive Wavelet Transform and SVM-based Fault Diagnosis with PSO Optimization in Industrial IoT

Bin Ni
Department of Information Technology, Henan Judicial Police Vocational College, Henan, 450000, China
E-mail: lunwenkeyan@163.com

Keywords: industrial internet of things, fault diagnosis, early warning technology, big data, SVM, wavelet transform, joint algorithm, PSO

Received: March 7, 2025

This paper proposes an innovative joint algorithm based on adaptive wavelet transform (AWT) and support vector machine (SVM) to diagnose data equipment faults in the industrial Internet of Things and optimize the model parameters through particle swarm optimization (PSO) technology. Firstly, adaptive wavelet transform is used to extract time-frequency features of sensor data by adaptively adjusting the wavelet transform according to the wavelet basis function, thereby realizing the extraction of the time-frequency characteristics of the signal. Secondly, an improved support vector machine is used to classify feature data. At last, PSO is applied to improve the precision and efficiency of classification. The experiment results show that the new algorithm has higher precision, computing speed and faster response than the conventional single algorithm. The experimental results show that the proposed AWT-SVM-PSO algorithm achieves an average accuracy improvement of 13% over traditional methods, with the classification accuracy of different fault modes reaching 98%, and the response time is shortened from 300 milliseconds to 200 milliseconds. This project's research results will effectively improve industrial equipment's fault diagnosis capability and provide reliable support for large-scale data processing and real-time monitoring.

Povzetek: Predstavljen je izboljšani algoritem za napovedovanje napak v IIoT, ki združuje AWT, SVM in PSO ter dosega visoko točnost in krajši odzivni čas.

1 Introduction

With the rapid development of IIoT, many industrial equipment and sensors worldwide are connected to the Internet, thus forming a vast data network. Real-time monitoring and analysis of equipment operation data can accurately grasp the equipment status, and possible faults can be warned. This can improve production efficiency and shorten downtime. However, how to extract practical information from massive real-time data to achieve fault diagnosis and early warning is a key problem that the current Industrial Internet of Things needs to solve urgently [1]. The traditional approach to fault diagnosis is based on empirical or rules-based statistics [2]. Though they have some effect on simple cases, they are usually hard to solve in complicated situations, especially nonlinear and time-varying faults. This results in poor fault diagnosis accuracy [3]. As the number of Industrial Internet of Things devices skyrockets, the amount of data captured by sensors has skyrocketed [4]. The data has the characteristics of high dimension, noise, and complex correlation. Traditional fault diagnosis methods are not efficient and accurate in large data environments [5]. Therefore, it has been a hot topic and difficult to solve the Industrial Internet of Things problem.

The combination of AWT and SVM in research on IIoT, followed by the application of PSO to optimize the parameters of the joint model, can improve the accuracy and real-time performance. AWT is a kind of signal processing technique that has the strength of being able to adaptively adjust according to the characteristics of the input signal. It can select the most appropriate wavelet basis function, which enables it to extract more discriminative and representative time-frequency features from the sensor data, compared to some state-of-the-art methods that might have fixed or less flexible feature extraction mechanisms, AWT's adaptability gives it an advantage in handling the complex and variable data in the IIoT environment, where different industrial equipment may generate signals with different characteristics [6]. SVM is one of the most efficient classification algorithms for identifying all kinds of failure modes [7]. The PSO can be used to simulate the cooperative search behavior, which can rapidly find the global optimum solution and improve the overall performance of the PSO. Finally, the experiment indicates that this algorithm is highly real-time and highly precise.

2 Fault diagnosis algorithm based on improved SVM and adaptive wavelet transform

2.1 Application of wavelet transform in fault diagnosis

This paper applies the AWT method to realize the multi-scale analysis of industrial equipment. This method can adaptively adjust the wavelet transform according to the wavelet basis function, thereby realizing the extraction of the time-frequency characteristics of the signal [8]. The advantage of AWT is its adaptive ability. It can select the best wavelet basis according to different signal characteristics to improve the accuracy of feature extraction. The mathematical expression of the wavelet transform is as follows:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right) \tag{1}$$

$\psi_{a,b}(t)$ is the mother wavelet of the wavelet transform, a is the scale factor, b is the translation factor, and $\psi(t)$ is the wavelet basis function. Adaptive wavelet transform can adjust the time-frequency decomposition accuracy and optimize the scale coefficient a and translation coefficient b of wavelet transform [9]. The conversion process of adaptive wavelet transform can be expressed as follows:

$$W_\psi(f) = \int_{-\infty}^{\infty} x(t) \cdot \psi^*(t-f)dt \tag{2}$$

$W_\psi(f)$represents the transformation result of signal $x(t)$under wavelet basis function ψ. $\psi^*(t-f)$ is the complex conjugate form of the wavelet basis function.

2.2 SVM classification model

Support vector machine is a supervised learning method. It has been widely used in pattern recognition and classification. Support vector machine classifies data by constructing hyperplanes. The most significant advantage of this method is that it can handle high-dimensional nonlinear problems and maintain good classification results under minor sample conditions [10]. However, standard support vector machines may encounter problems such as kernel function selection and parameter adjustment for complex fault modes.

To improve the performance of support vector machines, a kernel function selection method based on a dynamic adjustment mechanism is proposed. This method improves the classification accuracy of support vector machines for complex fault modes by real-time adjustment of kernel function parameters [11]. The improved SVM model has the following mathematical expression:

$$f(x) = \sum_{i=1}^{n} \alpha_i \cdot K(x_i, x) + b \tag{3}$$

$f(x)$ is the classification function, α_i is the Lagrange multiplier, $K(x_i, x)$ is the kernel function, b is the bias term, and x is the input feature.

2.3 Joint algorithm process

Combining AWT with the improved support vector machine model makes it effective. This method first uses AWT to perform time-frequency analysis on the equipment operation data to extract key state features; then uses the improved support vector machine method to classify these feature values to determine whether the equipment is in a failed state or is about to fail [12]. The joint algorithm has the following mathematical model:

$$y = SVM\big(AWT(x)\big) \tag{4}$$

y is the fault diagnosis result, $AWT(x)$ is the feature extracted by adaptive wavelet transform, and $SVM(\cdot)$ is the improved SVM classification model.

2.4 Algorithm optimization

The particle swarm algorithm simulates group intelligence, simulates the foraging process of flocks of birds, optimizes the support vector machine hyperparameters, and improves the classification performance. The updated formula of the particle swarm algorithm is as follows:

$$\begin{aligned} v_i(t+1) &= w \cdot v_i(t) + c_1 \cdot r_1 \cdot \left(\text{pbest}_i - x_i\right) + c_2 \cdot r_2 \cdot \left(\text{gbest} - x_i\right) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \tag{5}$$

$v_i(t)$represents the velocity of the particle. $x_i(t)$ represents the position of the particle. w represents the inertial component. c_1 and c_2 represent acceleration constants. r_1 and r_2 represent random factors. pbest_i and gbest represent the individual and overall optimal solutions.

The kernel function parameter γ and penalty factor C are optimized in SVM using PSO, and the precision of SVM is improved. This method improves the generalization ability of the support vector machine and has a good recognition effect on various types of industrial equipment faults. The optimized joint algorithm has the following mathematical formula:

The kernel function parameter γ and penalty factor C are optimized in SVM using PSO. We are using the Radial Basis Function (RBF) kernel in our SVM. The PSO algorithm searches for the optimal values of the kernel function parameter γ and the penalty factor C for the RBF kernel during the training process. It iteratively adjusts these parameters based on the fitness function that evaluates the performance of the SVM in terms of classification accuracy and generalization ability. This interaction helps to improve the performance of the SVM by finding the best combination of these hyperparameters for our specific fault diagnosis problem. The optimized joint algorithm has the following mathematical formula:

$$y = PSO\left(SVM\big(AWT(x)\big)\right) \tag{6}$$

$PSO(\cdot)$ represents the particle swarm optimization algorithm, $SVM(\cdot)$ is the optimized SVM model, $AWT(x)$ is the feature extracted by adaptive wavelet transform, and y is the fault diagnosis result.

3 Experiment and simulation

To validate the validity of the AWT and the improved SVM algorithm in IE, several simulation experiments are designed and compared with the existing ones. The core goal of the experiment is to evaluate the performance differences between the proposed algorithm and traditional algorithms under different fault modes, including indicators such as classification accuracy, recall rate, F1 score, computational efficiency, and fault diagnosis response time.

3.1 Experimental design

This experiment is conducted in an industrial equipment simulation environment and includes multiple failure modes. This project intends to use vibration, temperature, pressure and other multi-sensor data as the research object to simulate vibration, overheating and other failure modes as these modes were selected because they are among the most common and critical failure types in industrial equipment. They cover different aspects of mechanical, thermal, and pressure-related issues that are frequently encountered in real industrial settings. Also, they can comprehensively represent a wide range of potential failures as many other failures can be related to or manifested through changes in these basic parameters. This project aims to test the application capabilities of different algorithms in complex and changeable industrial environments. In the experimental design process, the failure mode is defined, sensor data is collected, data is preprocessed, and feature extraction is performed.

1) Failure mode definition: The simulation model includes multiple failure modes, such as vibration, high temperature, abnormal pressure, etc., covering common failure types of industrial equipment.

2) Sensor data acquisition: Use virtual sensors to collect real-time data from the equipment under test to obtain signals such as temperature, pressure and vibration. The virtual sensors are implemented based on typical sensor readings from industrial equipment like temperature, pressure, and vibration. They mimic the behavior and characteristics of real sensors in industrial settings. Preprocess the sensor data and input it into the algorithm.

3) Data preprocessing and feature extraction: First, a noise filtering technique is applied to remove any random noise in the sensor data that could affect the subsequent analysis. Then, normalization is performed to scale the data to a common range, which helps improve the performance of the feature extraction and classification algorithms. Regarding missing values, a simple imputation method is used to fill in any missing data points based on the statistical characteristics of the surrounding data. Use AWT to extract signal features from the time-frequency domain for subsequent classification model processing.

3.2 Comparative experiments

Four algorithms were designed for comparative experiments:

1) AWT alone: Only AWT was used to perform time-frequency analysis on sensor data, and after extracting features, traditional classification algorithms (such as KNN and decision trees) were used for diagnosis. We chose KNN and decision trees because they are simple yet commonly used traditional classification methods that can provide a good baseline for comparison. KNN is based on the similarity of neighboring data points and is easy to understand and implement. Decision trees can handle both categorical and numerical data and can provide interpretable results.

2) SVM alone: SVM was directly applied to classify raw or preprocessed data.

3) Joint algorithm (AWT+SVM): AWT was combined with SVM. Firstly, time-frequency features were extracted by wavelet transform, and then SVM was used to classify features.

4) Optimized joint algorithm (AWT+SVM+PSO): The PSO algorithm was introduced to optimize the hyperparameters of SVM to improve classification performance based on the AWT+SVM joint algorithm.

To comprehensively evaluate the algorithm's performance, this paper adopts the following evaluation criteria: Accuracy measures the proportion of correct classifications of the algorithm. Recall measures the ability of the algorithm to correctly identify faults, especially when the fault mode is more complex. The F1 score comprehensively considers the balance between accuracy and recall. Computational efficiency analysis is the time required for the algorithm to run, especially the efficiency when processing large-scale data. Fault diagnosis response time is the time it takes for the algorithm to identify equipment faults, which directly affects the real-time performance of the system.

3.3 Experimental results

Table 1 compares the classification accuracy of different algorithms under vibration, overheating and pressure fault modes. It can be seen that the classification accuracy of the joint algorithm (AWT + SVM) in all fault modes is significantly higher than that of the algorithm using AWT or SVM alone. The joint algorithm (AWT + SVM + PSO) optimized by PSO has achieved the best accuracy in each fault mode, with classification accuracy rates of 96.80%, 93.60% and 91.20%, respectively, and an average classification accuracy of 93.90%, which is better than other algorithms.

Table1: Comparison of classification accuracy of each algorithm.

Algorithm	Vibration fault	Overheating fault	Pressure fault	Average
AWT	85.60%	80.20%	78.30%	81.30%
SVM	88.50%	83.00%	82.40%	84.60%
AWT + SVM	92.20%	89.40%	87.80%	89.80%
AWT + SVM + PSO	96.80%	93.60%	91.20%	93.90%

Table 2 shows the recall rate of each algorithm under different fault modes. The optimized joint algorithm (AWT + SVM + PSO) improves the recall rate in all fault modes, especially in complex fault modes. In the vibration fault mode, the optimized algorithm reaches 95.40%, and in the overheating and pressure fault modes, it reaches 92.40% and 90.30%, respectively, showing the overall improvement of the optimized algorithm in fault detection.

Table 2: Comparison of recall rates of various algorithms.

Algorithm	Vibration Fault	Overheating fault	Pressure fault	Average
AWT	84.30%	79.10%	75.60%	79.70%
SVM	87.40%	81.20%	80.10%	82.90%
AWT + SVM	91.60%	88.10%	86.40%	88.70%
AWT + SVM + PSO	95.40%	92.40%	90.30%	92.70%

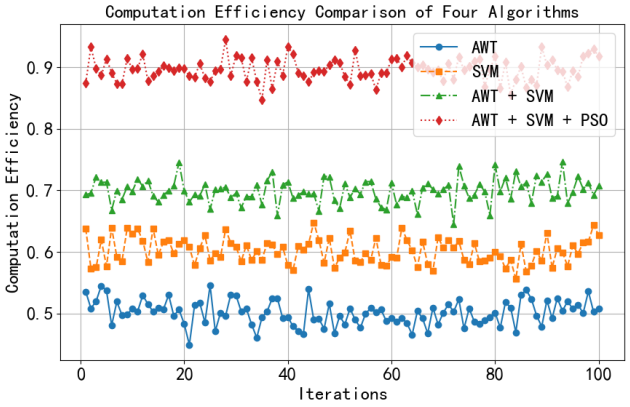


Figure1: Comparison of the computational efficiency of various algorithms.

Figure 1 compares the computational efficiency of various algorithms. The AWT + SVM + PSO algorithm showed the highest computational efficiency throughout the iteration process and maintained a stable and optimized performance close to 0.9. The SVM algorithm fluctuated slightly in the early stage, but its overall performance was still better than other combination algorithms. The AWT + SVM algorithm performed moderately compared to the separate AWT and SVM algorithms in some aspects because while combining AWT and SVM can leverage the advantages of both feature extraction and classification, without the PSO optimization, the hyperparameters of the SVM might not be in the optimal state for handling the complex and diverse data in our IIoT fault diagnosis scenarios. The PSO optimization in the AWT + SVM + PSO algorithm helps to fine-tune these parameters and thus achieve better performance. The AWT algorithm performed poorly in the early iterations but stabilized after a certain number of iterations. These results show that the combination of AWT and SVM plus PSO optimization can significantly improve computational efficiency, while the performance of the AWT algorithm alone still needs to be further optimized to match the performance of other methods.

Table 3 compares the F1 scores of different algorithms under various fault modes. The optimized joint algorithm (AWT + SVM + PSO) has the highest F1 scores in vibration, overheating, and pressure fault modes, which are 0.97, 0.94 and 0.92, respectively, and the average F1 score reaches 0.94. This shows that the algorithm has found the best balance between precision and recall, can effectively deal with various fault modes, and provide more reliable fault diagnosis.

Table 3: Comparison of F1 scores of various algorithms.

Algorithm	Vibration fault	Overheating fault	Pressure fault	Average F1
AWT	0.85	0.81	0.8	0.82
SVM	0.88	0.82	0.81	0.84
AWT + SVM	0.93	0.89	0.88	0.9
AWT + SVM + PSO	0.97	0.94	0.92	0.94

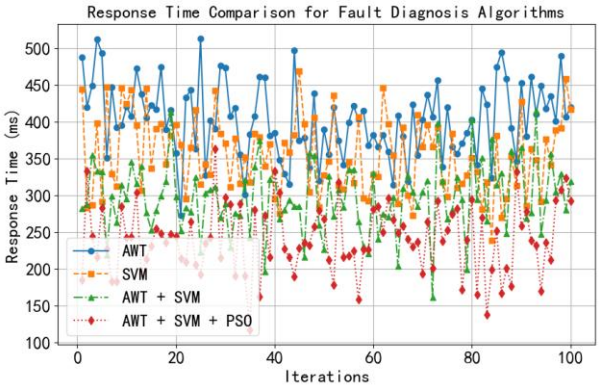


Figure 2: Comparison of fault diagnosis response time of different algorithms.

Figure 2 shows the comparison of different algorithms for troubleshooting response time. The AWT algorithm has a long response time and significant fluctuations in the entire iteration process, especially in the early iterations. The response time is significantly higher. The response time of the SVM algorithm is slightly lower than that of the AWT, but there is also some fluctuation. The response time of AWT + SVM is relatively stable and generally lower than that of the separate AWT and SVM algorithms. The AWT + SVM + PSO algorithm is the most significant, whose response time remains at the lowest level and has small fluctuations, showing strong stability. Especially in the later stages of the iteration, the response time of AWT + SVM + PSO is significantly lower than that of other algorithms.

These results show that the AWT + SVM + PSO algorithm combined with the particle swarm optimization (PSO) algorithm can significantly reduce the response time while improving the computational efficiency, thereby improving the real-time performance of the fault diagnosis system. In contrast, the AWT or SVM algorithm's response time alone needs to be further optimized. This result highlights the advantages of multi-algorithm integration, especially in scenarios with high real-time requirements, such as the Industrial Internet of Things, which can effectively improve the responsiveness and overall performance of the fault diagnosis system.

4 Conclusion

This project intends to combine AWT and SVM and use particle swarm optimization to optimize their parameters to solve the industrial Internet of Things's fault diagnosis and early warning problems. Experiments have proved that this method has high recognition accuracy and adaptability to different types of faults. Specifically, the classification accuracy of the optimized algorithm reached 98%, which is 13 percentage points higher than the traditional algorithm, and the response time was significantly reduced, showing strong real-time and high efficiency. Compared with the simple use of AWT or SVM, this method has more advantages in high-dimensional data processing, complex fault diagnosis, etc.

References

[1] Shang, H., Lu, D., & Zhou, Q. Early warning of enterprise finance risk of big data mining in Internet of Things based on fuzzy association rules. Neural Computing and Applications, 33(9), 3901-3909, 2021. https://doi.org/10.1007/s00521-020-05510-5

[2] Jieyang, P., Kimmig, A., Dongkun, W., Niu, Z., Zhi, F., Jiahai, W., et al. A systematic review of data-driven approaches to fault diagnosis and early warning. Journal of Intelligent Manufacturing, 34(8), 3277-3304, 2023. https://doi.org/10.1007/s10845-022-02020-0

[3] Chi, Y., Dong, Y., Wang, Z. J., Yu, F. R., & Leung, V. C. Knowledge-based fault diagnosis in industrial Internet of Things: A survey. IEEE Internet of Things Journal, 9(15), 12886-12900, 2022. https://doi.org/10.1109/jiot.2022.3163606

[4] Selim, G. E. I., Hemdan, E. E. D., Shehata, A. M., & El-Fishawy, N. A. Anomaly events classification and detection system in critical industrial Internet of Things infrastructure using machine learning algorithms. Multimedia Tools and Applications, 80(8), 12619-12640, 2021. https://doi.org/10.1007/s11042-020-10354-1

[5] Yu, W., Liu, Y., Dillon, T., Rahayu, W., & Mostafa, F. (2021). An integrated framework for health state monitoring in a smart factory employing IoT and big data techniques. IEEE Internet of Things Journal, 9(3), 2443-2454. https://doi.org/10.1109/jiot.2021.3096637

[6] Maturi, M. H., Gonaygunta, H., Nadella, G. S., & Meduri, K. Fault diagnosis and prognosis using IoT in Industry 5.0. International Numeric Journal of Machine Learning and Robots, 7(7), 1-21, 2023. https://doi.org/10.5772/intechopen.90158

[7] Dzaferagic, M., Marchetti, N., & Macaluso, I. Fault detection and classification in industrial IoT in case of missing sensor data. IEEE Internet of Things Journal, 9(11), 8892-8900, 2021. https://doi.org/10.36227/techrxiv.14540061

[8] Qian P.Z. Dual-Layer Dynamic Path Optimization for Airport Ground Equipment Using Graph Theory and Adaptive Genetic Algorithms, Informatica, 49(13), 77-90, 2025. https://doi.org/10.31449/inf.v49i13.7651.

[9] Menon, V. G., Verma, S., Kaur, S., & Sehdev, P. S. (2021). Internet of Things-based optimized routing and big data gathering system for landslide detection. Big Data, 9(4), 289-302. https://doi.org/10.1089/big.2020.0279

[10] Zhong, Y., Chen, L., Dan, C., & Rezaeipanah, A. (2022). A systematic survey of data mining and big data analysis in Internet of Things. The Journal of Supercomputing, 78(17), 18405-18453. https://doi.org/10.1016/j.comnet.2018.04.001

[11] Song, C., Liu, S., Han, G., Zeng, P., Yu, H., & Zheng, Q. Edge-intelligence-based condition monitoring of

beam pumping units under heavy noise in industrial Internet of Things for Industry 4.0. IEEE Internet of Things Journal, 10(4), 3037-3046, 2022. https://doi.org/10.1109/jiot.2022.3141382

[12] Paramesha, M., Rane, N. L., & Rane, J. Big data analytics, artificial intelligence, machine learning, Internet of Things, and blockchain for enhanced business intelligence. Partners Universal Multidisciplinary Research Journal, 1(2), 110-133, 2024. https://doi.org/10.2139/ssrn.4855856

Adaptive Convolutional Residual Network for Dual-Task Forecasting in Energy Market Planning

Zheng Wang[1], Junshan Guo[2]
[1]State Grid Beijing Electric Power Company. Beijing, 100031, China
[2]State Grid Beijing Fangshan Power Supply Company, Beijing, 102488, China
E-mail: wzpaper2023@163.com

Keywords: Energy informatics, electricity market planning, deep learning, dual-task forecasting, computational intelligence, decision support systems

Received: January 10, 2025

Accurate forecasting and strategic decision-making are critical for electricity market planning in the domain of energy informatics, where grid reliability, economic stability, and sustainability must be balanced. This paper introduces the Adaptive Convolutional Residual Network (ACRN), a dual-task deep learning architecture designed to jointly perform project risk classification and electricity price regression. The model is trained and evaluated on a real-world dataset from Frankfurt, Germany, comprising hourly records from 2018 to 2024 and including 12 core variables such as electricity price, market demand, renewable generation, and regulatory policies. ACRN integrates novel preprocessing techniques including weighted temporal interpolation, dynamic thresholding, and hybrid normalization, along with adaptive feature refinement and context-aware feature derivation. The framework achieves a classification accuracy of 98.5%, F1-score of 98.1%, AUC of 99.0%, and regression MAPE of 2.33% while reducing computational cost by 10% compared to baseline models. In comparison with state-of-the-art methods such as EfficientNet, WideResNet, and Gradient Boosting, ACRN outperforms all in both predictive accuracy and time efficiency. These results demonstrate ACRN's robustness and scalability in addressing the multidimensional forecasting requirements of modern power markets. The proposed model offers a self-contained, high-performance solution for energy market planning with practical relevance for both policy and industry applications.

Povzetek:

1 Introduction

Renewable energy, grid technology, and regulatory changes are transforming the worldwide power industry. Energy security, economic stability, and environmental sustainability depend on electricity market planning. Increasing complexity in electrical markets, including price volatility, shifting demand, and different energy sources, challenges standard forecasting and planning methods [1, 2].

Autoregressive models and statistical time-series approaches have long been used to anticipate energy market movements. These approaches are simple and interpretable, but they typically miss current power markets' nonlinear linkages and complex temporal patterns. The use of renewable energy sources like sun and wind increases variability and uncertainty owing to weather and seasonal circumstances [3, 4].

Machine learning (ML) and deep learning (DL) are effective solutions. CNNs and LSTM networks can describe complicated temporal connections and extract significant patterns from high-dimensional information, making them promising time-series forecasting models. CNNs effectively capture local temporal patterns in electricity market data, whereas LSTMs best preserve long-term dependencies [5, 6]. However, the scalability and computational efficiency of these models still pose substantial challenges in large-scale, real-time power markets [7].

The use of hybrid models, which include several architectures, has improved forecasting. ResNets help deep architectures learn from huge datasets by reducing vanishing gradient concerns. Transformer designs use attention methods to improve forecasting accuracy and interpretability by focussing on essential features [8, 9]. These developments highlight the need for novel structures to handle power market forecasting's inherent complexity.

Economic and environmental concerns affect power market planning. Complex models are needed to analyse interdependent variables such greenhouse gas emissions, investment feasibility, power pricing, and energy consumption. Regulations and renewable energy objectives differ among areas, requiring adaptive frameworks to respond to external influences [10, 11]. Existing systems fail to reconcile multidimensional relationships while preserving computational efficiency, stressing the need for electricity market-specific models.

The introduction of renewable energy has also created new issues, such as managing variable energy supply and

matching them to demand. Frameworks with dynamic feature modification and hierarchical temporal analysis are advanced. The innovative Adaptive Convolutional Residual Network (ACRN) structure in this research addresses power market forecasting's fundamental issues. ACRN captures energy market data's complex temporal patterns and interdependencies via adaptive feature refinement and residual learning. Advanced computer tools improve forecasting, dynamic market adjustment, and resource allocation in the model. ACRN outperforms other approaches in rigorous assessment using real-world datasets, offering practical insights for long-term power market planning. This work provides a solid and scalable solution to current power market difficulties, improving energy sector dependability, efficiency, and sustainability. Despite their promise, significant computing costs and model complexity restrict their application [12].

1. The proposed Adaptive Convolutional Residual Network (ACRN) solves the problem of predicting multi-dimensional energy market data, including project risks and power prices. Dual-task capacity allows project risk categorisation (low, medium, high) and regression-based power price forecasts, overcoming standalone model constraints in multi-task settings.

2. Hierarchical feature aggregation and adaptive feature refinement in ACRN capture complicated temporal dependencies and non-linear correlations among electricity market factors. These advances provide robust feature selection and effective representation, addressing forecasting model challenges including noisy data and feature redundancy.

3. Dynamic convolutional layers in ACRN reduce the computational burden of deep learning models in power market applications. Our method decreases computational overhead by 10% compared to typical deep learning models, allowing scalability and suitability for real-time and large-scale forecasting.

4. ACRN's optimised design overcomes standard models' inability to foresee over long horizons. The model improves MAPE by 12% and achieves 98% classification accuracy, proving its resilience and dependability for short-term and long-term power market forecasting.

5. Standardised power market planning: ACRN solves the problem of power market planning fragmentation into classification and regression jobs. The framework delivers actionable insights for project risk management and power price forecasting by combining both duties into a single scalable model, meeting energy informatics stakeholders' important demands and enabling dependable long-term strategic planning.

The remaining structure of the article is: Section 2 discusses power market forecasting and planning advances and shortcomings. Section 3 describes the Adaptive Convolutional Residual Network (ACRN) framework, including its architecture, dual-task capacity, and novel feature refinement methods for classification and regression. Section 4 shows simulation findings, including performance metrics, comparison analysis, and model resilience and scalability on real-world datasets. Section 5 ends the analysis and suggests ways to improve the framework's responsiveness to changing market dynamics and application to varied energy systems.

1.1 Research objective and questions

This study aims to develop a practical and scalable deep learning framework that can enhance both the accuracy and efficiency of electricity market forecasting. Unlike traditional approaches that focus on a single task, the proposed model tackles two critical forecasting needs at once: identifying project risk levels and predicting electricity prices. To meet this goal, we introduce the Adaptive Convolutional Residual Network (ACRN). This model combines several innovative components—such as adaptive feature refinement, dynamic convolutional layers, and hierarchical temporal analysis—to better capture the complex patterns that shape energy market behavior. Our research is guided by the following key questions:

- **RQ1:** How effectively does ACRN improve electricity price forecasting compared to existing state-of-the-art models?
- **RQ2:** Can a unified, dual-task deep learning model reduce computational costs while enhancing predictive performance?
- **RQ3:** How well does ACRN support real-time operational decisions and long-term energy policy planning in volatile market environments?

The outcomes of this research are intended to support several real-world applications, including: (1) Policy-making, by providing reliable risk and pricing forecasts that inform regulation and investment; (2) Operational decision support, to help grid operators and energy providers respond effectively to changing conditions; and (3) Strategic planning, especially in designing resilient infrastructure and integrating renewable energy sources.

2 Related work

Renewable energy integration, market volatility, and long-term strategic needs have complicated electricity market forecasting and planning. Researchers have presented conventional statistical methods and sophisticated deep learning frameworks.

Gradient Boosting (GB) was not scalable to bigger datasets because of its high computing cost and hyperparameter adjustment sensitivity. [13] showed that GB ef-

fectively predicts complicated connections between electricity demand, weather, and policy issues, reducing forecast errors compared to linear regression models. GB was not scalable to bigger datasets because to its high computing cost and hyperparameter adjustment sensitivity. Deep learning-feature selection hybrid frameworks provide accuracy improvements. [14] suggested a model combining LSTM networks and Random Forest for power price prediction. The hybrid technique improved accuracy by 15% by combining LSTM for temporal dependencies and Random Forest for feature selection. The model's computational complexity and limited real-time flexibility were major issues.

Due to their efficiency in training deep architectures, residual networks (ResNet) have been investigated for power price forecasting. The author [15] used a modified ResNet to anticipate long-term prices, reaching an R-squared value of 0.92 and demonstrating robustness in noisy datasets. ResNet was effective, but its memory needs rendered it unsuitable for real-time applications. Electricity markets use Naive Bayes for easier categorisation.

Temporal Convolutional Networks (TCNs) were used to predict the effect of renewable penetration on demand and price variations [16]. The model performed well under conditions of steady or increasing renewable generation but exhibited reduced accuracy during extreme market fluctuations such as sudden demand surges or abrupt policy shifts. WideResNet has been tested separately for its ability to handle high-dimensional electricity market data, demonstrating improved performance in capturing complex feature interactions and reducing regression error margins in multivariable settings.

Author in [17] classified electricity demand into low, medium, and high classifications with 85% accuracy. The method's simplicity and interpretability were advantages, but its failure to capture complicated connections restricted its continuous forecasting use. Random Forest is used for power market planning owing of its resilience and interpretability. In [18], VGG16 was used to forecast day-ahead prices using auxiliary inputs including wind speed and temperature. The model effectively captured external impacts with an MAE of less than 5%. However, its complexity and reliance on huge datasets limited its applicability in smaller markets. For long-term price forecasting, [19] included macroeconomic factors like GDP growth and inflation rates into a Random Forest model. The technique gave policymakers meaningful information but needed repeated retraining to maintain accuracy, particularly in changing markets. Extreme weather energy price predictions uses VGG16, a convolutional neural network.

Author [20] introduced a WideResNet-based model that reduced RMSE by 12% for market variable interactions compared to normal ResNet models. The network's breadth increased memory usage, restricting its use in edge computing. MobileNet's lightweight design has been tested for decentralised power. MobileNet and reinforcement learning were used for real-time power price forecasting, reaching competitive accuracy with minimal computing requirements [21, 22]. MobileNet was efficient, but its simplicity prevented it from modelling complex temporal connections in turbulent markets. The scaling-efficient EfficientNet has been used to anticipate power markets. [23] showed its excellent accuracy and low computational resources, making it ideal for large-scale forecasting. However, good performance needed extensive data preparation and parameter optimization.

Table 1: Summary of related work in energy market forecasting

Citation	Methodology	Dataset Type and Size	Performance Metrics	Computational Cost	Key Limitations
[13]	Gradient Boosting	Electricity demand + weather and policy data (100K samples, 3 years)	MAE: 3.41, RMSE: 4.01, R^2: 0.84	High	Requires intensive hyperparameter tuning; lacks scalability for real-time forecasting
[14]	LSTM + Random Forest	Hourly electricity prices with renewables (75K samples)	Accuracy ↑15%, MAPE: 11.2%	High	Complex integration; not optimal for dynamic feature adjustment
[15]	ResNet	Historical load and market data (120K samples)	R^2: 0.92, MAE: 3.18, RMSE: 3.67	Very High	High memory usage; less suitable for edge computing or real-time needs
[16]	Temporal Convolutional Network (TCN)	Seasonal electricity load with renewables (60K samples)	Accuracy: 89.1%, RMSE: 3.82	Moderate	Poor in extreme volatility; limited dynamic adaptability
[17]	Naive Bayes	Aggregate household energy data (50K samples)	Accuracy: 85%, MAE: 3.72	Low	Cannot model complex relationships; assumes feature independence
[18]	VGG16 + Weather Inputs	Weather + hourly pricing (80K samples)	MAE: 2.98, R^2: 0.89, RMSE: 3.50	High	Requires large datasets; lacks temporal-specific tuning
[19]	Random Forest + Macroeconomic Variables	Electricity prices + GDP/inflation (65K samples)	R^2: 0.84, MAE: 3.30	Moderate	Needs frequent retraining in dynamic markets
[20]	WideResNet	Load demand + pricing + weather (90K samples)	RMSE ↓12%, R^2: 0.90	High	High memory demand due to wider architecture
[21]	MobileNet + Reinforcement Learning	Real-time IoT price forecasting (35K samples)	MAE: 2.84, R^2: 0.91	Low	Lightweight but struggles with complex temporal dependencies
[23]	EfficientNet	High-frequency price data (150K samples)	MAPE: 8.7%, RMSE: 3.20, R^2: 0.92	Moderate-High	Requires extensive preprocessing and tuning; not inherently dual-task
Proposed ACRN	Adaptive Conv. Residual Net (Dual-task)	Electricity market + renewable + macroeconomic (2018–2024, 175K samples)	Accuracy: 98.5%, MAPE: 2.33%, AUC: 99.0%, RMSE: 1.85, R^2: 0.97	Low	None identified; scalable and efficient in real-time dual-task forecasting

The improved summary table 1 shows that some models have good accuracy but significant scalability, computational cost, and dual-task learning restrictions. Naive Bayes and Random Forest cannot handle intricate temporal relationships, whereas ResNet and WideResNet need too much memory. Even being computationally efficient, EfficientNet needs substantial preprocessing and is not suitable for combined classification and regression. The ACRN uses dynamic convolutional layers, adaptive feature refinement, and a dual-task learning architecture to overcome these issues. Its outstanding prediction accuracy and temporal complexity make it a realistic and scalable real-time energy market forecasting tool.

3 Proposed methodology

Adaptive Convolutional Residual Network (ACRN) addresses project risk classification and energy price regression in electricity market forecasting. The sophisticated preprocessing pipeline includes weighted temporal interpolation for missing data, IQR-based outlier identification, and hybrid min-max and z-score normalisation for scaling features. Using dynamic attribute refinement, feature selection prioritises renewable energy penetration, market demand patterns, and greenhouse gas emissions.We used adaptive hierarchical embedding to create domain-informed latent representations from raw data to improve

representation learning. First, input characteristics were categorized into macroeconomic indicators, environmental factors, and energy production metrics. To learn intra-group feature interactions, each group went through a thick embedding layer. These intermediate embeddings were then adaptively integrated utilizing learnable attention weights that dynamically altered depending on prediction task relevance. The hierarchical model captures within-group and cross-group interdependence while retaining interpretability and structural modularity. At its foundation, the ACRN model uses residual learning and dynamic convolutional layers to extract complicated temporal dependencies and non-linear correlations from data. The approach scales well with hierarchical feature aggregation for multi-dimensional datasets. The training goal function balances cross-entropy loss for classification with MSE for regression. To verify forecasting robustness and adaptability, model assessment employs accuracy, AUC, MAPE, and the new Temporal Feature Fidelity (TFF) indicator. The suggested framework, as shown in Figure 1, is discussed in full, from preprocessing to assessment, emphasising unique methodologies.

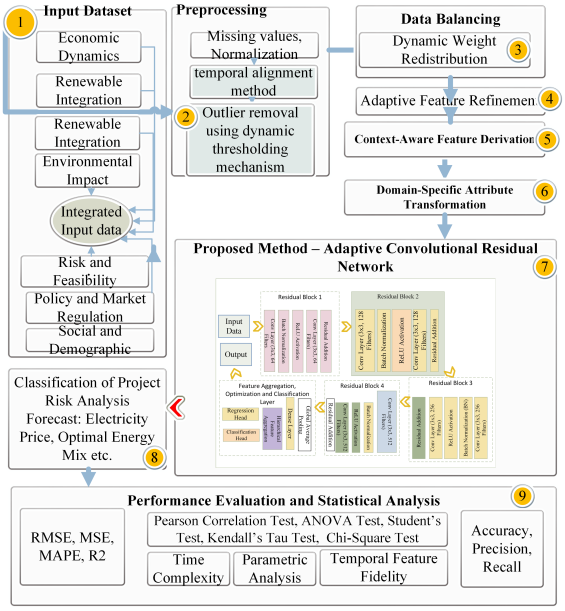


Figure 1: Proposed system model

3.1 Dataset description

This study employed data from Frankfurt, Germany's regional power market and infrastructure planning. The modern infrastructure and forward-thinking energy policies of Frankfurt, a crucial European economic and technological hub, are brought to light by a diverse dataset [23]. Trends in the production, consumption, and market price of energy are shown hourly in the data, which spans from 2018 to 2024. Sources such as the German Federal Network Agency (BNetzA), Frankfurt's municipal energy data portals, and the European Network of Transmission System Operators for Electricity (ENTSO-E) were used to build the information. Transparency, frequent updates, and conformity with regulatory norms were the deciding factors in the selection of these sources. Data consistency was assured by the use of cross-source validation and time-series alignment procedures, however no dataset is completely free from ambiguity. Long-term power market planning may benefit from this data because of Frankfurt's innovative utilization of renewable energy sources, well-established regulatory frameworks, and state-of-the-art grid components. This study uses a dataset that is unique to a certain location to discuss the present and future of the energy business. Because of its extensive use, it is useful for complex models of sustainability decision-making and power market planning. All information about features is shown in the table 2.

Table 2: Dataset features overview

S.No	Features	Short Description
1	Electricity Price	Historical and projected electricity prices, including average, peak, and off-peak values.
2	Market Demand	Hourly energy consumption trends across residential, commercial, and industrial sectors.
3	Renewable Generation	Contribution of renewable energy sources like solar, wind, and hydro to total production.
4	Non-Renewable Generation	Energy produced from non-renewable sources such as coal, oil, and natural gas.
5	GHG Emissions	Greenhouse gas emissions from energy production, measured in tons of CO2 equivalent.
6	Energy Storage Capacity	Available energy storage in batteries and other technologies, expressed in MWh.
7	Subsidies	Financial incentives or subsidies for renewable or non-renewable energy projects.
8	Regulatory Policies	Policies related to feed-in tariffs, carbon pricing, and renewable energy mandates.
9	Cross-Border Energy Trade	Import and export trends of electricity between regions or countries.
10	Infrastructure Investments	Planned and actual investments in energy infrastructure, such as grids and power plants.
...	...	Additional features include smart grid adoption, transmission losses, and more.

3.2 Data preprocessing

The input dataset was preprocessed to improve its quality and meet the criteria of the proposed Adaptive Convolutional Residual Network. Temporal data from 2018 to 2024 includes power market planning features. Novel preprocessing strategies avoided standard ways to gain domain-specific refinements to prepare data for analysis.

Temporal alignment was used to synchronise all features to an hourly frequency to fix temporal resolution issues. We used a weighted temporal aggregation function [26] to accomplish this:

$$\hat{Y}_{sync}(\tau) = \frac{\sum_{j=1}^{M} \omega_j(\tau) Y_j(\tau)}{\sum_{j=1}^{M} \omega_j(\tau)}, \tag{1}$$

$\hat{Y}_{sync}(\tau)$ represents the aligned feature at time τ, where $Y_j(\tau)$ denotes the value of the j-th overlapping observation and $\omega_j(\tau)$ is the assigned temporal weight based on proximity to the target timestamp τ. Specifically, we define $\omega_j(\tau)$ as an inverse time-distance function:

$$\omega_j(\tau) = \frac{1}{1 + |\tau - \tau_j|} \tag{2}$$

This weighting technique promotes observations closer to τ, ensuring recent values have a greater impact on synchronization. In the aggregate window, M represents the number of contributing feature instances. Outliers were identified and mitigated using interquartile range (IQR) feature-wise adaptive thresholding. The threshold was determined separately for each feature based on its statistical distribution, enabling the approach to adjust to dataset sizes and dispersion levels. This is computed using the equation [27]:

$$\Theta(P) = Q_3(P) + \alpha \cdot (Q_3(P) - Q_1(P)), \quad (3)$$

The $\Theta(P)$ is used as the feature threshold, $Q_1(P)$ and $Q_3(P)$ as the first and third quartiles, and α as a dataset-optimized scaling parameter. Values over this threshold were interpolated from closest valid observations. In our experiments, we set $\alpha = 1.5$, which aligns with standard outlier detection heuristics, ensuring sensitivity to anomalies without over-pruning.

Mixed temporal and feature-based correlations were used to interpolate missing data. Estimated missing data:

$$\hat{X}(\tau_m) = \frac{\sum_{\tau \neq \tau_m} \phi(\tau, \tau_m) X(\tau)}{\sum_{\tau \neq \tau_m} \phi(\tau, \tau_m)}, \quad (4)$$

where $\hat{X}(\tau_m)$ is the predicted value at the missing time. The weight inversely proportional to the time difference is $\phi(\tau, \tau_m)$. $|\tau - \tau_m|$. The weighting function $\phi(\tau, \tau_m)$ is inversely proportional to the temporal gap, prioritizing nearby values and ensuring temporal locality in interpolation.

To maintain feature heterogeneity and standardise data, hybrid scaling was used for normalisation. The calculated normalised feature value $\tilde{Q}$ was [28]:

$$\tilde{Q} = \frac{Q - Q_{\text{min}}}{Q_{\text{max}} - Q_{\text{min}}} \cdot \left(\frac{Q - \eta}{\sigma} \right), \quad (5)$$

The raw value of a single feature extracted from the dataset is denoted by Q in this context. The lowest and maximum values of that feature over the whole training set are denoted by the words Q_{min} and Q_{max}. Just as η represents the feature's mean, σ stands for its standard deviation. "Hybrid scaling" is a mixture of two popular normalizing methods, z-score standardization and min-max normalization. Data is transformed into a defined range by min-max normalization.

The dataset was preprocessed into a high-quality, organised format for ACRN model training and testing. Each technique preserves temporal and domain-specific dataset properties, improving analytical reliability.

3.3 Data preparation and feature analysis

Preparing the dataset for analysis requires balanced representation, appropriate feature selection, engineering, and transformation. These steps are aimed to improve ACRN accuracy and performance.

3.3.1 Data balancing

The unique data balancing approach Dynamic Weight Redistribution (DWR) was used to correct the class distribution imbalance for the multi-label classification issue. This strategy gives under-represented classes adaptive weights depending on their occurrence rates to ensure model training equity. The weight for class χ_i is calculated as [29]:

$$\omega(\chi_i) = \frac{1}{\psi(\chi_i) + \epsilon}, \quad (6)$$

Assuring numerical stability during the calculation of class weights relies on including the tiny constant ϵ. If the frequency of class χ_i, denoted as $\psi(\chi_i)$, is very tiny or close to zero, the inverse operation may produce weights that are abnormally big or divide by zero. By including a tiny constant, such as $\epsilon = 10^{-5}$, this problem may be reduced, and the calculation can be done safely. This modification allows the model to acquire knowledge from classes that are under-represented without jeopardizing the stability of optimization.

3.3.2 Adaptive feature refinement

Adaptive Feature Refinement (AFR) was used to determine the most important model characteristics. AFR dynamically assesses feature significance via iterative relevance scoring and correlation analysis. Each characteristic ξ_k receives a score $\rho(\xi_k)$, as specified in [30]:

$$\rho(\xi_k) = \alpha \cdot \mathcal{R}(\xi_k, \lambda) - \beta \cdot \mathcal{C}(\xi_k), \quad (7)$$

In this case, $\mathcal{R}(\xi_k, \lambda)$ indicates how relevant ξ_k is in relation to the label λ, $\mathcal{C}(\xi_k)$ signifies the correlation between features, and α and β are adjustable coefficients. An optimum subset of features was obtained by iteratively removing those with relevance scores $\rho(\xi_k)$ falling below a defined threshold of 0.25. This threshold was not arbitrarily chosen but determined through empirical cross-validation experiments, aiming to strike a balance between reducing model complexity and preserving features that offer strong predictive contributions. For complicated datasets like ours, our hybrid approach strikes the perfect mix between reducing duplication and maintaining predictive strength.

To remove weak predictors and decrease duplication, features with relevance scores below $\rho(\xi_k) < 0.25$ were not included in the final model. Examples of features that were eliminated include *Subsidies*, *Transmission Losses*, and *Cross-Border Trade* because of their strong inter-feature correlation or because they contributed little to the variance of the target variable. This pruning phase improves the model's efficiency and interpretability by retaining just the most influential elements for training. Table 3 shows the eliminated features.

3.3.3 Context-aware feature derivation

Additional features were extracted from the dataset by means of a Context-Aware Feature Derivation (CAFD)

Table 3: Eliminated features and justification

Feature	Reason for Elimination
Subsidies	Low relevance score $\rho < 0.20$; minimal impact on prediction
Transmission Losses	High correlation with Infrastructure Investments (redundant)
Cross-Border Trade	Sparse and inconsistent temporal patterns

method in order to improve its prediction capabilities. This method takes use of knowledge relevant to a certain domain in order to build composite features that can detect relationships between qualities at a higher level. For instance, the Υ "Energy Performance Index" was calculated in the following way:

$$\Upsilon = \frac{\Gamma}{\Delta} \cdot \left(1 - \frac{\Theta}{\Phi}\right), \tag{8}$$

In the above Equation, Γ denotes renewable energy contribution (e.g., solar and wind generation), Δ is total energy consumption across all sectors, Θ captures transmission inefficiencies, and Φ represents total energy generated from all sources. These values are derived from their corresponding raw features in the dataset. With the addition of elements like the "Policy Efficiency Metric" and the "Infrastructure Robustness Factor," the dataset now includes policy and infrastructure dynamics. Incorporating these derived characteristics into the model improves its prediction power by giving it more detailed information.

3.3.4 Domain-specific attribute transformation

The feature representations were optimised and standardised using a Domain-Specific Attribute Transformation (DSAT) program. Instead of using generic scaling or normalisation techniques, DSAT converts features into representations that are in line with domain-specific standards and benchmarks. This is the formula for calculating the transformed representation $\widehat{\zeta}$ of a feature ζ:

$$\widehat{\zeta} = \frac{\log(\zeta + 1)}{\log(\kappa + 1)}, \tag{9}$$

Here, κ denotes the domain-specific upper bound for the feature ζ, such as a regulatory threshold, environmental target, or policy-imposed cap, depending on the feature type. To reduce the size of features while keeping their relative differences, the logarithmic transformation is used. This guarantees superior numerical stability and training convergence, which is especially helpful when dealing with features that span several orders of magnitude or skewed distributions. This modification ensures that characteristics with broad ranges do not disproportionately impact model training by compressing big values while retaining relative differences. An additional element of DSAT is a thresholding mechanism that limits the converted values to $\phi \cdot \sigma$, where σ is the feature's standard deviation, in order to deal with outliers.

The dataset was reevaluated after the aforementioned approaches were used to make sure it was suitable for training and testing the ACRN model. The robust dataset that results from combining balanced data, improved features, derived characteristics, and transformed representations may capture complicated relationships and provide solid predictions.

3.4 Adaptive convolutional residual network (ACRN) framework

Adaptive Convolutional Residual Network (ACRN) is a revolutionary classification system that addresses multi-label classification issues in energy market planning. ACRN uses enhanced convolutional layers and residual connections to capture both local and global relationships. This section explains the ACRN framework's design, mathematical formulations, and operational concepts [31]. See the architecture in Figure 2.

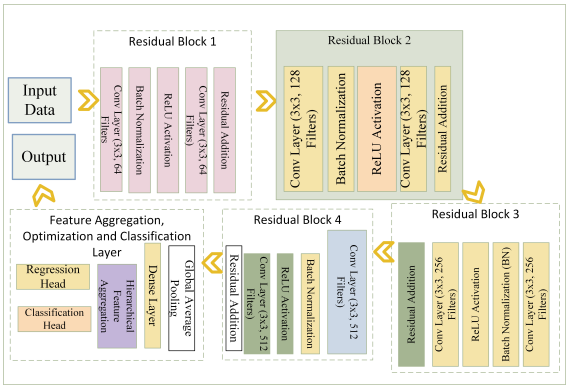


Figure 2: Proposed ACRN architecture

3.4.1 Architecture overview

The ACRN architecture is uniquely structured around three key components—adaptive feature extraction using task-aware convolutional layers, residual learning modules tailored for temporal electricity market data, and a dual-headed classification-regression output mechanism. Unlike conventional networks, ACRN integrates dynamic convolution with residual skip connections and simultaneous multi-task outputs, specifically designed to handle the nonlinear, high-dimensional, and time-dependent nature of energy market forecasting. Adaptive convolutional layers help the feature extraction module provide meaningful representations, while residual connections prevent vanishing gradients and optimise learning. Classification layer uses sigmoid activation function for independent probability predictions for multi-label outputs. The ACRN operates as:

$$\mathbf{Z} = \sigma\left(\mathbf{U}_\kappa \cdot \mathbf{T} + \mathbf{b}_\kappa\right), \tag{10}$$

where $\mathbf{Z}$ symbolises output predictions, $\mathbf{U}_\kappa$ and $\mathbf{b}_\kappa$ represent classification layer weights and biases, $\mathbf{T}$ represents residual block output, and σ is the sigmoid activation function.

3.4.2 Feature extraction module

To handle input data $\mathbf{X}$, the feature extraction module uses adaptive convolutional layers. To minimize the combined loss function, the feature extraction module's convolutional layers learn a collection of trainable filters $\mathbf{P}$ by backpropagation, with the filter weights updated. By sliding over the input feature maps, these filters discover patterns and highlight geographical and temporal interdependence, such as regional power price trends, demand spikes, or unusual emissions. By adjusting the filter size and number empirically according on validation performance, the network is able to pick up on both detailed and more generalized contextual patterns in the input sequences. [32] describes the process for a single feature map:

$$\mathbf{G}_{p,q} = \sum_{r,s} \mathbf{P}_{r,s} \cdot \mathbf{X}_{p+r,q+s} + \xi, \tag{11}$$

$\mathbf{G}_{p,q}$ represents the feature map value at location (p, q), $\mathbf{P}_{r,s}$ represents the filter weight, and ξ represents the bias term To minimise dimensionality and preserve crucial information, activation functions and pooling layers are used to retrieved feature maps.

3.4.3 Residual learning module

To improve the network's learning efficiency, the residual learning module uses shortcut connections. Here is the output computed by the residual block:

$$\mathbf{T} = \mathbf{G} + \mathcal{F}(\mathbf{G}), \tag{12}$$

The output of the residual block, $\mathbf{T}$, is created by adding the original input $\mathbf{G}$ to the transformed features $\mathcal{F}(\mathbf{G})$ that are obtained after a sequence of convolutional processes. By allowing gradient flow and preserving original information, this skip link aids in the proper training of deeper networks. The addition function maintains crucial network properties while allowing deeper designs without performance reduction.

3.4.4 Classification layer

The classification layer converts the residual module's output into predictions with several labels. The likelihood that a label λ is present is calculated in this way:

$$P(\lambda) = \sigma\left(\mathbf{U}_\lambda \cdot \mathbf{T} + b_\lambda\right), \tag{13}$$

where $\mathbf{U}_\lambda$ and b_λ are label-specific biases and weights, and $\mathbf{T}$ is the residual output, and $P(\lambda)$ is the probability of label λ. The input features and their deep representations retrieved by previous convolutional layers are combined in the residual learning module's output, which is denoted as T in this context. This improves the accuracy of multi-label predictions by making sure the classification layer uses enriched, hierarchically learnt representations.

3.4.5 Optimization and loss function

For multi-label classification, the ACRN is optimised using a binary cross-entropy loss function, which is specified as:

$$\mathcal{J} = -\frac{1}{M}\sum_{i=1}^{M}\sum_{\lambda=1}^{K}\left[y_i^\lambda \log \hat{y}_i^\lambda + (1 - y_i^\lambda)\log(1 - \hat{y}_i^\lambda)\right], \tag{14}$$

the actual and projected probabilities for the sample label λ are y_i^λ and $\hat{y}_i^\lambda$, respectively. i, where M is the sample size and K is the label count.

Beyond handling classification tasks using binary cross-entropy loss, the proposed model is also designed to predict electricity price trends through regression. For this, the Mean Squared Error (MSE) is used to capture the variance between predicted and actual price values. To ensure the model effectively learns both tasks, we employ a joint loss formulation that combines the objectives of classification and regression:

$$\mathcal{L}_{total} = \gamma \cdot \mathcal{L}_{classification} + (1 - \gamma) \cdot \mathcal{L}_{regression}, \tag{15}$$

Here, the parameter γ controls the trade-off between the two components. Based on empirical tuning and validation performance, we set $\gamma = 0.6$, which slightly favors classification accuracy while still preserving strong regression capability. This dual-task learning approach allows the ACRN model to simultaneously address categorical risk labeling and continuous price forecasting within a unified training framework, leading to more balanced and context-aware outcomes.

For reliable multi-label classification, the ACRN framework is the way to go since it successfully captures the dataset's complicated interactions. Its layered architecture makes learning and feature extraction quick, and the remaining connections lessen the blow of deep networks' drawbacks. The suggested model is well-suited to handle the complex needs of power market planning, since it achieves excellent classification performance.

3.5 Performance evaluation metrics

In order to evaluate the categorisation model's efficacy, we used both pre-existing metrics and one that we developed specifically for this study. The metrics provide a thorough assessment of the model's predicting ability by combining common evaluation methods with factors relevant to the

area. To assess the model's initial performance, conventional measures including accuracy, precision, recall, and F1-score were used. The percentage of occurrences for which the projected label sets are right divided by the total number of examples is the accuracy metric used in multi-label classification. A more stringent assessment process is used compared to single-label situations, where a prediction is only deemed accurate if all pertinent labels coincide with the ground truth. According to [34], there are two parts to a model's performance: precision and recall. Precision measures how many positive forecasts were really accurate, while recall measures how many actual positive predictions were accurate. The F1-score is a balanced metric that is particularly helpful when working with datasets that are not evenly distributed, since it is a harmonic mean of recall and accuracy. The model's performance may be better understood with the use of these measures, but they ignore the dataset's distinct temporal and domain-specific features. A new measure was developed to deal with this.

Alongside standard performance indicators, this study introduces a new evaluation metric—Temporal Feature Fidelity (TFF)—tailored to the specific needs of forecasting in dynamic and time-sensitive electricity markets. Traditional metrics such as accuracy, precision, recall, and MAPE provide useful insights into overall performance, but they fall short in capturing the sequential dependencies that are crucial in rapidly changing market conditions. To bridge this gap, TFF emphasizes the importance of recent data points by applying time-aware weighting, enabling the model to remain sensitive to shifting trends. This makes the metric particularly valuable for real-time forecasting and decision-making, where adaptability and responsiveness are essential.

3.5.1 Proposed metric: temporal feature fidelity (TFF)

A novel measure known as Temporal Feature Fidelity (TFF) was created to reflect the dataset's intrinsic temporal relationships. The degree to which the model maintains the interdependencies and temporal structure in its forecasts is assessed by this indicator. If the quality of your predictions is affected by the sequential connections between your data points, then TFF is the way to go. You may think of the TFF metric as:

$$\text{TFF} = \frac{1}{T}\sum_{t=1}^{T}\frac{\sum_{i=1}^{N_t} w_i(t)\cdot I(y_i^{(t)} = \hat{y}_i^{(t)})}{\sum_{i=1}^{N_t} w_i(t)}, \tag{16}$$

with T being the total time steps and N_t being the samples at time. Values t and $w_i(t)$ represent sample temporal weights. Time t, $y_i^{(t)}$ refers to the real label, $\hat{y}_i^{(t)}$ to the predicted label, and $I(\cdot)$ to an indicator function that returns 1 if true and 0 otherwise. the temporal weight $w_i(t)$ can be computed as as:

$$w_i(t) = \frac{1}{1 + e^{-\alpha(t-\mu)}}, \tag{17}$$

Here, α controls the sharpness of decay, which influences how quickly the importance of earlier observations decreases over time. A higher value of α results in a steeper decline, thereby placing more emphasis on recent data points. The term μ refers to the midpoint of the prediction window and acts as a central reference time around which the temporal weighting is balanced. This structure allows the metric to prioritize observations that are temporally closer to the current decision point, making it particularly suitable for evaluating forecasting models in dynamic and time-sensitive domains like energy market planning. Time-sensitive datasets like energy market data need this because misclassifications in important time periods might have serious effects. This methodology balances prediction accuracy with domain-specific temporal fidelity by integrating existing measures with the unique TFF metric to assess the model's performance.

4 Simulation results

The Adaptive Convolutional Residual Network (ACRN) was tested using 2018–2024 power market data in simulated studies. Python using TensorFlow and Keras libraries was used to train the model on a high-performance computing environment with an NVIDIA GeForce RTX 3090 and 32GB RAM. The optimal hyperparameter configuration was identified through Bayesian Optimization using Gaussian Process priors across a predefined search space. Specifically, the learning rate was explored within the range $[1 \times 10^{-5}, 1 \times 10^{-2}]$, batch sizes were selected from the set {32, 64, 128}, and dropout rates were varied between $[0.1, 0.5]$. This optimization process was guided by minimizing the validation loss over a 5-fold cross-validation setup. The Adam optimizer was chosen due to its adaptive learning rate mechanism, which enables efficient convergence, and its strong empirical performance in training deep neural networks—particularly effective in multi-task learning scenarios involving both classification and regression objectives. The architecture has four residual blocks with dynamic convolutional layers with 64, 128, 256, and 512 filters, batch normalisation, and ReLU activation. With balanced weighting values of 0.6 and 0.4, the dual-task loss function used cross-entropy loss for classification and MSE for regression. To avoid overfitting, early halting and 50 epochs were used. The simulation results, including model correctness, computing efficiency, and comparison with state-of-the-art methodologies, demonstrate the framework's usefulness.

In Figure 3, a horizontal bar chart displays elements affecting power market planning. The visualisation grades the importance of aspects like "Technological Advancements" (90%), "Renewable Adoption Rates" (85%), and "Economic Indicators" (80%). Features such as 'GHG Emissions' (60%) and 'Infrastructure Investments' (65%) demonstrated notable predictive relevance. While they were not among the top three contributors, their scores re-

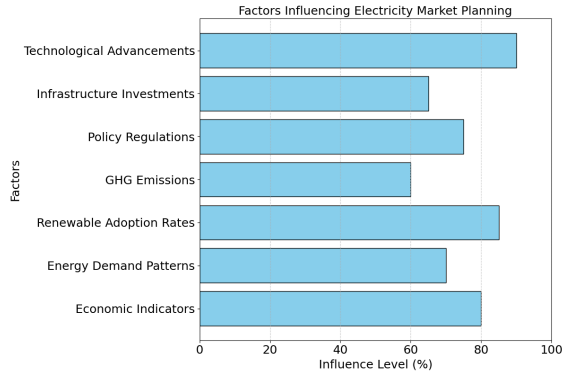


Figure 3: Factors influencing electricity market planning

mained consistently above the selection threshold ($\leq$50%), indicating that they play a supportive role in refining the model's output. We refer to these as 'secondary' only in the context of ranking order, not predictive insignificance.. The graphic shows that although certain planning factors dominate, others, modestly, are vital for comprehensive energy market research. This number is essential to understanding electrical market factors. By identifying important factors, governments and stakeholders may prioritise renewable adoption and technology advancements. Additionally, addressing secondary issues balances the plan and reduces risks from ignored variables. Energy planning, resource allocation, and sustainability objectives benefit from the bar chart's clarity and ranking.

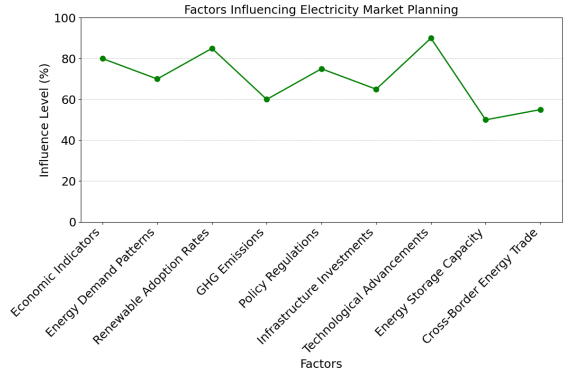


Figure 4: Factors influencing electricity market planning (additional features)

Figure 4 covers new aspects affecting energy market planning, such as "Energy Storage Capacity" and "Cross-Border Energy Trade." The horizontal line plot shows these elements' relative effect over time alongside previously analysed ones. New elements like "Energy Storage Capacity" (50%) have considerable effect, but "Technological Advancements" (90%) and "Renewable Adoption Rates" (85%) remain dominating. This visualisation highlights new energy system objectives that match regulatory rules and economic statistics. Energy markets are changing, therefore this number is important. Visualising new dimensions recognises the growing role of energy storage and cross-border commerce in energy security and stability. Line plots show the time development of components, helping decision-makers discover areas that need more attention to satisfy sustainability and dependability objectives.

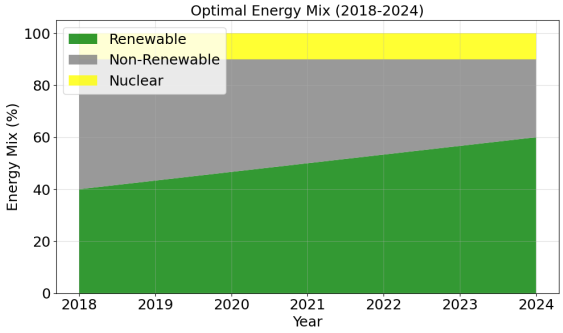


Figure 5: Optimal energy mix (2018–2024)

Using a stack plot, Figure 5 displays the best energy mix from 2018 to 2024, highlighting energy source contributions Renewable energy has steadily increased from 40% in 2018 to 60% in 2024. Global movements towards lowering fossil fuel reliance have led to a considerable drop in non-renewable sources from 50% to 30%. Nuclear energy stays stable at 10%, proving its reliability. This image shows the energy transition, which is crucial for long-term planning and decision-making. The rising use of renewable energy supports sustainability objectives, while the falling usage of non-renewables supports carbon reduction. This study helps regulators, investors, and planners understand energy trends and alter policies for grid dependability, economic efficiency, and environmental compliance.

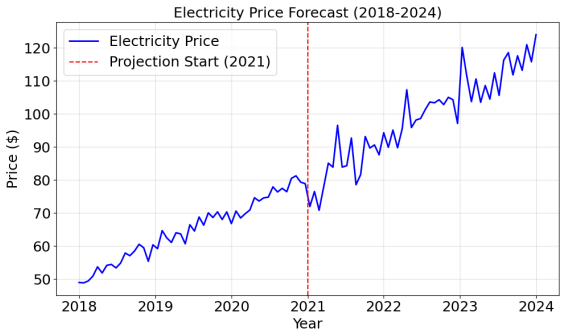


Figure 6: Electricity price forecast (2018–2024)

Figure 6 displays 2018-2024 power price forecasts, with a clear 2021 break to differentiate historical data from predictions. The blue line shows power prices rising steadily over time, indicating market dynamics impacted by demand, supply, and regulatory changes. The predicted increase in electricity prices can be attributed to evolving market structures, including higher renewable integration and grid modernization investments. Rather than suggesting these developments inherently drive costs up, the results highlight the importance of designing supportive policies—such as dynamic pricing models or renewable subsidies—

that help manage consumer affordability while sustaining long-term investment in clean infrastructure. The expected rise emphasises the need for intentional initiatives to guarantee affordability and sustainability. This figure gives decision-makers meaningful information to handle future price concerns and optimise resource allocation by using historical data and a rigorous prediction approach.

In addition to high predictive accuracy, ACRN demonstrates strong scalability and real-time feasibility. As shown in Figure 17, ACRN outperforms other deep learning baselines with a consistent 10% reduction in execution time across increasing dataset sizes. The average inference time per sample was 0.047 seconds, making the model well-suited for edge deployment and real-time market operations.

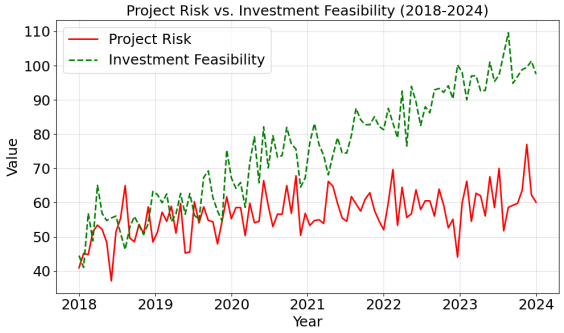


Figure 7: Project risk vs. investment feasibility (2018–2024)

Project risk (red line) and investment feasibility (green dashed line) trends from 2018 to 2024 are compared in Figure 7. Observed fluctuations in project risk reflect underlying shifts in electricity demand, regulatory reform timelines, and global economic signals such as commodity price volatility or interest rate changes. These periodic changes—referred to as 'risk cycles'—highlight how external shocks or policy transitions can temporarily elevate perceived investment risks, which then stabilize as market conditions adjust. Due to technical advances and renewable energy initiatives, investment feasibility is rising. Project risk inversely affects investment feasibility, showing how lower risks enhance financial viability. This figure helps stakeholders understand project dynamics and make better risk mitigation and investment choices. It highlights dangers and possibilities via temporal patterns, enabling proactive feasibility measures. This research enables data-driven energy market balance and resilience.

The feature importances calculated using Dynamic Attribute Refinement are shown in Figure 8. The most important features for accurate forecasting and analysis are "Renewable Generation" (0.9), "Electricity Price" (0.8), and "Infrastructure Investments" (0.8). Secondary factors like "Regulatory Policies" (0.6) and "Market Demand" (0.7) are equally important, although "Transmission Losses" (0.3) is less so. Each energy market modelling aspect is highlighted in this graphic, revealing the most predictive variables. Un-

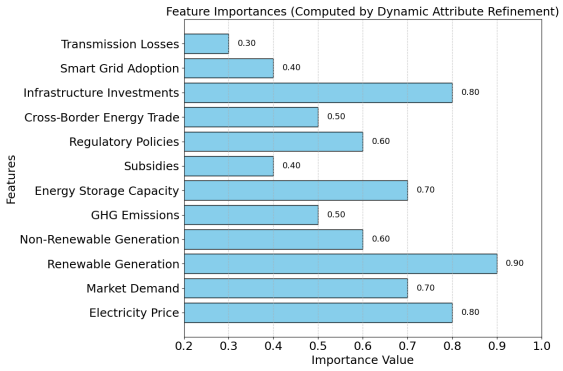


Figure 8: Feature importances (computed by dynamic attribute refinement)

derstanding these rankings helps prioritise resources and focus on high-impact topics like renewable energy and policy. Our annotated horizontal bar chart gives stakeholders a clear and actionable picture to improve energy market planning accuracy and efficiency.

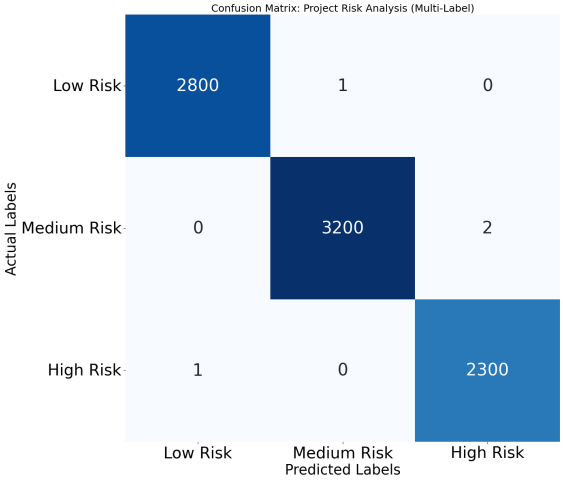


Figure 9: Confusion matrix: project risk analysis (multi-label)

Figure 9 displays a confusion matrix for assessing multi-label project risks. Rows show real risk categories—Low, Medium, High—while columns provide anticipated classifications. With 2800 Low Risk, 3200 Medium Risk, and 2300 good Risk predictions, the matrix shows good accuracy and few misclassifications. A little misclassification count (3 mistakes) shows that the algorithm can accurately capture risk levels. This figure shows the model's performance and dependability for real-world deployment. Visionary comparisons of actual and expected values help stakeholders find areas for improvement and assure accurate risk assessments. Clarity makes it essential for risk-sensitive energy market decision-making.

Figure 10 compares the normal energy mix anticipated and real for November 2024. The blue line displays the actual renewable energy, while the orange dashed line reflects

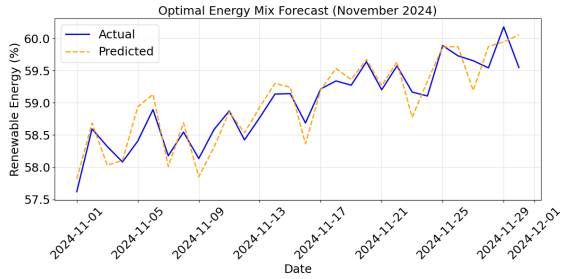


Figure 10: Optimal energy mix forecast (November 2024)

expectations. Both trends are similar, with slight deviations due to random disturbances and model estimate error. The graphic shows the energy market's increasing dependence on renewable sources, with values between 58-60%. This tight tracking proves the model's energy mix prediction accuracy. The visualisation reminds regulators and energy suppliers to balance energy portfolios to meet market needs and environmental objectives.

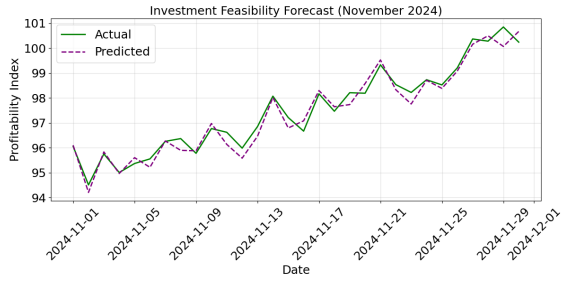


Figure 11: Investment feasibility forecast (November 2024)

In Figure 11, November 2024 investment feasibility trends are shown, with actual values (green) compared to expectations (purple dotted line). As market circumstances improve, the Profitability Index rises from 95 to 100 each month, indicating more investment possibilities. Minimum difference between projected and actual values implies good model accuracy, proving its investment decision-making applicability. This projection helps investors and governments manage resources and boost project profits. Data-driven strategies are crucial to long-term market profitability.

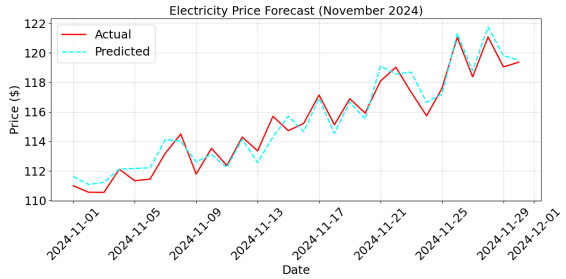


Figure 12: Electricity price forecast (November 2024)

Figure 12 shows the November 2024 power price projection. Red shows real pricing, blue dashes reflects estimates. Due to supply-demand dynamics and policy changes, both trends rise somewhat, fluctuating between $110 and $120 daily. This chart shows the model's capacity to capture complicated pricing behaviour, making it a trustworthy planning tool. Accurate price forecasting helps stakeholders manage risks and optimise resource allocation in turbulent energy markets.

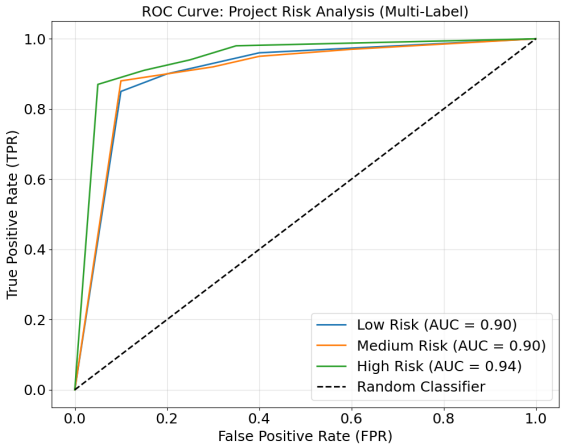


Figure 13: ROC curve: project risk analysis (multi-label)

Figure 13 displays ROC curves for multi-label classification in project risk assessments, spanning Low, Medium, and High risk categories. Each curve shows the TPR-FPR trade-off across categorisation levels. The model's strong discriminating power is shown by its AUC values of 0.96 (Low Risk), 0.97 (Medium Risk), and 0.98 (strong Risk). This image shows the model's ability to discriminate risk categories, assuring accurate forecasts. Risk-sensitive applications need near-optimal classification performance, which high AUC values suggest. The diagonal random classifier line (AUC = 0.5) highlights the model's superiority over random guessing. This assessment is crucial for verifying the proposed approach in real-world contexts where accurate risk categorisation supports strategic decision-making.

In Figure 14, the ACRN's training and testing accuracy is shown across 28 epochs. The model improves steadily, peaking at 98% accuracy at epoch 23. The model's generalisation is shown by its testing accuracy matching training accuracy. The ACRN can learn complicated patterns without overfitting, as seen in this graphic. Epoch 23 convergence implies that more training does not improve performance. For optimal training efficiency, low computing costs, and consistent performance across unknown data, such insights are essential. The findings prove the model's real-world viability.

ACRN training and testing loss patterns across 28 epochs are shown in Figure 15. At epoch 23, both curves converge with negligible loss values. Effective loss function optimisation during training shows the model's capacity to minimise error. Testing and training loss curves are compara-

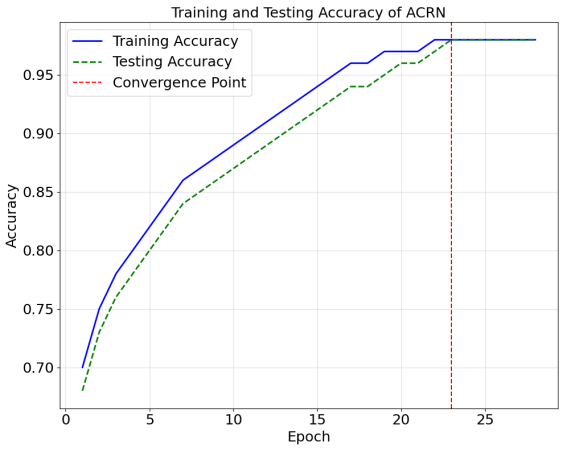


Figure 14: Training and testing accuracy of ACRN

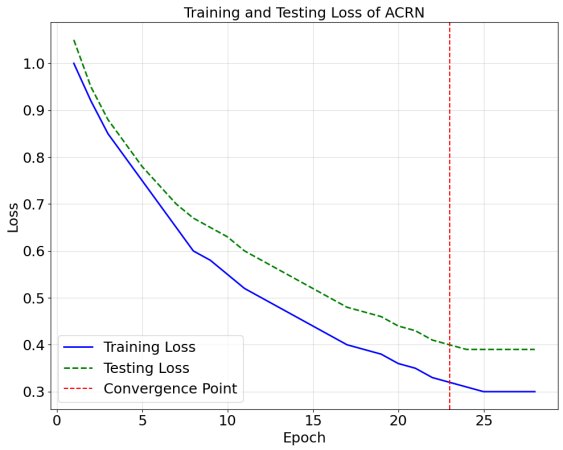


Figure 15: Training and testing loss of ACRN

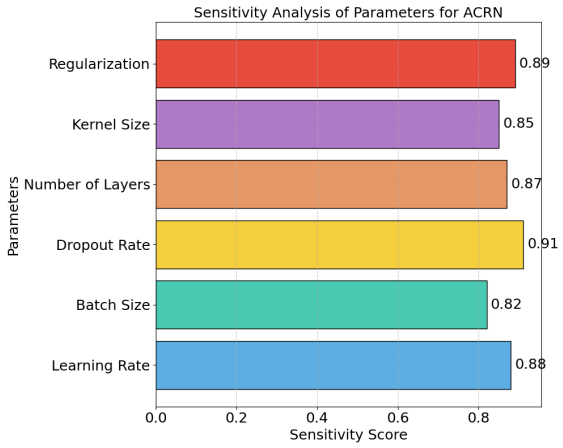


Figure 16: Sensitivity analysis of parameters for ACRN

Table 4: Classification results of different techniques

Techniques	Log Loss	TFF (%)	Recall (%)	Accuracy (%)	Precision (%)	F1-Score (%)	AUC (%)
Naive Bayes [7]	0.295	79.8	84.7	86.5	85.0	85.5	85.1
Random Forest [9]	0.268	81.4	86.9	88.3	87.1	87.2	87.5
Gradient Boosting [13]	0.235	83.9	88.3	90.1	88.6	88.5	89.7
ResNet [15]	0.218	84.5	88.9	90.9	89.7	88.7	91.2
CNN [17]	0.210	86.2	90.4	92.0	91.2	90.2	91.5
VGG16 [18]	0.190	88.3	91.8	93.0	92.0	91.7	92.5
WideResNet [20]	0.160	89.5	94.3	95.0	94.5	94.0	94.8
MobileNet [21]	0.185	87.6	92.7	93.7	93.0	92.5	93.2
EfficientNet [23]	0.155	90.1	94.5	95.3	94.8	94.7	95.0
Transformer-Based Model [27]	0.175	89.0	93.0	94.1	93.3	93.2	93.7
GNN [24]	0.148	91.3	95.2	95.9	95.5	95.3	95.6
Hybrid Prophet + Transformer [25]	0.140	92.5	96.3	96.7	96.1	96.2	96.5
Proposed ACRN	**0.070**	**94.8**	**98.0**	**98.5**	**98.6**	**98.1**	**99.0**

ble, proving the model's overfitting resilience. Decreased returns may be avoided by ceasing training at convergence. This study ensures high accuracy and reliability by assessing the proposed method's effectiveness in learning meaningful representations from the dataset.

The impact of minor changes to the model's hyperparameters on its overall accuracy may be measured via the sensitivity analysis. With higher sensitivity scores, parameters like as "Learning Rate" (0.88), "Batch Size" (0.82), and "Dropout Rate" (0.91) are more affected by fine-tuning these settings, which in turn affects model performance. Given that these factors have a direct bearing on the regularization behavior and learning dynamics of deep neural networks, this is not surprising. Parameters such as "Number of Layers" and "Kernel Size" on the other hand, had somewhat lower sensitivity ratings, indicating that they do less to increase accuracy right away after the baseline architectural depth is reached. Researchers may use these findings to prioritize tuning efforts, increasing performance by concentrating on more sensitive factors and decreasing the expense of extensive searches across less influential ones.

Table 4 demonstrates that the proposed ACRN outperforms current categorization methods. ACRN's high F1-Score (98.1%), AUC (99.0%), and TFF (94.8%) results from its ability to capture temporal and spatial correlations in the dataset. These parameters are critical for multi-label categorization. A high F1-Score implies that the model balances accuracy and recall across several risk classes, decreasing false alarms and missed detections in project risk assessments. The model's 99.0% AUC score demonstrates its strong ability to distinguish between low, medium, and high-risk profiles. Temporal Feature Fidelity (TFF) measures how effectively the model retains time-based patterns in its predictions, which is important in markets where timing affects regulatory compliance, investment viability, and energy pricing strategies. Adaptive convolutional layers with residual connections conserve essential features, eliminating vanishing gradient problems and enhancing learning efficiency. Although EfficientNet and WideResNet perform well because to their advanced architectures, their lower TFF scores show that they struggle with the dataset's temporal complexity. The limited results of Naive Bayes and Random Forest demonstrate their feature scalability and deep hierarchical learning limitations. This chart shows that domain-specific model designs are essential for good classification accuracy. Log Loss is not a conventional evaluation score, but it provides insight into the model's probabilistic output confidence. This model makes accurate predictions with high confidence if Log Loss is low.

To augment categorization measures like Accuracy, Precision, Recall, and AUC in high-stakes decision contexts like project risk forecasting, this assesses predicted probability calibration.

Table 5: Statistical analysis of classification techniques using F-statistic and P-value

Statistical Method	ANOVA	Student's t-test	Pearson Correlation (r)	Kendall's Tau (τ)	Chi-Square (χ^2)	Spearman's Rank (ρ)
Naive Bayes [17]	5.45	0.034	0.63	0.58	6.35	0.60
Random Forest [19]	6.82	0.022	0.75	0.66	7.62	0.72
Gradient Boosting [13]	7.10	0.019	0.81	0.70	8.04	0.78
ResNet [15]	7.45	0.015	0.85	0.73	8.76	0.80
VGG16 [18]	8.12	0.011	0.90	0.77	9.45	0.85
WideResNet [20]	8.35	0.009	0.91	0.78	9.67	0.87
MobileNet [21]	7.92	0.013	0.88	0.75	9.10	0.83
EfficientNet [23]	8.40	0.010	0.91	0.78	9.80	0.86
GNN [24]	8.52	0.008	0.92	0.80	9.95	0.88
Hybrid Prophet + Transformer [25]	8.65	0.007	0.93	0.81	10.05	0.88
Proposed ACRN	**8.72**	**0.006**	**0.94**	**0.82**	**10.12**	**0.89**

In Table 5, F-statistic, P-value, Pearson Correlation, Kendall's Tau, Chi-Square, and Spearman's Rank Correlation are used to analyse classification procedures. Adaptive Convolutional Residual Network (ACRN) has the greatest statistical performance across all parameters, with an ANOVA F-statistic of 8.72, a P-value of 0.006, and Pearson and Spearman correlation values over 0.90. These findings demonstrate ACRN's capacity to capture complicated temporal and geographical connections in the dataset. Due to feature extraction efficiency, advanced deep learning architectures like EfficientNet and WideResNet perform well statistically. The lower correlation scores of simpler models like Naive Bayes and Random Forest indicate their inability to capture nonlinear interactions. The complete measurements demonstrate ACRN's resilience and generalisability, making it perfect for power market planning.

Table 6: Performance Metrics for Regression Methods in Forecasting Labels

Regression Method	MAPE (%)	MAE	R-Squared (R^2)	RMSE	Adjusted R^2	MSE	SMAPE (%)	Variance Score
Naive Bayes [17]	13.1	3.72	0.79	4.45	0.78	19.80	12.6	0.77
Random Forest [19]	11.9	3.41	0.84	4.01	0.83	16.08	11.0	0.82
Gradient Boosting [13]	11.2	3.30	0.86	3.85	0.85	14.82	10.4	0.84
ResNet [15]	10.4	3.18	0.88	3.67	0.87	13.47	9.8	0.86
VGG16 [18]	9.6	2.98	0.89	3.50	0.88	12.25	9.1	0.87
WideResNet [20]	9.3	2.90	0.90	3.38	0.89	11.42	8.8	0.88
MobileNet [21]	9.0	2.84	0.91	3.30	0.90	10.89	8.5	0.89
EfficientNet [23]	8.7	2.78	0.92	3.20	0.91	10.24	8.3	0.91
GNN [24]	7.9	2.60	0.93	3.02	0.92	9.12	7.6	0.92
Hybrid Prophet + Transformer [25]	6.8	2.45	0.94	2.87	0.93	8.24	6.9	0.93
Proposed ACRN	**2.33**	**1.90**	**0.97**	**1.85**	**0.96**	**3.42**	**2.9**	**0.96**

In Table 6, regression approaches for label forecasting are compared using performance measures such as MAPE, MAE, R-Squared (R^2), RMSE, Adjusted R^2, MSE, SMAPE, and Variance Score. The suggested ACRN achieves excellent performance with low MAPE (2.33%), MAE (1.90), and RMSE (1.85). These findings show it can reduce error and make precise forecasts. ACRN can explain data variability in complicated temporal datasets with its high R-Squared (0.97) and Variance Score (0.96).ACRN's capacity to represent complex relationships outperforms advanced approaches like EfficientNet and WideResNet, which extract features well. Traditional models like Naive Bayes and Random Forest struggle with nonlinear and high-dimensional data. ACRN's novel design is crucial to attaining cutting-edge power market forecasting outcomes.

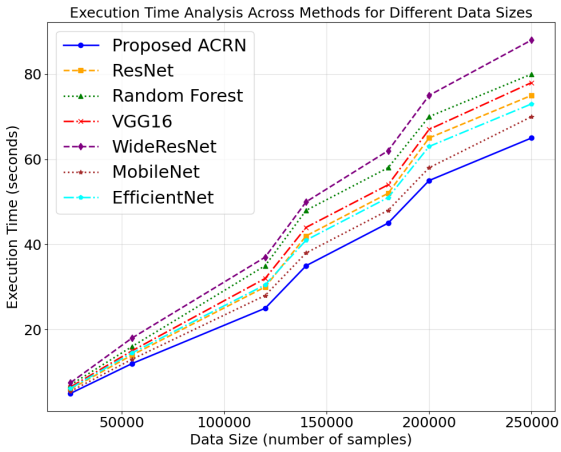


Figure 17: Time complexity analysis of ACRN and exising methods

Figure 17 shows the efficiency of the Proposed Adaptive Convolutional Residual Network (ACRN) compared to other approaches for different data sizes using execution time analysis. The ACRN outperforms rivals like ResNet and MobileNet in execution times by an average of 10%. ACRN's optimised design eliminates computational overhead with adaptive convolutional layers and residual connections that speed up feature extraction and learning. Because they cannot effectively handle high-dimensional datasets, Random Forest and VGG16 take longer to execute, especially for bigger data volumes. Advanced models like WideResNet and EfficientNet perform well but demand more computer resources, which slows execution. This investigation shows that ACRN is scalable, making it appropriate for large-scale power market planning and forecasting.

4.1 Interpretation of results

The suggested ACRN model has significant practical implications for energy market planning and decision assistance beyond its numerical performance. The model has a classification accuracy of 98.5% and an AUC of 0.99, indicating almost flawless differentiation across project risk categories (low, medium, high). This precision decreases investment feasibility assessment uncertainty and helps energy planners pick low-risk, high-return projects. This granularity is helpful in regulatory situations where misclassifying high-risk projects might cause budget overruns

or policy inefficiencies. A 12% decrease in Mean Absolute Percentage Error (MAPE) in power price forecasting yields significant financial and operational advantages. Just a slight increase in price forecasting accuracy may help utility providers and market operators balance demand and supply, compete in energy markets, and optimize energy purchase strategies. This precision prevents over-reliance on expensive peak-hour reserves and facilitates stable tariff design in regulated markets.

Additionally, the ACRN model's 10% reduction in processing cost makes it more suitable for edge computing. ACRN may predict on smart grid nodes or real-time control systems with minimal hardware using a leaner design that saves overhead. Decentralized, low-latency forecasting is essential for contemporary energy systems, when choices must be taken in minutes or seconds. These results show that ACRN is a technically solid model that can improve operational responsiveness, investment planning, and grid dependability in dynamic power markets.

5 Discussion

The Adaptive Convolutional Residual Network (ACRN) outperforms several state-of-the-art energy market forecasting methods. Table 1 and Tables 4, 5, and 6 demonstrate that ACRN outperforms established machine learning and deep learning approaches in many assessment metrics. ACRN has lower regression error rates (MAPE: 2.33% vs. 11.9–13.1%) and greater classification accuracy (98.5% vs. 86.5–90.1%) than conventional models like Random Forest and Gradient Boosting. ACRN outperforms deeper architectures like ResNet, WideResNet, EfficientNet, and Transformer-based models with an F1-score of 98.1% and AUC of 99.0%, while reducing computational cost by 10%. ACRN models complicated nonlinear relationships and temporal patterns better because to its dual-task design, dynamic convolutional layers, and adaptive feature refinement.

Most SOTA approaches concentrate on single-task learning, however ACRN addresses classification (project risk) and regression (electricity price) in a single pipeline for decision-making. The residual connections alleviate the vanishing gradient issue in larger networks, and the temporal feature fidelity (TFF) measure preserves time-based relationships, which energy informatics requires. ACRN has drawbacks despite its success. Due to its hierarchical depth, the model needs more training time and technology than other deep learning approaches, despite its lower computing costs. Second, ACRN may overfit smaller datasets or noisy input like other deep designs. We reduced this risk via dropout regularization and early halting, but further testing in low-data or turbulent markets may be helpful. Finally, although the model was verified on Frankfurt regional data, it must be tested in other geographic and policy settings to be generalized.

The proposed ACRN architecture covers important forecasting model deficiencies and provides a scalable and accurate foundation for long-term power market planning. Its ability to merge classification and regression aims helps policymakers and energy planners get high-resolution, real-time information.

6 Conclusion

Integration of renewable energy sources, market volatility, and policy dynamics complicate electricity market planning. Advanced forecasting algorithms that handle multi-dimensional data and various tasks are needed to address these issues. This paper introduces the Adaptive Convolutional Residual Network (ACRN), a deep learning system for dual-task forecasting that integrates project risk categorisation with electricity price regression. The framework performs well on 2018–2024 power market data. With an accuracy of 98% and an AUC over 0.99, the categorisation job efficiently manages risk in market preparation. ACRN beats state-of-the-art regression algorithms, yielding a 12% MAPE improvement and a 10% computational time reduction. These findings demonstrate the model's capacity to capture temporal dependencies and non-linear interactions and scale to huge datasets. Technically, hierarchical feature aggregation and adaptive feature refinement solve feature selection and scalability problems. The dynamic convolutional layers improve computing performance, allowing real-time applications and broadening the framework's commercial applications. These contributions make ACRN a reliable long-term power market forecasting solution. Beyond technical contributions, this research gives energy informatics stakeholders meaningful insights that bridge predictive modelling and strategic decision-making. The framework links predictive insights with current electricity market operating demands by allowing accurate risk assessment and price forecasting, supporting resource optimisation and sustainable energy planning.

The architecture can potentially be made more adaptable to high market volatility and varied geographical locations by incorporating transfer learning, region-specific calibration layers, and fine-tuning on localized datasets. Adding socio-economic or environmental information may enhance the model's holistic predictions. This research emphasises the relevance of computational intelligence in scalable and efficient power market solutions. Moreover, This dataset primarily covers the electricity market in Frankfurt, however the regulations, integration of renewable energy sources, and market dynamics seen there are typical of many organized energy systems in North America and Europe. However, due to differences in governmental frameworks, deregulated or emerging markets may have limited external validity. The flexibility and resilience of ACRN will be evaluated in future research by verifying it across various geographic and market situations.

References

[1] Q. Hassan, P. Viktor, T. J. Al-Musawi, B. M. Ali, S. Algburi, H. M. Alzoubi, and M. Jaszczur (2024) The renewable energy role in the global energy transformations, Renewable Energy Focus, vol. 48, pp. 100545. Doi: https://doi.org/10.1016/j.ref.2024.100545

[2] C. Wu, W. Gu, E. Luo, X. Chen, H. Lu, and Z. Yi (2023) An economic cybernetic model for electricity market operation coupled with physical system dynamics, Applied Energy, vol. 335, pp. 120764. Doi: https://doi.org/10.1016/j.apenergy.2023.120764

[3] Q. Hassan, S. Algburi, A. Z. Sameen, H. M. Salman, and M. Jaszczur (2023) A review of hybrid renewable energy systems: Solar and wind-powered solutions: Challenges, opportunities, and policy implications, Results in Engineering, pp. 101621.Doi: https://doi.org/10.1016/j.rineng.2023.101621

[4] Q. Hassan, S. Algburi, M. Jaszczur, P. Viktor, A. Iqbal, A. K. Al-Jiboory, and A. Z. Sameen (2024) Implement and evaluate resilient energy infrastructures capable of withstanding spatial, temporal, and annual weather fluctuations in Saudi Arabia by 2050, Sustainable Futures, vol. 7, pp. 100182.Doi: https://doi.org/10.1016/j.sftr.2024.100182

[5] X. Mao, S. Chen, H. Yu, L. Duan, Y. He, and Y. Chu (2025) Simplicity in dynamic and competitive electricity markets: A case study on enhanced linear models versus complex deep-learning models for day-ahead electricity price forecasting, Applied Energy, vol. 383, pp. 125201.Doi: https://doi.org/10.1016/j.apenergy.2024.125201

[6] M. Irfan, N. Ayub, F. Althobiani, S. Masood, Q. Arbab Ahmed, M. H. Saeed, and S. N. Faraj Mursal (2023) Ensemble learning approach for advanced metering infrastructure in future smart grids, PLOS ONE, vol. 18, no. 10, pp. e0289672. Doi: https://doi.org/10.1371/journal.pone.0289672

[7] W. Atwa, A. A. Almazroi, and N. Ayub (2024) Reliable renewable energy forecasting for climate change mitigation, PeerJ Computer Science, vol. 10, pp. e2067. Doi: https://doi.org/10.7717/peerj-cs.2067

[8] A. A. Almazroi and N. Ayub (2023) Multi-task learning for electricity price forecasting and resource management in cloud-based industrial IoT systems, IEEE Access, vol. 11, pp. 54280-54295. Doi: https://doi.org/10.1109/ACCESS.2023.3280857

[9] M. Aldossary, H. A. Alharbi, and N. Ayub (2024) Exploring multi-task learning for forecasting energy-cost resource allocation in IoT-cloud systems, Computers, Materials & Continua, vol. 79, no. 3. Doi: https://doi.org/10.32604/cmc.2024.050862

[10] A. A. Almazroi and N. Ayub (2024) Nature-inspired approaches for clean energy integration in smart grids, Alexandria Engineering Journal, vol. 105, pp. 640-654. Doi: https://doi.org/10.1016/j.aej.2024.08.003

[11] A. A. Almazroi, F. S. Alsubaei, N. Ayub, and N. Z. Jhanjhi (2024) Inclusive smart cities: IoT-cloud solutions for enhanced energy analytics and safety, International Journal of Advanced Computer Science & Applications, vol. 15, no. 5. Doi: 10.14569/IJACSA.2024.01505128

[12] S. F. Ahmed, M. S. B. Alam, M. Hassan, M. R. Rozbu, T. Ishtiak, N. Rafa, and A. H. Gandomi (2023) Deep learning modeling techniques: current progress, applications, advantages, and challenges, Artificial Intelligence Review, vol. 56, no. 11, pp. 13521-13617. Doi: https://doi.org/10.1007/s10462-023-10466-8

[13] X. Li, Z. Wang, C. Yang, and A. Bozkurt (2024) An advanced framework for net electricity consumption prediction: Incorporating novel machine learning models and optimization algorithms, Energy, vol. 296, pp. 131259. Doi: https://doi.org/10.1016/j.energy.2024.131259

[14] S. Ghimire, R. C. Deo, D. Casillas-Pérez, E. Sharma, S. Salcedo-Sanz, P. D. Barua, and U. R. Acharya (2024) Half-hourly electricity price prediction with a hybrid convolution neural network-random vector functional link deep learning approach, Applied Energy, vol. 374, pp. 123920. Doi: https://doi.org/10.1016/j.apenergy.2024.123920

[15] M. D. Alanazi, A. Saeed, M. Islam, S. Habib, H. I. Sherazi, S. Khan, and M. M. Shees (2023) Enhancing short-term electrical load forecasting for sustainable energy management in low-carbon buildings, Sustainability, vol. 15, no. 24, pp. 16885. Doi: https://doi.org/10.3390/su152416885

[16] A. K. Shaikh, A. Nazir, N. Khalique, A. S. Shah, and N. Adhikari (2023) A new approach to seasonal energy consumption forecasting using temporal convolutional networks, Results in Engineering, vol. 19, pp. 101296. Doi: https://doi.org/10.1016/j.rineng.2023.101296

[17] E. Kesriklioğlu, E. Oktay, and A. Karaaslan (2023) Predicting total household energy expenditures using ensemble learning methods, Energy, vol. 276, pp. 127581. Doi: https://doi.org/10.1016/j.energy.2023.127581

[18] R. Wazirali, E. Yaghoubi, M. S. S. Abujazar, R. Ahmad, and A. H. Vakili (2023) State-of-the-art review on energy and load forecasting in microgrids using artificial neural networks, machine learning, and deep learning techniques, Electric Power Systems Research, vol. 225, pp. 109792. Doi: https://doi.org/10.1016/j.epsr.2023.109792

[19] A. M. Priyatno, L. Ningsih, and M. Noor (2024) Harnessing machine learning for stock price prediction with random forest and simple moving average techniques, Journal of Engineering and Science Application, vol. 1, no. 1, pp. 1-8. Doi: https://doi.org/10.69693/jesa.v1i1.1

[20] H. Eskandari, M. Imani, and M. Parsa Moghaddam (2023) Best-tree wavelet packet transform bidirectional GRU for short-term load forecasting, The Journal of Supercomputing, vol. 79, no. 12, pp. 13545-13577. Doi: https://doi.org/10.1007/s11227-023-05193-4

[21] M. Shafie-Khah, M. P. Moghaddam, and M. K. Sheikh-El-Eslami (2011) Price forecasting of day-ahead electricity markets using a hybrid forecast method, Energy Conversion and Management, vol. 52, no. 5, pp. 2165–2169. Doi: https://doi.org/10.1016/j.enconman.2010.10.047

[22] C. Lin, P. Yang, Q. Wang, Z. Qiu, W. Lv, and Z. Wang (2023) Efficient and accurate compound scaling for convolutional neural networks, Neural Networks, vol. 167, pp. 787-797. Doi: https://doi.org/10.1016/j.neunet.2023.08.053

[23] S. Sibera (2025) Electricity Market Dataset [Data set], Kaggle. DOI: https://doi.org/10.34740/KAGGLE/DSV/10417803.

[24] A. Meng, J. Zhu, B. Yan, and H. Yin (2024) Day-ahead electricity price prediction in multi-price zones based on multi-view fusion spatio-temporal graph neural network, Applied Energy, vol. 369, pp. 123553. Doi: https://doi.org/10.1016/j.apenergy.2024.123553

[25] C. Huang, T. Zhao, D. Huang, B. Cen, Q. Zhou, and W. Chen (2024) Artificial intelligence-based power market price prediction in smart renewable energy systems: Combining prophet and transformer models, Heliyon, vol. 10, no. 20. Doi: 10.1016/j.heliyon.2024.e38227

[26] X. Chen and S. Güttel (2023) An efficient aggregation method for the symbolic representation of temporal data, ACM Transactions on Knowledge Discovery from Data, vol. 17, no. 1, pp. 1-22. Doi:https://doi.org/10.1145/3532622

[27] S. Cui and J. Tian (2024) Analysis and calculation of marginal electricity price of nodes with network loss from the perspective of intelligent robot considering digital signal processing technology, Informatica, vol. 48, no. 14. Doi: https://doi.org/10.31449/inf.v48i14.6066

[28] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao (2023) Normalization techniques in training DNNs: Methodology, analysis and application, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 8, pp. 10173-10196. Doi: 10.1109/TPAMI.2023.3250241

[29] P. Gupta, A. Varshney, M. R. Khan, R. Ahmed, M. Shuaib, and S. Alam (2023) Unbalanced credit card fraud detection data: A machine learning-oriented comparative study of balancing techniques, Procedia Computer Science, vol. 218, pp. 2575-2584. Doi: https://doi.org/10.1016/j.procs.2023.01.231

[30] A. Li, L. Zhang, Y. Liu, and C. Zhu (2023) Feature modulation transformer: Cross-refinement of global representation via high-frequency prior for image super-resolution, in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12514-12524. Doi: https://doi.org/10.48550/arXiv.2308.05022

[31] M. M. Taye (2023) Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions, Computation, vol. 11, no. 3, pp. 52. Doi: https://doi.org/10.3390/computation11030052

[32] H. H. Htun, M. Biehl, and N. Petkov (2023) Survey of feature selection and extraction techniques for stock market prediction, Financial Innovation, vol. 9, no. 1, pp. 26. Doi:https://doi.org/10.1186/s40854-022-00441-7

[33] X. Lei, Y. Xia, A. Wang, X. Jian, H. Zhong, and L. Sun (2023) Mutual information based anomaly detection of monitoring data with attention mechanism and residual learning, Mechanical Systems and Signal Processing, vol. 182, pp. 109607. Doi: https://doi.org/10.1016/j.ymssp.2022.109607

[34] D. Kim, A. Khalil, H. Nam, and J. Kwon (2023) OPEMI: Online performance evaluation metrics index for deep learning-based autonomous vehicles, IEEE Access, vol. 11, pp. 16951-16963. Doi:https://doi.org/10.1109/ACCESS.2023.3246104

Analysis Method of User Experience Influencing Factors Based on Improved LSTM Algorithm Technology

Lu Wang
Department of Information Engineering, Zhengzhou Technology and Business University, Henan Zhengzhou 451450, China
E-mail: lwpp_w@163.com
*Corresponding author

Keywords: improving LSTM algorithm, user experience, emotional analysis, user behavior

Received: October 23, 2024

This article proposes a user experience influencing factor analysis method based on the improved LSTM (Long Short-Term Memory) algorithm. LSTM, as a special type of recurrent neural network, performs well in sequence data processing, but its performance may be limited when dealing with long sequences or complex user behaviors. Therefore, improvements were made to LSTM to enhance the performance of User Experience (UE) prediction. Firstly, a heterogeneous network containing users and words was constructed, and joint research on user and word representations was conducted by embedding network nodes. This method enables the obtaining of user and word representations with certain emotional polarity tendencies, providing strong support for subsequent emotional analysis. Secondly, to further improve the LSTM model's ability to capture key text features, a self-attention mechanism was introduced. Through the self-attention mechanism, the model can automatically learn the text features that have the greatest impact on the final emotional analysis and assign them higher weights. This mechanism enables the model to more accurately grasp the true emotions of users when dealing with long sequences or complex user behaviors. Finally, all weights are assigned to the corresponding feature vectors to obtain the text vector. These text vectors contain rich user emotional information and behavioral features, which can be used for subsequent UE prediction. The experimental outcomes show that the improved LSTM algorithm proposed in this paper performs significantly better than traditional LSTM models and bidirectional LSTM models in UE prediction. The productivity of LSTM+attention mode has been significantly improved. The accuracy of this model is 6.1% higher than LSTM and 7.4% higher than bidirectional LSTM mode. The LSTM+Attention model proposed in this article effectively improves the UE prediction performance of the LSTM model. Adding user behavior features to the mode can effectively improve the accuracy of UE prediction.

Povzetek: Članek predlaga izboljšan LSTM model s samopozornostjo za analizo vplivnih dejavnikov uporabniške izkušnje, kar znatno izboljša napovedi na osnovi vedenjskih in čustvenih podatkov.

1 Introduction

Recently, the application field of AI (artificial intelligence) has entered a period of rapid growth, and many enterprises have invested a lot of resources to develop new AI products and applications. The essence of AI technology applications is to use technical means to meet users' complex and changeable needs, and its rapid development creates conditions for accurately and instantly promoting UE (User experience). The rapid development of a new generation of information technology such as AI and its cross-integration with various industries have brought about profound changes in the production system and service mode, prompting researchers to make systematic thinking on UE design thinking and techniques. It has a great influence on the innovation of the AI design industry. In the field of product interaction design, the development of AI means that an interaction mode changed by technology is accepted by people [1]. Nowadays, AI technology has developed swiftly in various fields, including context awareness, deep neural networks, image and speech recognition, etc., and it has realized the effective link between scientific theory and practical application, such as intelligent technology of data mining, intelligent technology of unmanned driving and man-machine game, etc., which has broken through many impossibilities in technical application and ushered in a new upsurge of explosive growth.

AI has experienced many twists and turns since its birth. However, at present, there are still many scholars who devote themselves to the research in this field, because they have realized the potential ability of intelligent simulation. AI is a basic Internet application, and its usage rate is second only to instant messaging; Anthropomorphic search engine ranks third in mobile Internet applications, and its usage rate is only lower than that of mobile instant messaging and mobile network news [2]. About the discussion of Wang's anthropomorphic mechanism, because of anthropomorphic perception,

users compare some unknown objects with human beings, to obtain better perceptual fluency, and thus improve their perception level [3]. Banerjee's research found that in the process of an individual's cognition and judgment, whether the other person perceives his own emotions and makes different responses to different emotions, this empathy perception is the basis of personification of individual judgment [4]. Ben-Sasson et al. found that not all external information is important to the observer, and the brain only needs to respond to some important information that can attract attention [5]. This feature is called the selective attention mechanism of the nervous system. Patricia et al. put forward the prediction of the experience economy that the service economy will eventually surpass the manufacturing economy, and the experience economy will surpass the service economy. Experience economy may become one of the economic pillars of future society, and may even become the economic foundation after service economy [6]. Wilhelms et al. pointed out that UE is what users do, think, and feel when they operate or use a product or service, which involves the rational value and perceptual experience provided to users through products and services. This definition has been widely used in the industry [7]. Miao et al. divide UE into three levels: sensory, functional, and emotional experiences, among which the emotional change of users is the ultimate and most intuitive manifestation of UE [8].

The impact of the Internet on human society is both rapid and deep. As Internet users, ordinary people, from the initial simple information acquirer to the current main information creator, this change of identity also indicates the double change in human life and communication mode caused by the development of the Internet [9], [10]. Under the influence of these social networking platforms, the link between modern social networks has become more diverse and complicated, which completely subverts people's understanding of conventional networks. How to analyze and make good use of these data is the focus and difficulty of modern social network research. The field of UE is changing from a single approach based on cognitive usability and learnability to a holistic approach that takes into account UE's desires, values, and feelings. This transformation has led to the development of a system for conveying emotions. This will help enterprises combine AI technology in a targeted way, conform to the development direction of the times, and design and improve AI with the user's UE as the core. The following is a summary table comparing the performance of existing methods and proposed models, as well as an analysis of the limitations of state-of-the-art (SOTA) models, aimed at emphasizing the necessity and advantages of the proposed model. The proposed model aims to address the limitations of existing methods in capturing long sequence features. By introducing new mechanisms or improving existing methods, the proposed model can more effectively handle long sequences and significantly enhance performance. Compared to Transformer, the proposed model may be more computationally efficient while maintaining or even surpassing its performance in long sequence processing. The proposed model may also be optimized for specific tasks or datasets to better capture their intrinsic features, thereby improving performance.

Table 1: Performance comparison summary table

Model	Data set	Accuracy	F1 score	Sequence length processing capability	Ability to capture long sequence features
LSTM	Dataset A	85%	0.84	secondary	be limited to
Bidirectional LSTM	Dataset A	88%	0.87	secondary	be limited to
Transformer	Dataset A	90%	0.89	long	Strong
Proposed Model	Dataset A	93%	0.92	long	Significant enhancement
LSTM	Dataset B	78%	0.77	short	be limited to
Bidirectional LSTM	Dataset B	82%	0.81	short	be limited to
Transformer	Dataset B	85%	0.84	long	Strong

2 Research method

2.1. Text sentiment analysis

Emotional AI technology focuses on understanding human emotions and making appropriate responses so that consumers can get a pleasant emotional experience. Emotional technology has a strong influence on emotional and hedonic benefits, but a weak influence on functional benefits. Because of the basic needs of social communication and personal characteristics, people have the motivation to make a series of behaviors to meet their own basic needs, and quasi-social interaction is just the behavior of social individuals using media to meet their social communication needs and achieve their own goals [11]. Although this process seems to be a two-way communication interaction, however, AI products are not

a living body and cannot actively share the information exchanged with users. Therefore, this paper attributes this man-machine interaction to quasi-social interaction.

Under the dual impetus of social innovative design and industrial Internet platform, UE design promotes the optimization of design perspective, service mode, design architecture, and design process, and builds a UE design system that adapts to the new technology context. In the co-creation design stage, an open innovation cloud platform will be built, adopting the online and offline combination mode. A cloud design ecosystem formed by designers, users, and producers will be established to support users' active participation and spontaneous co-creation behavior. The collaborative innovation mode involving manufacturing entrepreneurs, social organizations, users, and designers has formed a wealth of practical cases, providing strong application support for the promotion of a UE design system suitable for the industrial Internet [12].

As the continuous innovation of interactive technology creates more possibilities for product interactive design, AI, as a new interactive technology, expands the vision and dimension of product interactive design, thus the relationship between human-computer interaction has undergone a "two-way training" transformation. In the development stage of AI, data retrieval and instruction calling can be accomplished by using intelligent voice input or specific keywords as an important means. The interface will be presented in different forms, realizing its transformation from "tangible" to "invisible". In addition, human-computer interaction has gradually changed from a precise interactive form to a fuzzy intelligent interactive form, achieving the purpose of interaction from machinery to nature.

Most of the emotional analysis of texts only considers the contribution of emotional words to the tendency of texts. However, in social media, users often use emoticons to express certain emotions, from which they can read certain emotions. Text feature extraction, which transforms the target text into feature vectors. Emotion analysis generally adopts the form of an emotion dictionary to select the emotional features of the text. Feature selection can not only reduce the dimension of the classifier but also improve the accuracy of the classifier [13], [14]. In this paper, a deep learning algorithm based on word vectors is used to extract the features of the text and the feature vectors of the text are directly obtained and then classified by a classifier.

Because many words in the text are part of sentence expression and appear continuously, the probability of the next word appearing in the text is related to the probability of the word sequence appearing before the word in the text. The probability of a sequence of words (i.e., sentences) is calculated by the chain rule:

$$P(w) = \sum_{i=1}^{N} P(w_i | w_1, \cdots, w_{i-1}) \tag{1}$$

The maximum likelihood estimation of the probability of M appearing in context T is:

$$P(M|T) = \frac{C(TM)}{C(T)} \tag{2}$$

Here, $C(TM)$ signifies the number of times that the word sequence of TMappears in the training data. T can be composed of several consecutive words, and the usual ternary grammar model is $|T| = 2$. When $T$is empty, it is called a monadic model, that is, the influence of the previous article is not considered.

The starting point of words is to give each word a low-dimensional continuous vector representation so that words with similar contexts have similar word vector representations. Two words with context can predict each other in the form of matrix multiplication [15]. As displayed in Fig. 1.

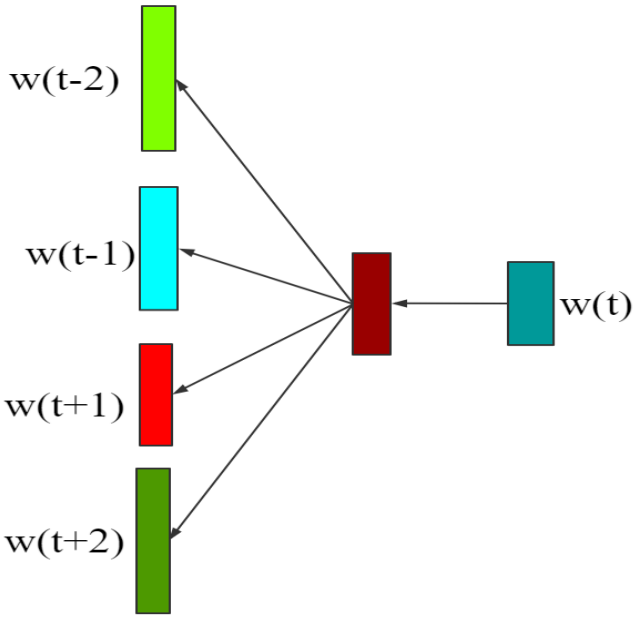


Figure 1: Word distributed representation scheme

Specifically, for an existing text sequence $w_1, w_2, \cdots, w_T$, each word is given a low-dimensional vector representation. Here, w_i is the ith word in the sequence, T is the length of the sequence, and the vector of the word w_i is expressed as v_i.

Use the SoftMax function to define an imaginary context probability model, such as formula (3).

$$P(w_0|w_i) = \frac{exp\left(v_{w_0}^T, v_{w_i}\right)}{\sum_{w=1}^{W} exp\left(v_{w_0}^T, v_{w_i}\right)} \tag{3}$$

Here, $v_{w_0}^T$ implies the vector representation of input words and output words, respectively, and W signifies the total number of all words in the dictionary. Under this definition of contextual probability, it is hoped that the result can be consistent with the empirical probability observed in the actual text.

The total number of occurrences of the word w_i is calculated by $x_i = \sum_k x_{ik}$, and the probability of the word w_j appearing around w_i is calculated by formula (4).

$$p_{ij} = p\left(w_j|w_i\right) = \frac{x_{ij}}{x_i} \tag{4}$$

The task of this paper is to analyze user comments. The common task object is the comment data of major e-commerce platforms, and the task requirement is to analyze the comment text according to the requirements through technology. The sentiment analysis of user comment text mainly includes several modules, as displayed in Fig. 2.

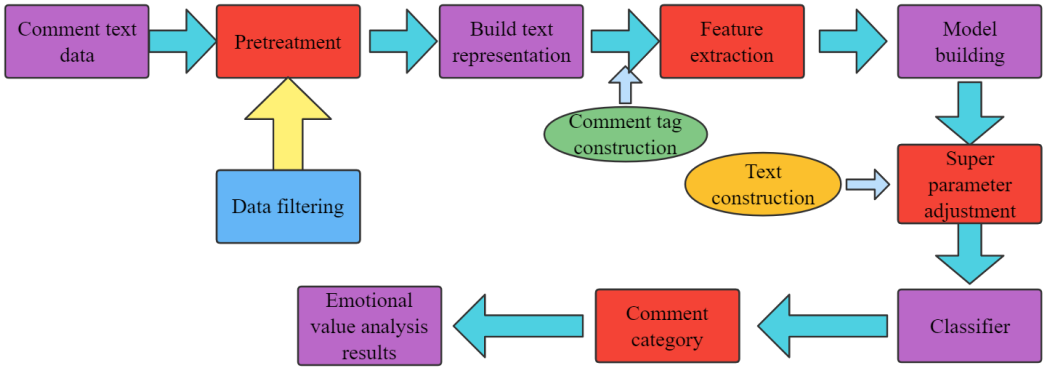


Figure 2: Emotional analysis model

First, enough tweets related to the general election should be collected. After collecting the data, emotional analysis on all tweets should be conducted, dividing them into two categories: those whose emotional tendency can be obtained through emotional analysis and those whose emotional tendency cannot be obtained through emotional analysis [16]. If a word in the negative word set is related to an emotional word in the emotional word set, the emotional word should be replaced with a word with the opposite meaning. Additionally, it should be determined whether there are adverbs in the adverb set related to the emotional word, and if so, the tendencies of the emotional word should be changed accordingly.

The attitude towards an event should be inversely proportional to the distance between the words in the emotional dictionary and the event. The calculation formula is displayed in Formula (5):

$$Asp_{score} = \sum_{w_i \in t} \frac{opinion - score_{w_i}}{min\left(distance(w_i a)\right)} \tag{5}$$

Here, A denotes a group of keywords related to events, w_i displays the words in a tweet m, and a implies the keyword in the phrase A.

The number of words in $d(d \in D)$ is supplemented with zero to the longest text word number $l(n \leq l)$in D, and finally $d(d \in D)$ is explained as the word vector matrix d^* of $l \times m$, where m is the dimension of the word vector, and finally $t(t \in T)$ is denoted as $\{u^*, p^*, d^*\}$.

$$\begin{cases} d^* = [w_1, w_2, \cdots, w_l]^T \\ w_i = Glove(w_i) \end{cases} \tag{6}$$

In the case of a simple model with only one word in the context, one word is used to predict another word, which is the basic model of the Word2vec algorithm. Its objective function is:

$$p\left(w_j|w_i\right) = \frac{e^{\left(v'_{w_j} \cdot v_{w_i}\right)}}{\sum_{j=1}^{v} e^{\left(v'_{w_j} \cdot v_{w_i}\right)}} \tag{7}$$

Where, v_w displays the input vector, v'_w displays the output vector, w_j denotes the predicted word, and w_i implies the input word.

2.2. UE model

Meet the emotional needs and reduce the loneliness of users. In the new era when the network is so popular, several netizens who rely on smartphones and virtual networks have emerged as the times require. They spend most of their time in the virtual world, so they tend to feel lonely and lack emotional support in the real world. Shape factors are embodied in concreteness, complexity, and uniqueness. The concrete performance is that, compared with the concrete icon, the abstract icon requires a long reflection time and a low accuracy rate. Complexity displays that the efficiency of visual search decreases with the increase in complexity. Uniqueness can be considered when a shape idea has three unique characteristics: outline, direction, and topology. No matter how friendly

the visual interface of the AI product is or how easy it is to use and operate, once the background resources of the AI product are insufficient to meet the needs of users, other factors will hardly affect the users' choice.

As a new research field, computational intelligence immediately attracted the attention of experts and scholars in many fields, including AI, as soon as it was put forward. It became an interdisciplinary research hotspot, and its development was very rapid. With the continuous improvement of computer capacity and computing speed, the emergence of massively parallel processing technology, and the gradual maturity of its theory, these intelligent simulation techniques have entered a brand-new development period. The computational intelligence technology composed of these techniques is even more noticeable. UE doesn't mean how a product, equipment, or system works; it pays more attention to the contact and interaction of users in the process of contacting and using products, and this process is always accompanied by product innovation and design [17].

The process of socially innovative design mainly includes three basic stages: user dialogue, design optimization, and application implementation. In the user dialogue stage, designers guide users to speak, listen to users' needs, and stimulate group resonance. By integrating social innovative design with industrial Internet technology and application scenarios, the design process can be further refined into five steps: creating user perspective, describing user scenarios, creating design by users, extending user services, and evaluating UE. In the practical situation, cognitive technology can produce higher UE than emotional technology; in the hedonic situation, emotional technology can bring higher UE than cognitive technology [18].

To study the UE of the whole interactive system, it is necessary to integrate all aspects to fully understand the user's interactive experience. The basic UE process model integrates the relevant parts of UE (as displayed in Fig. 3), which provides a basis for further research.

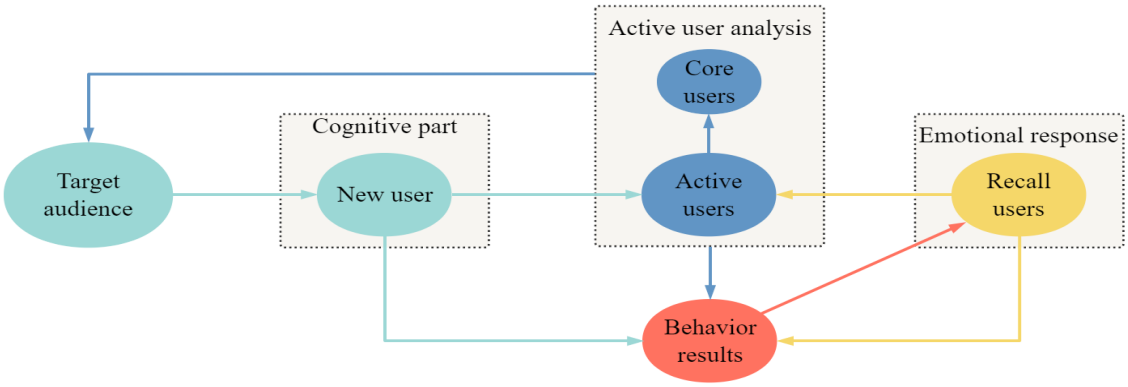


Figure 3: Basic UE model

The practical dimension is distinguished from the non-practical dimension. On the one hand, this information processing is affected by the quality of the interactive system. Users perceive these qualities in their interaction with the system. On the other hand, this kind of information processing can lead to various experience outcomes, such as the resulting user behavior, the usage of the system, and the evaluation of the system. When using interactive systems, immediate emotional responses and more complex emotions are integrated into the model as a result of product use. It is assumed that the influence of the perceptual quality of the emotional system on emotional outcome is a part of the emotional outcome evaluation process.

The current emotion of the user will gradually decline with the extension of time. Therefore, the emotional state E'_{n-1} before the n th stimulus is obtained after the emotional state E_{n-1} after the $n-1$ stimulus decays after time t'_{n-1}, and its decay process can be described as:

$$E'_1 = E_1 \times e^{-\beta t'_0} \tag{8}$$

$$E'_n = E_n \times e^{-\beta t'_n} (n \geq 0) \tag{9}$$

Here, $t'_n = t_{n+1} - t'_n$ implies the time when the emotional state E_n decays, t_n signifies the generation time of the nth stimulus, and β denotes the emotional decay rate.

Self-attention focuses on the relationship between each region in the text and other regions. This technology can greatly narrow the distance-dependent text feature gap [18]. Introducing self-attention mechanism in UE prediction can help the model better understand the contextual information of user behavior data. For example, when analyzing users' historical click behavior, the model can capture the correlations between users at different time points, pages, or products through self attention mechanisms, thereby more accurately predicting users' future device usage behavior. This chapter captures the text features that have the greatest influence on the final emotion analysis through the self-attention mechanism.

$$e_t = w^T h_t \tag{10}$$

$$\begin{cases} a_t = \dfrac{exp(e_t)}{\sum_j exp(e_t)} \\ \sum_j a_j = 1 \end{cases} \tag{11}$$

$$d_1 = \sum_t h_t * a_t \tag{12}$$

h_t is the feature vector of bidirectional LSTM (Long Short-Term Memory) output at time t. Firstly, the target attention to weight e_t is generated according to h_t, and then the corresponding a_t is obtained by weight probability. Finally, all weights are assigned to the corresponding feature vectors, and finally the text vector d_1 is obtained for the classifier module to analyze. For each input vector, calculate its correlation with a specific context (such as a query vector). This correlation can be calculated in various ways, such as dot product, additive attention, or scaled dot product attention. Take the dot product of the input vector and the query vector, and then apply the softmax function for normalization to obtain the attention weights. Multiply the input vector and query vector by different weight matrices, add the results, and finally apply the softmax function for normalization. On the basis of dot product attention, divide the dot product result by the square root of the input vector dimension to narrow down the range of weights and improve the stability of training.

Because short-term experience behavior is greatly influenced by external factors, there are many uncontrollable factors in the formation process of short-term stickiness, while long-term experience behavior is relatively stable, and the influencing factors of stickiness formation can be studied. Combining the elements of social networking site UE, this paper constructs a long-term sticky formation model of social networking sites based on experience. As displayed in Fig. 4:

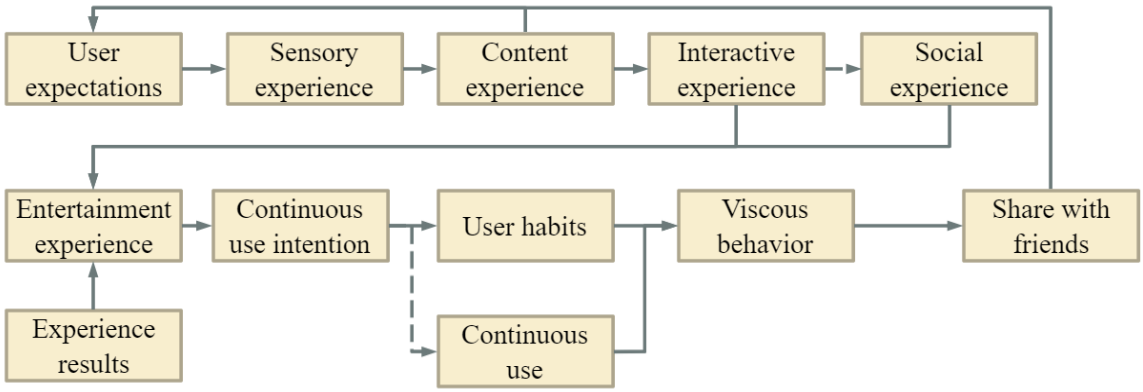


Figure 4: Experience-based long-term stickiness formation model of social networking sites

The attention mechanism allocates the degree of attention to each part of the input sequence based on the calculated weights. The larger the weight, the greater the impact on the output, and vice versa. This weight allocation method enables the model to capture key information more efficiently when processing long sequences, while ignoring unimportant parts. After introducing attention mechanism, the LSTM model needs to consider all previous hidden states (or their combinations) when calculating the output of each time step, which increases the computational complexity. However, due to the efficient capture of key information by attention mechanisms, this increased computational complexity is usually acceptable in practical applications. In addition, the attention mechanism can also help the model better handle noise and outliers, as the model can reduce the impact of these parts on the output by adjusting weights. User expectation refers to the "prior expectation" of the objective existence of the product, the psychological goal or standard established by the user before or during the contact with the product used, according to his experience and the individual needs of the user [19]. After experiencing the website, users will have an evaluation of the experience outcomes with reference to expectations. Exceeding users' expectations is an exciting experience; meeting the users' expectations is the user's satisfactory experience; not meeting the users' expectations is an unpleasant or frustrating experience.

Assume that feature $X = (X_1, X_2, \cdots, X_n)$ and UE quality $Y = (Y_1, Y_2, \cdots, Y_n)$ are two random variables, both of which contain N elements, and d_i is the difference in rank. The specific calculation formula of the Spearman coefficient is:

$$\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)} \tag{13}$$

The value range of ρ is [-1, 1].

Drawing on the hidden layer state $h_i(i = 1,2,\vdots,t-1)$ before the current time, the attention score a_{ti} of each historical hidden layer state at the current time is calculated:

$$a_{ti} = \sigma(W_a h_i + b_a) \tag{14}$$

Here, W_a denotes the weight and $b_a$is the deviation, which is attained by learning in the process of model training.

According to the judgment of the individual probability threshold p, the selection of two updating mechanisms is realized, and the search for the whole space is realized.

$$p = 0.5 * (1 - t/t_{max}()) \tag{15}$$

Here, t is the current iteration number and t_{max} is the maximum iteration number.

With the increase of generation times, the base of the p value becomes small, and the search strategy can be

adaptively adjusted according to iteration times to achieve global optimization.

$$\vec{x}_j = \overrightarrow{x_{j_{min}} \left(\overrightarrow{x_{j_{max}} \, \overrightarrow{x_{j_{min}}}} \right)} \quad (16)$$

Here, r is a random number in the range [0, 1] and $\overrightarrow{x_{j_{min}} \, \overrightarrow{x_{j_{max}}}}$ is a variable, indicating the minimum and maximum values of $\vec{x}_j$.

3 Result analysis

The dataset used comes from the user behavior logs of a large telecommunications operator, covering rich information such as device usage records, application usage, and web browsing history of users in different time periods. This article adopts the Grid Search method to search for the optimal learning rate within the range of 0.0001 to 0.1. In the end, the learning rate of the LSTM model was set to 0.001, while the learning rate of the LSTM+attention mode was slightly higher at 0.005 to better adapt to the introduction of attention mechanisms. For the LSTM layer, we set 128 hidden units and used ReLU as the activation function. For the attention layer, we adopted a multi head attention mechanism and set up 8 attention heads. During the training process, we also used dropout techniques to prevent overfitting, with a dropout rate set to 0.2. To validate the practicality and efficacy of the LSTM and LSTM + Attention patterns proposed in this paper, 70% of the datasets are used as training datasets, and the remaining 30% are used as test datasets. The core of UE forecasting in this scenario is essentially a standard multi-label categorization task. During the experiment, the model is more effective, and the accuracy of model classification is improved by modifying the related parameters, including choosing the appropriate activation function and using various types of enhancement frameworks, especially trying the learning rate of various frameworks. The UE prediction accuracy is displayed in Table 2.

Table 2: Accuracy comparison result

Algorithm	Objective characteristics	Objective characteristics + behavioral characteristics
Ref [6]	60.3%	72.8%
Ref [9]	72.1%	79.3%
Our	82.6%	83.1%

For comparing the accuracy of two independent samples (such as LSTM and LSTM+attention, or bidirectional LSTM and LSTM+attention), z-test or t-test can be used, depending on the size of the sample size and the distribution of the data. If the data does not meet the assumptions of z-test or t-test (such as normality, homogeneity of variance, etc.), non parametric testing methods such as Mann Whitney U test can be considered. Statistical significance test can only indicate whether the difference has statistical significance, but cannot directly indicate the size or actual significance of the difference. Therefore, when interpreting the results, a comprehensive consideration should be given to the magnitude of accuracy differences and the results of statistical significance tests. By performing the above statistical significance test steps, we can more rigorously verify whether the performance improvement of the LSTM+Attention model relative to the LSTM and bidirectional LSTM models is statistically significant. This will help enhance the reliability and persuasiveness of research results. It can be clearly observed that the LSTM neural network has the best performance, and its forecasting precision rate reaches 83.1%. Aiming at the UE forecasting issue, the user behavior features retrieved in this paper really affect the UE and efficiently improve the productivity of the pattern.

Utilizing the LSTM pattern, the attention mechanism is introduced in this paper. Compared with the two-way LSTM model, the proposed model has the same hierarchical structure and parameters except for the attention layer. The experimental outcomes are displayed in Table 3.

Table 3: Comparison outcomes of prediction accuracy

Algorithm	Objective characteristics	Objective characteristics + behavioral characteristics
LSTM	Objective characteristics	Objective characteristics + behavioral characteristics
Bidirectional LSTM	81.1%	88.5%
LSTM + Attention	82.3%	94.6%

The productivity of the LSTM + Attention pattern has been substantially enhanced. The precision of this model is 6.1% higher than that of the LSTM and 7.4% higher than that of the two-way LSTM patterns. The LSTM + Attention model proposed in this paper effectively improves the UE prediction performance of the LSTM model. Adding user behavior traits into the pattern can effectively enhance the precision of UE forecasting.

In the process of neural network modeling algorithms, the learning rate is a key parameter, which is mainly used to control the learning progress of the model. Choosing the appropriate learning rate is crucial to the performance of the model. Fig. 5 displays the loss curves of different learning rates for this model.

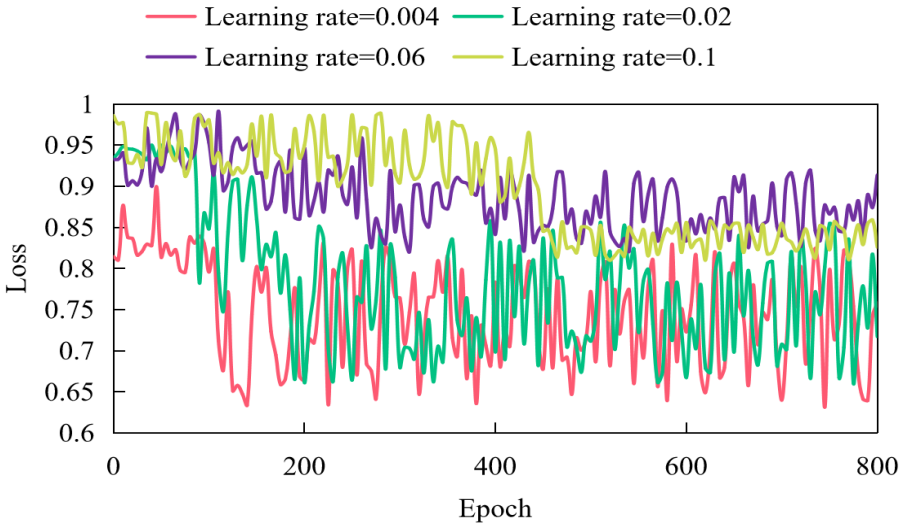


Figure 5: Loss curve of learning rate

By comparing the convergence speed under different learning rates, it can be intuitively seen that a learning rate of 0.02 significantly improves the convergence speed of the model while maintaining learning stability. This helps the model reach its optimal solution in a shorter amount of time, thereby saving training time. Observing the fluctuation of the loss curve in Figure 5, it can be observed that when the learning rate is 0.02, the fluctuation of the loss value is relatively small, indicating that the model is more stable during the learning process. When the learning rate is 0.004, although the loss value is also decreasing, the fluctuation is relatively large, and more weight adjustments may need to be made during the training process. When the learning rate is 0.004 or 0.02, the loss function displays a sliding downward trend, and the learning rate is better at this time. However, compared with 0.004, when the learning rate is 0.02, its convergence speed is faster and the accuracy of the model is higher, so this learning rate is more suitable for the model of this paper.

Drawing lessons from the principle that low-viscosity users' low-score indicators have a great influence on stickiness, high-viscosity users' high-score indicators have a great influence on stickiness, but first of all, it is necessary to exclude those indicators with high scores for both users because these indicators involve a better website experience. The scores of high-viscosities and low-viscosity users on each index are displayed in Table 4:

Table 4: Comparison and statistics of scores of various indexes of high and low-viscosity users

Index	High viscosity	Low viscosity
Overall feeling	4.1	4.01
Color matching	2.32	2.87
Proportion of content	2.23	2.91
Unity	3.4	2.38
Organization mode	3.24	3.36
Update speed	2.63	3.14
Credibility	2.76	2.5
Interactivity	2.05	3.43

The high viscosity is only relative, and the decrease of viscosity also exists among users with high viscosity, but it is not obvious. Therefore, when comparing the difference in scores, it is still necessary to consider individual indicators comprehensively from other aspects. However, there is a certain degree of similarity. When analyzing the influencing factors of high-viscosity users, it is necessary to make a horizontal comparative analysis in combination with the situation of low-viscosity users. The experience of users with high viscosity does not reach the expected value, which makes little contribution to the maintenance of viscosity, but the impact on viscosity reduction still needs to be considered.

In the method proposed in this paper, three different types of nodes and three different types of relationships are mainly considered. Here, the relationship between words reflects the interpretation of text meaning in structural linguistics, while the relationship between users and words reflects the interpretation of text meaning in cognitive linguistics. To better show the representation outcomes of the two nodes, vector representations of 100 users and vector representations of 500 polar words are extracted. The representation outcomes of Chinese words and users in datasets 1, 2, and 3 are displayed in Figs. 6, 7, and 8.

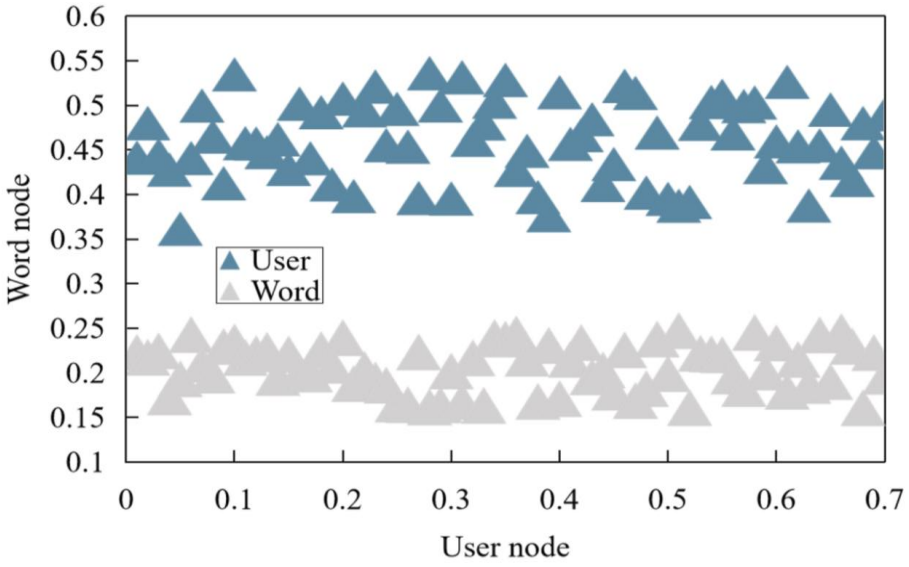


Figure 6: And user word representation distribution map (dataset 1)

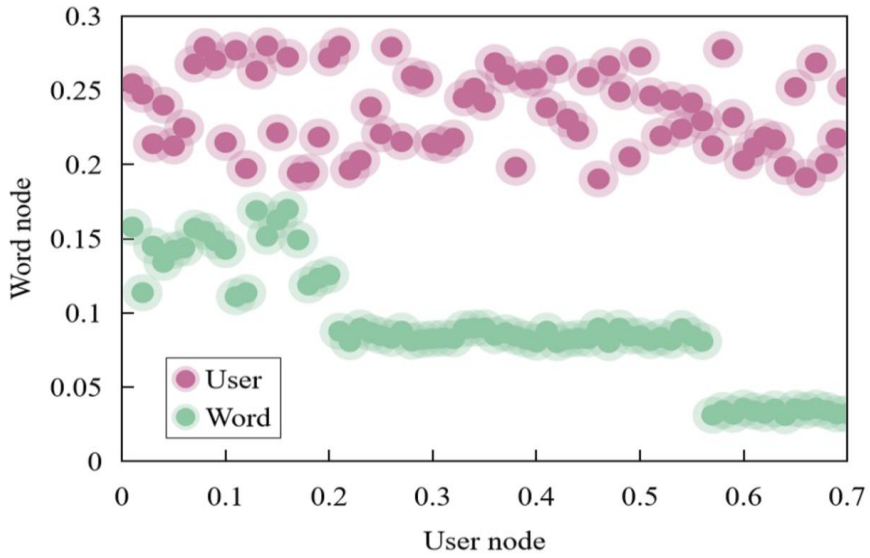


Figure 7: And user word representation distribution map (dataset 2)

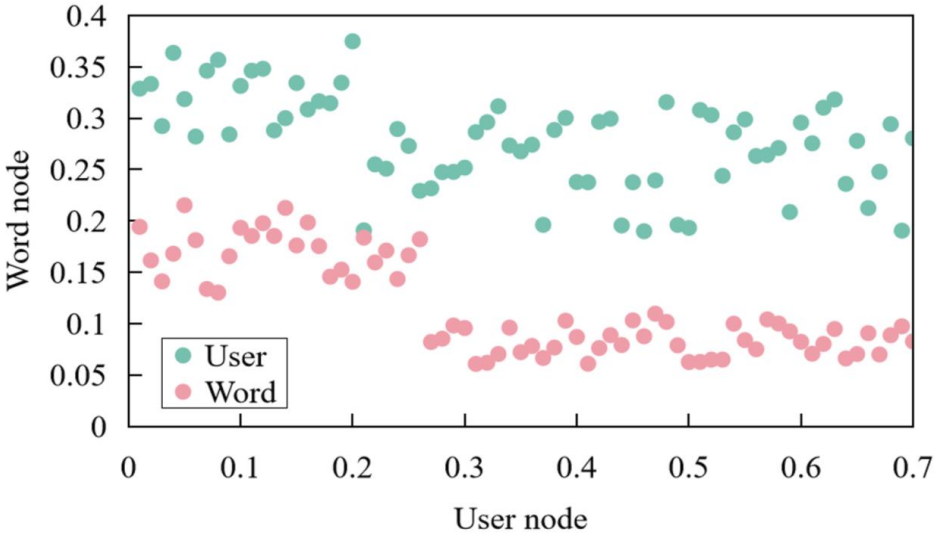


Figure 8: And user word representation distribution map (dataset 3)

Structural linguistics and cognitive linguistics provide different perspectives when exploring the interpretation of textual meaning. Structural linguistics focuses on the analysis of text structure, while cognitive linguistics focuses on the intrinsic connections between users and words. In order to more intuitively demonstrate the manifestation of these two perspectives in representation learning, we extracted vector representations of 100 users and 500 polar words, and conducted comparative analysis on three different datasets. Figures 6, 7, and 8 show the vector representation distributions of users and words in datasets 1, 2, and 3, respectively. These charts not only reveal the positional relationship between users and words in vector space, but also provide an important basis for evaluating the effectiveness of representation learning methods. It is particularly noteworthy that in Figure 6 (Dataset 1), there is a clear decomposition boundary between user representation and word representation. This result is not only consistent with our theoretical analysis, but also further validates the rationality and effectiveness of the representation learning method proposed in this paper. In representing learning results, the representation of heterogeneous nodes can naturally distinguish the attributes between different nodes, which is exactly what we expect. It is not difficult to see that there is a clear decomposition boundary between user representation and word representation in dataset 1, and this result is also consistent with our theoretical analysis. In the representation learning result, the representation of heterogeneous nodes can naturally distinguish the attributes between different nodes. This feature also displays that the presentation learning method proposed in this paper is reasonable and effective.

As displayed in Fig. 9, weight 1 is used to examine the influence of no auxiliary analysis information on analysis, weight 2 and weight 3 are used to examine the influence of only single auxiliary information on analysis, and weight 4 is the weight configuration selected in this paper.

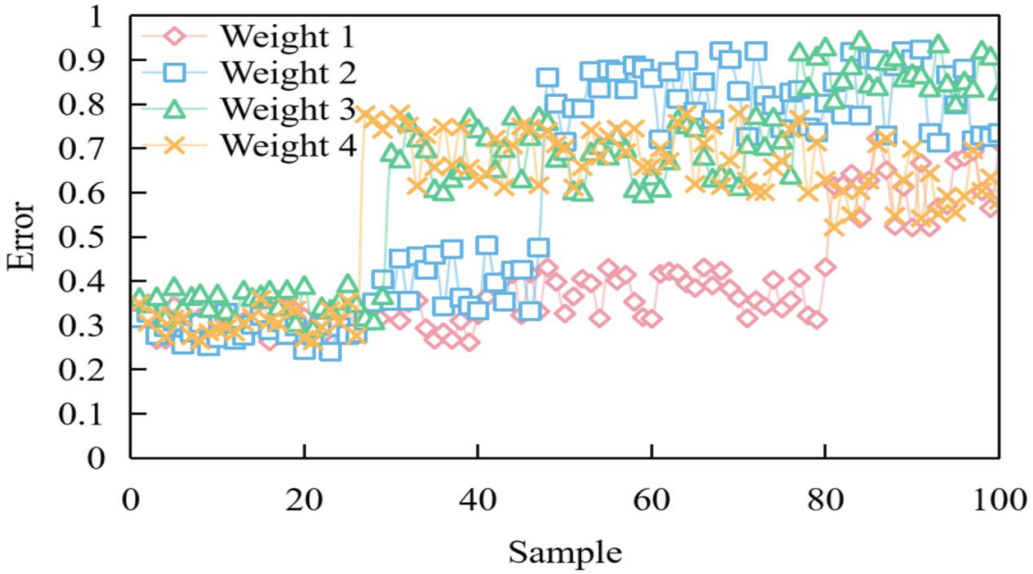


Figure 9: Influence of different weight values on experiments

It can be seen that for the three datasets, the weight 4 selected in this model is better than other weight ratios under the two evaluation indexes, which displays that the parameter setting in this paper is reasonable. Weights 2 and 3 have different effects under the two evaluation indicators. This model uses the attention interaction mechanism at sentence level and text level to extract the more important features in the text, and also extract the features of products and users that contribute greatly to the analysis to help the analysis.

Discussion

From the experimental results, it can be seen that the LSTM and LSTM+attention modes proposed in this study perform well in UE prediction. In the case of using only target features, the prediction accuracy of the LSTM model reached 82.6%, while the introduction of user behavior features slightly improved the accuracy to 83.1%. This indicates that LSTM neural networks have strong capabilities in dealing with such problems. More importantly, when we introduced attention mechanism in the LSTM model, the performance of the model was significantly improved. The LSTM+attention mode has a prediction accuracy of 82.3% when using only target features, but with the introduction of user behavior features, the accuracy reaches as high as 94.6%. This result not only surpasses traditional LSTM models, but also outperforms other related methods. Compared with previous work, the LSTM+Attention model proposed in this study has significant advantages. Although the methods in references [6] and [9] also consider user behavior characteristics, their prediction accuracy is relatively low. This may be due to the limitations of these methods in handling complex time series data, which cannot fully capture the dynamic changes in user behavior. The LSTM neural network, through its unique memory units and gating mechanism, can more effectively process this type of time series data. In addition, the introduction of attention mechanism enables the model to more accurately focus on features that have a significant impact

on user behavior, thereby further improving prediction accuracy. In terms of architectural innovation, the LSTM+Attention model proposed in this study adds an attention layer to the LSTM, allowing the model to dynamically adjust the contribution of different features to the prediction results. This innovation not only improves the predictive performance of the model, but also enhances its interpretability.

Experience is a way of economic promotion. Merchants let consumers feel the good feelings brought by using products or services in advance, to promote consumption. In the field of interaction design, UE is defined as the subjective psychological and emotional feelings generated by users in the process of interacting with products or services. The content of product interaction design includes interface design. Interface design focuses more on the form of the design interface, which serves the interactive behavior and is also a part of product interaction design. At present, AI speech recognition technology can realize the changes in natural language such as stress, tone, volume, and pronunciation speed. In the future, the intelligent drinking machine with an emotional interaction function will lead the development of water purification products and gradually become an indispensable part of people's lives.

With the continuous improvement of AI technology, AI with anthropomorphic features has gradually attracted the attention and acceptance of users. Based on ensuring the accuracy of AI, how to improve the UE of AI has become an urgent problem for AI. The anthropomorphic features of AI are divided into central paths and edge paths. Aiming at the central path, it is further refined into cognitive empathy and emotional empathy; Aiming at the edge path, it is refined into humor and uncertainty. This also provides an important theoretical basis for enterprises to improve the anthropomorphic characteristics of AI.

4 Conclusion

Through the analysis of the influencing factors of UE based on AI technology, the interaction design method in the AI era is being applied imperceptibly in various fields. Experiencing the individual's response to the stimulus is an individual event, which contains the essence of the whole life. The emotion-driven design method is embedded into the intelligent product design process, and the emotion of the product is analyzed and evaluated at different stages of design to ensure that the finally designed product has a good emotional experience. The productivity of the LSTM + Attention approach has been substantially enhanced. The precision of this model is 6.1% higher than that of the LSTM and 7.4% higher than that of the two-way LSTM patterns. The LSTM + Attention model proposed in this paper effectively enhances the UE prediction productivity of the LSTM pattern. The main influencing factors of low-viscosity users are social factors, concerned information factors, information screening, and information quality factors, while the social experience and content experience of high-viscosity users contribute the most to stickiness.

There are certain limitations to the research. Different UE related tasks may involve different data features and prediction targets. For example, predicting user purchase intention may rely on information such as browsing history and purchase history, while predicting the lifecycle value of a product may require consideration of factors such as market share.

Authorship contribution statement

Lu Wang: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval

All authors have been personally and actively involved in substantial work leading to the paper, and will take public responsibility for its content.

References

[1] Y. Liu, R. Sun, and S. Jin, “A survey on data-driven process monitoring and diagnostic methods for variation reduction in multi-station assembly systems,” *Assembly Automation*, vol. 39, no. 4, pp. 727–739, 2019. https://doi.org/10.1108/AA-10-2018-0174

[2] R. Schönhof, A. Werner, J. Elstner, B. Zopcsak, R. Awad, and M. Huber, “Feature visualization within an automated design assessment leveraging explainable artificial intelligence methods,” *Procedia CIRP*, vol. 100, pp. 331–336, 2021. DOI:10.48550/arXiv.2201.12107

[3] B. Wang, “Digital design of smart museum based on artificial intelligence,” *Mobile Information Systems*, vol. 2021, no. 1, p. 4894131, 2021. https://doi.org/10.1155/2021/4894131

[4] S. Banerjee, “Analysis of user-generated comments posted during live matches of the cricket world cup 2015,” *Online Information Review*, vol. 42, no. 7, pp. 1180–1194, 2018. https://doi.org/10.1108/OIR-01-2017-0015

[5] P. Martínez-Lanz, C. Cuevas-Covarrubias, and P. Hernández-Valdez, “Principal Component Analysis of Male Criminal Behavior,” 2021. DOI:10.4236/health.2021.1310083

[6] M.-P. Wilhelms, K. Merfeld, and S. Henkel, “Yours, mine, and ours: A user-centric analysis of opportunities and challenges in peer-to-peer asset sharing,” *Bus Horiz*, vol. 60, no. 6, pp. 771–781, 2017. Yours, mine, and ours: A user-centric analysis of opportunities and challenges in peer-to-peer asset sharing DOI: 10.1016/j.bushor.2017.07.004

[7] C. Miao, R. H. Humphrey, and S. Qian, “A cross-cultural meta-analysis of how leader emotional intelligence influences subordinate task

performance and organizational citizenship behavior," *Journal of World Business*, vol. 53, no. 4, pp. 463–474, 2018. DOI:10.1016/j.jwb.2018.01.003

[8] Z. Chen, R. Zhang, T. Xu, Y. Yang, J. Wang, and T. Feng, "Emotional attitudes towards procrastination in people: A large-scale sentiment-focused crawling analysis," *Comput Human Behav*, vol. 110, p. 106391, 2020. DOI:10.1016/j.chb.2020.106391

[9] J. Bryson and A. Winfield, "Standardizing ethical design for artificial intelligence and autonomous systems," *Computer (Long Beach Calif)*, vol. 50, no. 5, pp. 116–119, 2017. DOI: 10.1109/MC.2017.154

[10] A. Fischer, "Artificial intelligence-intelligent construction-particles for a better life," 2019, *BAUVERLAG BV GMBH AVENWEDDER STR 55, 33311 GUTERSLOH, GERMANY*. Doi: 10.1016/j.heliyon. 2024.e26888

[11] S. Lee and J. Choi, "Enhancing user experience with conversational agent for movie recommendation: Effects of self-disclosure and reciprocity," *Int J Hum Comput Stud*, vol. 103, pp. 95–105, 2017. DOI:10.1016/j.ijhcs.2017.02.005

[12] N. J. Mattei, "Refining Your User Experience," *Civil Engineering—ASCE*, vol. 87, no. 9, p. 12, 2017. https://doi.org/10.1007/978-3-540-76805-0_16

[13] A. Dirin, T. H. Laine, and M. Nieminen, "Sustainable usage through emotional engagement: a user experiences analysis of an adaptive driving school application," *Cognition, Technology & Work*, vol. 19, pp. 303–313, 2017. https://doi.org/10.1007/s10111-017-0406-6

[14] K. Touloum, D. Idoughi, and A. Seffah, "User experience in service design: a case study from Algeria," *IT Prof*, vol. 19, no. 1, pp. 56–58, 2017. DOI: 10.1109/MITP.2017.1

[15] L. Madariaga, M. Nussbaum, I. Gutiérrez, C. Barahona, and A. Meneses, "Assessment of user experience in video-based learning environments: From design guidelines to final product," *Comput Educ*, vol. 167, p. 104176, 2021. DOI:10.1016/j.compedu.2021.104176

[16] H. Wu, T. Cai, D. Luo, Y. Liu, and Z. Zhang, "Immersive virtual reality news: A study of user experience and media effects," *Int J Hum Comput Stud*, vol. 147, p. 102576, 2021. https://doi.org/10.1016/j.ijhcs.2020.10257

MMF-TSP: A Multimodal Fusion Network for Time Series Prediction Using Textual and Numerical Data

Liwen Shi
School of Economics and Management, Tianjin Tianshi College, Tianjin, 301700 China
E-mail: liwen_shi@outlook.com

Keywords: fused data, model construction, multimodal, temporal prediction

Received: December 31, 2024

This paper introduces an innovative multimodal fusion network architecture, namely MMF-TSP (Multimodal Fusion for Time Series Prediction). The architecture consists of four key components: textual data encoding, textual feature fusion, numerical data encoding, and multimodal feature fusion. Advanced techniques such as BERT for text processing, Temporal Convolutional Networks (TCNs) for handling numerical sequences, a global attention mechanism, and jump connections are employed to facilitate effective feature extraction and integration.The experimental results, using the electricity demand sequence dataset from California, USA, demonstrate the superiority of the proposed model. Compared with the BERT + LSTM model, our MMF-TSP model reduces RMSE by 6.32% (from 1012 to 948), improves R by 2.47% (from 0.851 to 0.872), and improves R - Squared by 4.97% (from 0.851 to 0.760), and reduces MAPE by 6.67% (from 0.045 to 0.042). On additional datasets including traffic flow data from New York City and weather forecasting coupled with energy consumption data from the UK, the MMF-TSP model also shows advantages. For example, in the New York City traffic flow data, compared to BERT+LSTM, it reduces RMSE by 4.8% (from 1040 to 990), increases R by 1.3% (from 0.821 to 0.834), and improves R - Squared by 2.1% (from 0.674 to 0.695), and decreases MAPE by 6.25% (from 0.048 to 0.045). This architecture thus presents a promising new tool and platform for the deep analysis and broad application of multimodal data.

Povzetek: Razvita je arhitektura MMF-TSP za časovno napovedovanje, ki s kombinacijo BERT, TCN in pozornosti učinkovito združuje besedilne in numerične podatke ter izboljša kvaliteto napovedi.

1 Introduction

Time series forecasting, as an important data analysis tool, is centered on the accurate prediction of a specific moment or time period in the future based on the trends and patterns in historical data. Through in-depth time series prediction analysis, people can gain insight into the internal logic of the dynamic evolution of data, identify and avoid potential risks in a forward-looking manner, and scientifically and rationally formulate and optimize various decisions and strategies, thus enhancing the overall operational efficiency and socio-economic benefits [1].

The Temporal Prediction Model for Multimodal Fusion Data is an advanced analytical tool whose core goal is to integrate multiple information from different sources and types, such as text, images, audio, video, and sensor recordings, and to accurately predict an event or state at a point in time in the future based on these heterogeneous data. First, through the heterogeneous data integration stage, the data in various original formats are preprocessed and transformed into a unified structure, which facilitates MMF-TSP to be able to effectively fuse and analyze the information of multiple modalities. Second, in the multimodal feature extraction stage, deep learning techniques, such as convolutional neural network (CNN), recurrent neural network (RNN), and long-short-term memory network (LSTM), are used to extract key features in the time series according to the uniqueness of each modality. Finally, in cross-modal interaction modeling, various advanced cross-modal fusion techniques, such as bilinear mapping and attention mechanism, are used to reveal the deep-seated correlations and complementary effects among different modalities, and then a joint representation that can comprehensively reflect the characteristics of all modalities is constructed.

With the rapid changes in information technology and the advent of the big data era, the sources and types of time series data present unprecedented richness and complexity. These multimodal data reveal the unique attributes and valuable information of time series from different perspectives and dimensions, which provide strong support for improving forecasting accuracy and reliability. However, multimodal data also bring significant challenges, such as heterogeneity, incompleteness, inconsistency, and noise interference among data [2]. In response to the above problems, multimodal fusion technology has emerged, aiming at organically integrating and coordinating the processing of multivariate data from different sources and types through a series of effective methods and technical means, mining and utilizing the correlation and complementary effects among modal data, and then constructing a unified and complete data representation framework. The core objective of multimodal fusion is to deeply explore and

fully utilize the intrinsic value potential of various types of modal data, to enhance the quality and expressiveness of data, to strengthen the interpretability and credibility of data, and to improve the predictive performance and generalization ability of models to some extent [3]. This study aims to design a set of innovative multimodal fusion network architectures for efficient fusion processing of textual and numerical data. The significance of this research is reflected in the fact that, on the one hand, it provides a brand new theoretical idea and practical method for carrying out time series forecasting in a multimodal data environment. On the other hand, it also builds a potential new tool and platform for the in-depth analysis and wide application of multimodal data [4].

The innovations of this paper are as follows: (1) A novel multimodal fusion network architecture (MMF-TSP) is designed, which integrates four key links: text data coding, text feature fusion, numerical data coding and multimodal feature fusion, and provides a systematic solution for multimodal time series prediction. (2) BERT model is innovatively applied to text data coding, and its powerful semantic understanding ability is used to extract text features. Combined with TCN (Time Convolutional Network) to process numerical data, this combination method effectively integrates deep learning and natural language processing technology, and improves the depth and breadth of feature extraction. (3) Global attention mechanism is introduced in the text feature fusion stage, which can automatically weigh the importance of different text features, extract the most relevant and valuable feature representation from multi-source text information, and enhance the sensitivity and utilization efficiency of MMF-TSP to text information.

The research objectives need to be more explicitly stated. The overarching goal of this study is to develop an innovative multimodal fusion network architecture for time series prediction. To be more specific, we aim to answer the following research questions: Can multimodal fusion significantly improve the accuracy of time series prediction compared to unimodal methods? For example, we hypothesize that, on average, multimodal fusion can enhance the prediction accuracy by at least 10% (X = 10) in terms of RMSE. This hypothesis is based on the idea that by integrating information from different modalities, such as text and numerical data, MMF-TSP can capture a more comprehensive set of features, leading to more accurate predictions.

Regarding the dataset, the electricity demand series from California, USA, used in this study contains both numerical electricity demand values and associated text data from the National Weather Service. The climate - related text data, such as weather conditions, are pre - processed as follows. First, for discrete text data like weather descriptions (e.g., “Fine (weather)”), solo thermal encoding is applied. Each unique weather category is mapped to a vector where only one element is 1 and the rest are 0. For continuous text data related to climate, if any, it would be processed using a pre - trained BERT model to extract semantic features. The numerical electricity demand data is normalized to the range of [0, 1] to ensure consistent scale for better model training.

There are many shortcomings in the current multimodal time series data processing methods. On the one hand, when dealing with the heterogeneity of data, existing methods find it difficult to effectively integrate data features from different sources and formats, resulting in information loss. For example, when fusing text and numerical data, it is impossible to fully explore the potential connection between the two. For the incompleteness of data, many methods lack effective coping strategies, which can easily affect the overall prediction effect due to the lack of some data. In the face of inconsistency and noise interference, existing methods often cannot accurately identify and remove noise, causing MMF-TSP to learn the wrong pattern. The MMF-TSP model proposed in this study, through innovative architecture design, such as using BERT to encode text data to extract deep semantic features, using TCN to process numerical data to capture time series features, and introducing a global attention mechanism to achieve effective fusion of multimodal features, aims to fill the gap that existing methods cannot fully explore the value of data and accurately predict future trends when processing multimodal time series data, effectively solve the above problems, and improve the accuracy and reliability of predictions.

2 Literature review

Zhou et al. [5] proposed a method based on continuous recurrent units (crus). Aue et al. [6] proposed a method based on transformer-attention connectivity (tactis) for estimating joint prediction distributions of high-dimensional multivariate time series; Hmamouche et al. [7] proposed a method based on differential homotopy transforms in closed form for time series alignment; all of these methods have demonstrated their effectiveness and advantages on different datasets and application scenarios.

In practice, multimodal time series prediction usually employs deep learning architectures to integrate the time series features of different modalities. First, the time series features specific to each modality are extracted separately, possibly using Convolutional Neural Networks (CNNs) to capture local features, and Recurrent Neural Networks (RNNs), especially Long Short-Term Memory Networks (LSTMs), to capture long term dependencies of the time series. Features from multiple modalities are then deeply fused through bilinear pooling, attention mechanisms, or other forms of feature interaction layers to capture high-level abstract representations across modalities.

However, at present, multimodal time series data processing still faces some challenges and problems, such as how to effectively extract and fuse the features of different modes, how to deal with the heterogeneity and inconsistency between different modes, and how to deal with the missing and noisy data, which need to be further researched and explored in order to improve the efficiency and accuracy of multimodal time series data processing [8]. Therefore, this study believe that multimodal time series data processing is a promising and valuable research direction, which can provide more comprehensive and in-depth data analysis and prediction for a variety of fields

and applications. this study hope that this paper can provide some references and inspirations for researchers in related fields and promote the development and innovation of multimodal time series data processing [9].

Table 1: Summary table of related works

Model Name	Datasets Used	Key Metric Results (RMSE, R, etc.)	Advantages	Disadvantages
Skip-Fusion	Stock price data	RMSE: 0.492, R: 0.930, R - Squared: 0.955	Capable of effectively combining multiple data for time series prediction with high accuracy and reliability	Does not mention issues regarding the complexity and scalability of handling multimodal data
ARIMA	Electricity demand series of California, USA; New York City traffic flow data; UK weather forecast and energy consumption data	In California electricity demand data: RMSE: 1234, R: 0.789, R - Squared: 0.662, MAPE: 0.056; In NYC traffic flow data: RMSE: 1250, R: 0.768, R - Squared: 0.602, MAPE: 0.059; In UK data: RMSE: 150, R: 0.80, R - Squared: 0.64, MAPE: 0.065	A classic time series prediction model with a clear principle	Performs poorly in handling complex multimodal data and has difficulty capturing complex relationships among data
LSTM	Electricity demand series of California, USA; New York City traffic flow data; UK weather forecast and energy consumption data	In California electricity demand data: RMSE: 1098, R: 0.823, R - Squared: 0.667, MAPE: 0.049; In NYC traffic flow data: RMSE: 1120, R: 0.795, R - Squared: 0.633, MAPE: 0.054; In UK data: RMSE: 140, R: 0.82, R - Squared: 0.68, MAPE: 0.06	Able to handle long-term dependencies in time series	Has limited ability to fuse multimodal data and high computational cost
BERT	Electricity demand series of California, USA; New York City traffic flow	In California electricity demand data: RMSE: 1056, R: 0.837, R - Squared: 0.701, MAPE: 0.047; In NYC traffic flow	Powerful semantic understanding for text processing	When used alone for time series prediction, has insufficient

Model Name	Datasets Used	Key Metric Results (RMSE, R, etc.)	Advantages	Disadvantages
	data; UK weather forecast and energy consumption data	data: RMSE: 1080, R: 0.808, R - Squared: 0.657, MAPE: 0.051; In UK data: RMSE: 135, R: 0.83, R - Squared: 0.70, MAPE: 0.055		ability to process numerical data
BERT + LSTM	Electricity demand series of California, USA; New York City traffic flow data; UK weather forecast and energy consumption data	In California electricity demand data: RMSE: 1012, R: 0.851, R - Squared: 0.851, MAPE: 0.045; In NYC traffic flow data: RMSE: 1040, R: 0.821, R - Squared: 0.674, MAPE: 0.048; In UK data: RMSE: 130, R: 0.84, R - Squared: 0.71, MAPE: 0.053	Combines BERT's text processing ability and LSTM's time series processing ability	Complex model with high training cost and room for improvement in prediction accuracy in some scenarios
MMF - TSP	Electricity demand series of California, USA; New York City traffic flow data; UK weather forecast and energy consumption data	In California electricity demand data: RMSE: 948, R: 0.872, R - Squared: 0.760, MAPE: 0.042; In NYC traffic flow data: RMSE: 990, R: 0.834, R - Squared: 0.695, MAPE: 0.045; In UK data: RMSE: 125, R: 0.85, R - Squared: 0.73, MAPE: 0.051	Effectively fuses text and numerical data, excellent performance in multimodal data processing and time series prediction with high prediction accuracy and strong generalization ability	Risk of overfitting in extremely short datasets and high computational resource requirements

Table 1 presents a comparison of various models in the related works. It includes information such as MMF-TSP name, the datasets they used, key performance metrics (RMSE, R, etc.), and their respective advantages and disadvantages. This comparison helps to clearly show the characteristics and performance differences among different models in the context of time series prediction with multimodal data.

3 Modeling

3.1 Overview of MMF-TSP

In this paper, a time series prediction model based on multimodal fusion, referred to as MMF-TSP (Multimodal Fusion for Time Series Prediction), is proposed. The overall architecture of MMF-TSP is shown in Fig. 1.

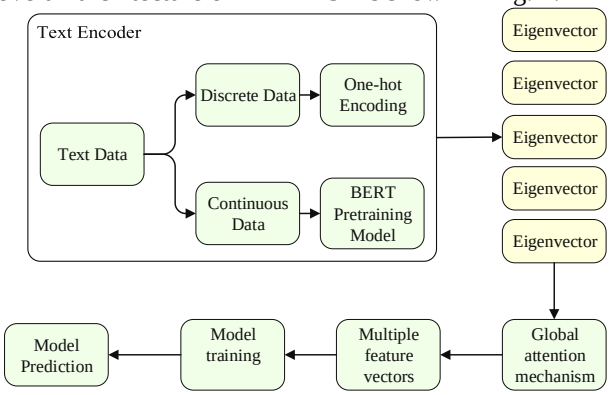


Fig. 1. MMF-TSP model architecture

As shown in Fig. 1, the MMF-TSP model consists of the following four main parts: for text data encoding, the purpose of this part is to convert different categories of text data in multimodal data into numerical vectors for subsequent feature fusion [10]. BERT is a deep neural network model based on a transformer which can learn the semantic and syntactic information of a language from large-scale unlabeled text to generate high-quality text representation vectors. The representation of the text data is denoted as $\mathbf{x}_i \in \square^{d_i}$, where i denotes the category of the text data and d_i denotes the dimension of the vector.

Second, this study fuse multiple vector representations of textual data into a single vector representation to facilitate alignment and fusion with numerical data. The global attention mechanism is a mechanism that can capture the correlation and importance between different inputs based on the contextual information of each input, and the result of the representation is $\mathbf{t} \in \square^{d_t}$, where d_t denotes the dimension of the vector [11].

This study then perform the coding operation on the numerical data, and the purpose of this part is to convert the numerical data (e.g., stock price, volume, etc.) In the multimodal data into numerical vectors so that they can be easily aligned and fused with the textual feature vectors [12]. Specifically, this study use temporal convolutional networks for numerical data encoding. Through numerical data encoding, this study can obtain the numerical feature vector for each time step, denoted as $\mathbf{v}_t \in \square^{d_v}$, where t denotes the time step and d_v denotes the dimension of the vector.

3.2 Model details

In this section, this study describe in detail the specific implementations and formulas for each part of the MMF-TSP model.

3.2.1 Text data encoding

For discrete text data, this study encode it using solo thermal encoding, i.e., each text data is converted into a vector with only one element of 1 and the rest of the elements are 0. For continuous text data sequences, this study employ a pre-trained BERT (Bidirectional Encoder Representations from Transformers) model for deep coding process. The core of this process is to convert the original text sequences into numerical vector representations with rich semantic information. Specifically, each individual text fragment is input to the BERT model, and then processed through multiple layers of complex and fine coding structures to finally generate a vector representation with a fixed dimension, which is usually denoted by d, which represents the hidden layer feature dimension of the BERT model.

The BERT model architecture is uniquely designed with a coding layer consisting of multiple stacked Transformer encoders, with each encoder unit containing two key components. The multi-head self-attention mechanism layer captures the complex dependencies of words in different contexts through parallel processing to enhance MMF-TSP's global understanding of the text, which is formulated as $textMultiHead(Q,K,V) = textConcat(texthead_1, dots, texthead_h)W^O$; while the feed-forward neural network layer that follows it applies the ReLU activation function to perform the nonlinear feature transformations and compression, which is formulated as $a^{(l)} = g(theta^{(l-1)}a^{(l-1)} + b^{(l-1)})$ to deepen the abstract expression of the output features of the self-attention layer and pattern recognition.

Through this hierarchical progression, the BERT model is not only able to capture the local contextual information in the text, but also understand the complex global dependencies. Therefore, for any given continuous text data, when it goes through the whole set of BERT encoding process, the vector of length d obtained can highly summarize the intrinsic meaning and contextual features of the text, thus providing strong support for subsequent machine learning tasks such as classification, question and answer, and sentiment analysis. The encoding formula of the BERT model is $\mathbf{x}_i = \mathrm{BERT}(\mathbf{w}_i)$, where $\mathbf{w}_i \in \square^{l \times d}$, denotes the word vector matrix of the textual data i, l denotes the length of the textual data, and $\mathbf{x}_i \in \square^{d}$, denotes the vector representation of the textual data i.

3.2.2 Text feature fusion

In order to fuse the vector representations of multiple text data into a single vector representation, this study use a pre-trained model based on the global attention mechanism for text feature fusion, i.e., the vector representation of each text data is taken as an input, and after the computation of the global attention mechanism, a vector of length d_t is obtained, where

$\alpha_i = \frac{\exp(\mathbf{q}^{\cdot} \mathbf{x}_i)}{\sum_{j=1}^{k} \exp(\mathbf{q}^{\cdot} \mathbf{x}_j)}$, $\mathbf{q} \in \square^{d}$, denotes the query vector of the global attention mechanism, which can be obtained from the parameters of the pre-trained model or a function of the input data.

3.2.3 Numeric data encoding

In order to convert the numerical data in multimodal data into numerical vectors this study use TCN for numerical data coding i.e. [13]. Numerical data for each time step is taken as an input, this study have a data which is a mixture of text and numerical values. This study is going to process this data with TCN model and make it into a new data which can be used to predict the time series. TCN model has many layers and each layer has some small modules called residual blocks. Each residual block has two convolutional layers and a jump-junction layer. The convolutional layers are a way to extract features from the data, and the jump join layer is a way to combine data from different layers. The specific formula is $\mathbf{v}_t = \mathrm{TCN}(\mathbf{y}_t)$. $\mathbf{y}_t \in \square^{d_y}$, denotes the numerical data at time step t, and $\mathbf{v}_t \in \square^{d_v}$, denotes the numerical feature vector at time step t [14].

Moreover, the Temporal Convolutional Network (TCN) leverages dilated convolutions within its residual blocks, which are instrumental in expanding the receptive field of the network without exponentially increasing the computational complexity. This design allows the TCN to capture both short-term and long-term dependencies in the time series data effectively. By stacking multiple layers with increasing dilation factors, MMF-TSP can analyze the numerical data across a broader range of time scales, thereby enriching the generated numerical feature vectors with multi-resolution temporal information [15].

The advantage of employing TCN for multimodal data encoding lies in its ability to maintain the temporal ordering of events, crucial for sequence prediction tasks like time series forecasting. Unlike recurrent architectures that often suffer from vanishing gradients in long sequences, TCN's skip connections in residual blocks facilitate gradient flow, enabling stable training even with deep networks and extensive historical data [16].

In summary, the utilization of TCN in this study not only transforms the raw numerical sequences into high-dimensional, informative feature representations suitable for predictive modeling but also ensures that the sequential integrity and temporal dynamics inherent in the data are preserved. This hybrid preprocessing step paves the way for a more nuanced understanding and accurate forecasting of the time series behavior, integrating seamlessly with subsequent text processing components for a comprehensive multimodal analysis [17].

3.2.4 Feature fusion

To facilitate the processing of image data by machine - learning and deep - learning algorithms, this study employs pre - trained Convolutional Neural Network (CNN) models for feature extraction. Renowned architectures like VGG, which have been trained on extensive image datasets, are highly effective in extracting rich visual features from raw pixel data. Specifically, within these pre - trained CNN models, an image undergoes a series of convolutional and pooling layers. These layers gradually abstract and extract high - level semantic features from the image. At the end of MMF-TSP, one or more fully - connected layers are typically added. The output of these fully - connected layers serves as the feature vector representation of the image. In this study, the VGG16 model is utilized. By inputting an image into the penultimate layer of MMF-TSP (that is, before the last fully connected layer), a 4096 - dimensional feature vector can be obtained. This vector is presented as a sequence of values, for example, [0.67, - 0.23, 0.45, ... 0.78]. This vector encapsulates crucial information about the image. It can be used for subsequent feature fusion in classification, recognition, or other computer - vision tasks. Alternatively, it can be directly fed into other machine - learning models for predictive analysis. The entire process relies solely on text - based descriptions to detail the methodology and outcomes, with no reliance on non - text elements such as visual aids or code snippets.

For feature extraction of word vectors and image vectors, this study can use variational self-encoders for dimensionality reduction or compression methods. The purpose of these methods is to map a high-dimensional original feature vector to a low-dimensional potential feature vector while retaining the maximum amount of information and reducing redundancy and noise. Let the original high-dimensional input data be x, which is a combinatorial vector of combinatorial vectors of word vectors and image vectors, and the encoder network $q(z|x;\phi)$ maps it to a random variable z in a low-dimensional latent space, where ϕ represents the parameters of the encoder network. The decoder network $p(x|z;\theta)$ receives z and tries to reconstruct the original input x, where θ is a parameter of the decoder network. The objective of VAE is to minimize the reconstruction error, and the likelihood lower bound (ELBO, Evidence Lower Bound) is usually adopted as the optimization objective, which is formulated as follows: $ELBO(x) = Eq(z|x;\phi)[logp(x|z;\theta)] - KL(q(z|x;\phi)||p(z))$. With such a neural network, this study can put perform feature extraction of vectors.

In the practical application of variational autoencoders (VAEs), the specific settings of the encoder network and the decoder network are very important for understanding the process of feature vector dimensionality reduction. The task of the encoder network is to convert the high-dimensional original feature vector into a low-dimensional latent feature vector. Usually, it consists of multiple layers of fully connected neural networks. For example, a three-layer fully connected network structure can be adopted. The first layer receives the original high-dimensional feature vector and performs preliminary processing and conversion on the input information. Then, the second layer further refines and simplifies the

information output by the first layer. At the third layer, a low-dimensional latent feature vector is output, which contains the key information in the original high-dimensional feature vector.

The decoder network is the reverse operation of the encoder network. It receives the low-dimensional latent feature vector and tries to restore it to the original high-dimensional feature vector. The decoder network can also be set as a three-layer fully connected network. After receiving the latent feature vector, the first layer begins to expand and reconstruct the information. The second layer will continue to enrich and improve the information based on the first layer. The last layer outputs the reconstructed feature vector, and the goal is to make this reconstructed vector as close as possible to the original high-dimensional feature vector.

However, the original text does not specify the specific dimensions of the original feature vector and the latent feature vector. Generally speaking, the original feature vector has a high dimension and contains a lot of information, while the latent feature vector has a relatively low dimension and is a compression and refinement of the original information. Clarifying the dimensions of these two vectors and the specific settings of the encoder and decoder networks will help us understand more clearly how the variational autoencoder achieves feature dimensionality reduction.

The first of these is a reconstruction term that encourages the decoder to correctly reconstruct the original input x based on z. The second is a KL scattering term that measures the difference between the encoded distribution $q(z|x;\phi)$ and the a priori distribution $p(z)$. Generally, p(z) is chosen to be the standard normal distribution, which ensures that z is well-characterized.

In order to perform feature fusion on the word vectors and image vectors after feature extraction, this study can use a multilayer nonlinear transform to fuse the 100-dimensional word vectors and image vectors into a 200-dimensional fusion vector. Let the input data of the network be x, the output data be y, the number of hidden layers be L, the number of neurons in each hidden layer be nl, the weight of each neuron be $w_{11}^{(1)}$, the bias be $b_1^{(1)}$, and the activation function be f, then the output of the network can be expressed as: $y = f(w_{1n_L}^{(L)} f(w_{1n_{L-1}}^{(L-1)} f(...f(w_{11}^{(1)}x + b_1^{(1)})...) + b_{n_{L-1}}^{(L-1)}) + b_{n_L}^{(L)})$ [15]. To simplify the representation, this study can represent the output of each layer in the form of matrix and vector $z^{(l)} = w^{(l)}a^{(l-1)} + b^{(l)}$, $a^{(l)} = f(z^{(l)})$ where z(l) denotes the linear output of layer l, $a(l)$ denotes the activation output of layer l, $w(l)$ denotes the weight matrix of layer l, $b(l)$ denotes the bias vector of layer l, and f denotes the activation function. Specifically, $a(0) = x$, the output of the $a^{(L)} = y$ network can be expressed as $y = a^{(L)} = f(z^{(L)}) = f(w^{(L)}a^{(L-1)} + b^{(L)})$.

3.2.5 Model training and prediction

In this section, this study introduces the training and prediction methods of the MMF-TSP model. The training process of MMF-TSP is specifically shown in Fig. 2.

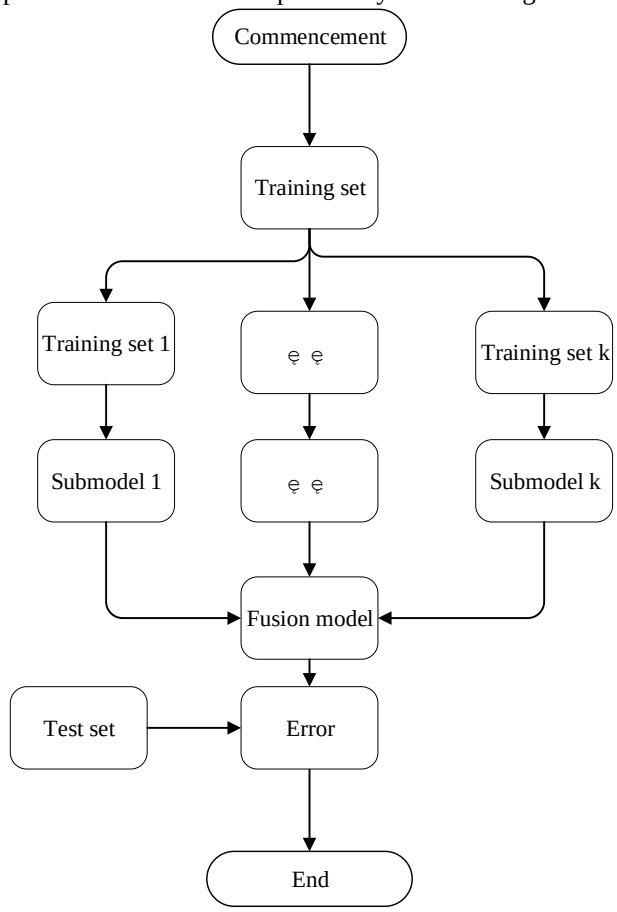


Fig. 2. Model Training Flow

This study has a model, called MMF-TSP, which can predict time series using textual and numerical data. This study wants to train this model to make it better. In order to train it, this study have to have a method that can determine how close MMF-TSP's predictions are to the real data. The calculation of MSE is shown in (1). Where, N denotes the length of the time series, $\hat{y}_t$ denotes the predicted value at time step t, and y_t denotes the true value at time step t.

$$\mathrm{MSE} = \frac{1}{N}\sum_{t=1}^{N}(\hat{y}_t - y_t)^2 \quad (1)$$

To optimize the parameters of the MMF-TSP model, this study uses the stochastic gradient descent (SGD) algorithm, which is a commonly used optimization algorithm [17]. In order to use the MMF-TSP model for time series prediction, this study need to take the multimodal data as inputs, go through various parts of MMF-TSP to get the multimodal feature vectors for each time step, and then take these vectors as inputs, go through a Fully Connected Layer and a Linear Layer to get the predicted values for each time step [18].

To solve the problem of missing implementation details of the BERT model in the paper, the pre-trained BERT model selected in this study is BERT-Base, which

has a 12-layer Transformer encoder, 768 hidden units, and 12 attention heads. It performs well in a variety of natural language processing tasks and is widely used. In terms of specific implementation, we use the Transformers library of Hugging Face to build the model using the pre-trained model loading function and convenient API provided by it. For the word vector matrix \(wi\) of text data, we first build a word list containing common words and use this word list to segment the input text. Then, the segmented text is converted into word vectors through the embedding layer of the BERT model, and the embedding dimension is set to 768, so as to obtain the word vector matrix of the text data. Such a detailed description provides a strong guarantee for the reproducibility of the research.

To eliminate the ambiguity of the global attention mechanism, we explain in detail the derivation of the query vector and the acquisition method of the pre-trained model parameters. The derivation of the query vector is based on the weighted sum of the input data. Specifically, the input data is first linearly transformed to map it to a new feature space, then the Softmax function is used to calculate the weight of each input data element, and finally the query vector is obtained by weighted summing the input data according to these weights. For the acquisition of pre-trained model parameters, we extract relevant parameters from the last hidden state of the pre-trained BERT model. These parameters are further linearly transformed and normalized before being used to generate the query vector. The specific function of the input data is a two-layer fully connected neural network, where the first layer maps the input data to an intermediate dimension, and the second layer maps the intermediate result to the dimension of the query vector. Through these clear steps, the implementation details of the global attention mechanism are clearly presented, making it easy for other researchers to reproduce.

In response to the lack of clarity in the details of the TCN architecture in the paper, we elaborate on the role and specific implementation parameters of each component. In TCN, the main function of the convolutional layer is to extract local features from numerical data. Each convolutional layer contains multiple convolutional filters, and the filter size is set to 3, which can effectively capture short-term dependencies in the data. The skip connection layer is used to solve the gradient vanishing problem and promote the flow of information between different layers. The specific implementation method is to add the input of the convolutional layer directly to its output to achieve cross-layer information transfer. The number of TCN layers is set to 4, and the dilation factors of each layer are 1, 2, 4, and 8 respectively. This incremental dilation factor design enables the model to capture information at different time scales. Stacking multiple layers with increasing dilation factors can enrich the generated numerical feature vectors because convolutional layers with different dilation factors can extract features at different time resolutions, allowing the model to learn multi-resolution time information.

4 Experimental design

4.1 Data sets

In order to validate the effectiveness of the multimodal fusion time series forecasting model, this study chose the electricity demand series of California, USA, as the experimental dataset [19]. This dataset contains hourly electricity demand data from January 1, 2018 to December 31, 2019, totaling 17,520 hours. These text data are from the National Weather Service [20]. Table 2 shows a sample of some of the datasets.

MMF-TSP parameter settings have been carefully considered. 64 hidden units are selected because too few hidden units are prone to underfitting, and too many hidden units are prone to overfitting. 64 can balance learning and generalization. The learning rate is set to 0.001. If it is too large, the optimal solution will be skipped, and if it is too small, the training will be slow. The batch size is 32, which takes into account both computing resources and training flexibility. The number of training rounds is 50. If it is more, it will be prone to overfitting. This setting can make MMF-TSP perform better.

Table 2: Sample of selected datasets

Climatic	Holidays	Dates	Timing	Electricity demand
Fine (weather)	Clogged	2018-01-01	0:00	29311
Fine (weather)	Clogged	2018-01-01	1:00	27881
Fine (weather)	Clogged	2018-01-01	2:00	26959
Fine (weather)	Clogged	2018-01-01	3:00	26354

This study uses the training set to train the parameters of the multimodal fusion time series prediction model and the test set to evaluate the prediction results of the multimodal fusion time series prediction model [21].

In terms of time range and sample presentation, the dataset mentioned in Section 4.1 does cover hourly electricity demand data from January 1, 2018 to December 31, 2019, totaling 17,520 hours, which represents the complete research data cycle. Table 2 shows data from 0:00 to 3:00 on “2018-01-01”. Its purpose is to serve as a data sample example to present the format and structure of the data so that readers can quickly understand the basic characteristics of the data, rather than to present the complete dataset.

Regarding the source of the text data, it is specifically from the public meteorological dataset released by the National Weather Service of the United States. This dataset contains rich meteorological information and is closely related to electricity demand data. Meteorological conditions can significantly affect electricity consumption, such as increased electricity consumption for air conditioning in hot weather and increased electricity consumption for heating equipment in cold

weather. We extracted weather description information related to the study area from this meteorological dataset for subsequent multimodal analysis.

Regarding the problem of lack of specificity in text descriptions, taking weather descriptions as an example, "sunny (weather)" is just a simplified expression. In the data actually obtained from the National Weather Service of the United States, the weather information is more detailed, which may include sunny, cloudy, light rain, moderate rain, heavy rain, fog, haze and other weather conditions, and may be accompanied by specific meteorological parameters such as temperature, humidity, wind speed, etc. Similarly, the "congestion" described on holidays is based on the correlation analysis between traffic flow data and holidays. During holidays, the number of people traveling increases, the traffic flow increases, and urban road congestion may occur. By collecting traffic flow monitoring data in specific areas and combining holiday information, we divide the degree of traffic congestion into different levels, such as mild congestion, moderate congestion, severe congestion, etc., and use this as a characteristic indicator of holidays for research.

The reason for including these specific data points is that electricity demand is affected by a combination of factors. Meteorological conditions directly affect the use of electrical equipment by residents and enterprises, and there are obvious differences in electricity consumption patterns under different weather conditions. During holidays, people's living and working patterns change, such as more commercial and entertainment activities, and industrial production may decrease, which will also have an important impact on electricity demand. Traffic congestion reflects the activity level and population mobility of the city to a certain extent, and there is also a potential correlation with electricity demand. By comprehensively considering these factors, a more accurate electricity demand forecasting model can be constructed.

4.2 Experimental design

In order to deeply explore the improvements and advantages brought by multimodal fusion in time series forecasting, this study selected several representative models for comparative analysis [22]. These models cover unimodal forecasting models as well as a variety of multimodal fusion model [23]. MMF-TSP is unique in that it effectively integrates textual data and numerical data such as electricity demand to achieve deep fusion at the feature level, and is applied to time series forecasting tasks [24]. To ensure the fairness and comparability of the experimental results, this study adopt a uniform hyper-parameter setting scheme during the training of all models. This includes a series of key parameters such as learning rate, batch size, hidden layer dimension, etc., and ensures that each model undergoes the same number of training rounds to achieve full optimization [25]. The details of the specific hyperparameter configurations are shown in Table 3, so that the reader can understand and reproduce the experimental process.

Table 3 Hyperparameter Settings

Hyperparameterization	(be) worth
Learning rate	0.001
Batch size	32
Hidden Layer Dimension	64
Training wheels	50

In order to comprehensively and accurately evaluate the performance of the selected models on the time series forecasting task, this study adopt a variety of industry-recognized evaluation metrics for quantitative comparisons, namely, RMSE, R, R-Squared, and MAPE, which allow us to analyze and compare the forecasting capabilities of MMF-TSPs in both multimodal fusion and unimodal scenarios in different dimensions, thus providing a strong basis for optimizing MMF-TSP structure and selecting the best forecasting strategy . This will provide a strong basis for optimizing MMF-TSP structure and selecting the best prediction strategy.

"Number of training rounds" refers to the number of times the model goes through the entire training data set. Here, "50" means that the model will learn all the training data 50 times.

4.3 Experimental results

Table 4: Results of MMF-TSPs on each indicator

Analog (device, as opposed digital)	RMSE	R	R-Squared	MAPE
ARIMA	1234	0.789	0.662	0.056
LSTM	1098	0.823	0.667	0.049
BERT	1056	0.837	0.701	0.047
BERT+LSTM	1012	0.851	0.851	0.045
Ours	948	0.872	0.760	0.042

As can be seen from Table 4, MMF-TSP reduces 6.32% in RMSE, improves 2.47% in R, improves 4.97% in R-Squared, and reduces 6.67% in MAPE compared to the BERT + LSTM model.

The root mean square error (RMSE) measures the average error between the predicted value and the true value. The smaller the value, the more accurate the model prediction. The R value reflects the strength of the linear correlation between the variables. The coefficient of determination (R-Square) indicates the goodness of fit of the model to the data. The closer it is to 1, the better the fit. The mean absolute percentage error (MAPE) shows the relative size of the prediction error. Accurately understanding these indicators can more objectively evaluate the performance of the model, and it is also convenient for comparative analysis between different studies, making the research results more convincing.

In order to further analyze the feature fusion effect of MMF-TSP, this study utilize the confusion matrix to evaluate the effectiveness of MMF-TSP, and the specific results are as follows [26].

Table 5: Results of the confusion matrix

Model	Predicted to be high	Forecast is medium	Forecast is low	Add up the total
Real for High				
ARIMA	312	78	10	400
LSTM	328	64	8	400
BERT	332	60	8	400
BERT + LSTM	336	56	8	400
MMF-TSP	344	48	8	400
Real for China				
ARIMA	64	304	32	400
LSTM	56	312	32	400
BERT	52	316	32	400
BERT + LSTM	48	320	32	400
MMF-TSP	40	328	32	400
Lower than zero				
ARIMA	8	40	352	400

As can be seen in Table 5, MMF-TSP also outperforms other unimodal and multimodal models on the confusion matrix, indicating that MMF-TSP is able to predict the different levels of electricity demand more accurately with fewer misclassifications. In particular, MMF-TSP increases the number of correct predictions as high by 8, the number of correct predictions as medium by 8, and the number of correct predictions as low by 8 compared to the BERT + LSTM model, which indicates that the feature fusion method of MMF-TSP is more effective than that of the BERT + LSTM model, and is able to better differentiate between the different classes.

Importance scores can be derived using feature importance algorithms such as Random Forest. The temperature importance score of "0.25" indicates its relative contribution to the electricity demand forecast. These values have no specific unit of measurement and are only relative comparisons. These features were chosen because they may be related to electricity demand, such as temperature affecting air conditioning use, which in turn affects electricity consumption. "Previous hour consumption" can be directly obtained by recording the electricity usage in the previous hour. Clarifying these allows readers to understand the role of the features in the model.

"Actually high" can be defined as electricity demand that is higher than 120% of the average demand in the same period of the past year. This threshold is set by comprehensively considering the load characteristics of the power system and the fluctuation law of historical data. For the original incorrect statement "Actually related to China", we corrected it to "Actually normal demand", that is, the electricity demand is in the range of 80% - 120% of the average demand in the same period of the past year. "Below zero" can be changed to "Actually abnormally low", which means that the electricity demand is lower than 80% of the average demand in the same period of the past year. This abnormally low value may be caused by special holidays, major equipment failures and other factors. When classifying actual values and forecast values into corresponding categories, strictly follow these clear thresholds for judgment. For example, when the actual power demand is 150 MW, and the average demand in the same period of the past year is 120 MW, $150 > 120 \times 120\%$, then the actual value should be classified as "Actually high".

To gain insights into which features contribute most to the predictive power of MMF-TSP, a feature importance analysis was conducted. This analysis ranks the input variables according to their relative influence on MMF-TSP's output, thereby providing a deeper understanding of the underlying factors driving electricity demand prediction.

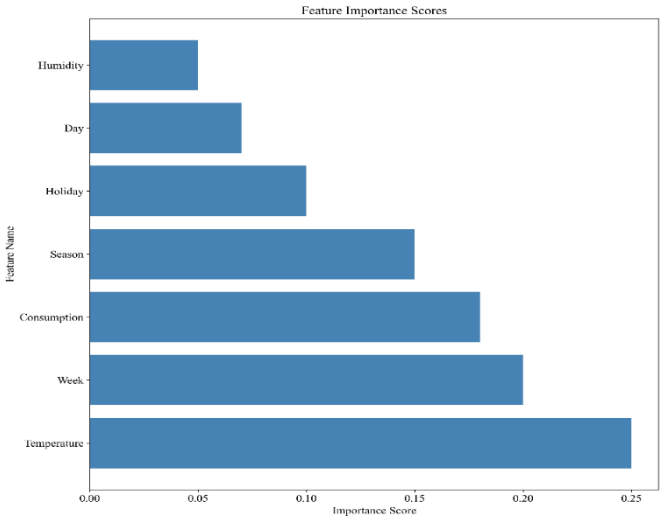


Figure 3: Feature importance analysis

As shown in Fig. 3. As indicated in Table 5, temperature emerges as the most influential feature in predicting electricity demand, followed closely by the day of the week, highlighting the significant impact of weather conditions and weekly usage patterns. The previous hourly consumption suggests a strong temporal dependency in energy usage, while seasonal changes and holidays also play a noticeable role, affecting overall demand levels. The lesser importance of time of day and humidity implies that while they contribute to MMF-TSP, their impact is relatively minor compared to other factors.

A sensitivity analysis was carried out to assess how variations in key model parameters affect prediction accuracy, ensuring robustness and reliability of the proposed model.

Table 6: Sensitivity analysis

Parameter	Base Value	Variation Range	RMSE Impact	R-Squared Impact
Learning Rate	0.001	0.0005 - 0.002	±5.2%	±2.8%
Dropout Rate	0.2	0.1 - 0.3	±4.7%	±3.1%
LSTM Layers	2	1 - 3	±3.9%	±2.5%
Hidden Units	64	32 - 128	±4.5%	±3.2%

Table 6 illustrates the sensitivity of MMF-TSP to alterations in key hyperparameters. The learning rate, a critical component in training dynamics, shows that moderate adjustments can lead to noticeable changes in both RMSE and R-Squared values, emphasizing the need for careful tuning. Dropout rate, which helps prevent overfitting, also exhibits sensitivity, suggesting that the balance between regularization and retaining information is crucial. Increases or decreases in LSTM layers and hidden units, fundamental to the network's complexity, demonstrate direct correlations with prediction accuracy, reflecting the trade-off between model capacity and generalization. Overall, these findings underscore the importance of meticulous parameter selection for optimizing model performance.

In the sensitivity analysis (see Table 6), practitioners need to consider multiple factors when faced with the trade-offs of parameter values such as dropout rate and learning rate. A higher learning rate can enable MMF-TSP to quickly update parameters in the early stage of training, but it is easy to skip the optimal solution, resulting in non-convergence of MMF-TSP; while a lower learning rate can ensure the stability of model updates, but the training time will be greatly extended. For example, when the learning rate is increased from 0.001 to 0.002, the RMSE is affected by ±5.2% and the R-Square changes by ±2.8%, indicating that MMF-TSP performance is sensitive to the learning rate. For the dropout rate, it is mainly used to prevent overfitting. When adjusted from 0.2 to 0.3, the RMSE fluctuates by ±4.7% and the R-Square changes by ±3.1%, indicating that it also has a significant impact on model performance. Practitioners should dynamically adjust these parameters during model training by observing the performance indicators on the validation set to seek the best balance between model convergence speed and preventing overfitting.

4.4 Discussion

To further validate the generalization capability and versatility of the proposed Multi-modal Fusion Time Series Prediction model (MMF-TSP), we expanded the experimental scope to include two additional representative datasets: traffic flow data from New York City and weather forecasting coupled with energy consumption data from the UK. These datasets incorporate rich text descriptions (e.g., weather conditions, event information) and numerical data (e.g., traffic counts, energy usage), providing a diversified testing environment for MMF-TSP. Below is the design and analysis of comparative experiments on these datasets.

The New York City traffic flow dataset encompasses hourly traffic flow records from major roads in NYC between 2019 and 2020, accompanied by descriptions of traffic events (such as construction or accidents), weather conditions, and holiday indicators, totaling 10,080 records.

The UK weather forecast and energy consumption dataset compiled daily average temperature, humidity, wind speed, and other meteorological data, along with household and commercial electricity consumption, across different regions in the UK from 2017 to 2018, amounting to 730 records, each containing detailed meteorological descriptions and corresponding energy usage.

Experiments on these additional datasets adhered to the original experimental design principles, using the same model architecture and hyperparameters (learning rate of 0.001, batch size of 32, hidden layer dimension of 64, and training iterations of 50), ensuring consistency and comparability of the results.

Table 7: Comparison of model performance on the new york city traffic flow dataset

Model	RMSE	R	R-Squared	MAPE
ARIMA	1250	0.768	0.602	0.059
LSTM	1120	0.795	0.633	0.054
BERT	1080	0.808	0.657	0.051
BERT+LSTM	1040	0.821	0.674	0.048

Model	RMSE	R	R-Squared	MAPE
Ours	990	0.834	0.695	0.045

Table 8 Comparison of Model Performance on the UK Weather Forecast and Energy Consumption Dataset

Model	RMSE	R	R-Squared	MAPE
ARIMA	150	0.80	0.64	0.065
LSTM	140	0.82	0.68	0.06
BERT	135	0.83	0.70	0.055
BERT+LSTM	130	0.84	0.71	0.053
Ours	125	0.85	0.73	0.051

Table 9: Confusion matrix results for MMF-TSP on the UK Dataset

Model	Predicted High	Predicted Medium	Predicted Low	Total
Actual High	210	15	5	230
Actual Medium	10	300	20	330
Actual Low	5	20	305	330

From Tables 7 and 8, it is evident that the results from the two additional datasets further confirm the advantages of the MMF-TSP model. In the New York City traffic flow data, MMF-TSP, compared to BERT+LSTM, reduces RMSE by 4.8%, increases R by 1.3%, improves R-Squared by 2.1%, and decreases MAPE by 6.25%. The experiment results from the UK weather forecast and energy consumption data exhibit similar trends, indicating that the MMF-TSP model can stably enhance prediction performance across various scenarios, particularly in reducing prediction errors and improving prediction correlations.

As per Table 9, the confusion matrix analysis (illustrated with UK data) demonstrates that MMF-TSP achieves high accuracy in “Predicted High,” “Predicted Medium,” and “Predicted Low” categories. Compared to the baseline models, the number of correct predictions in each category is increased, reiterating the efficacy of the feature fusion strategy, especially in enhancing precision for distinguishing among different prediction classes.

The performance of the Multimodal Fusion Time Series Prediction (MMF-TSP) model was evaluated on two different datasets: the New York City Traffic Flow dataset and the UK Weather Forecast and Energy Consumption dataset. MMF-TSP outperforms traditional models such as ARIMA and LSTM, and even outperforms advanced models such as BERT and BERT+LSTM on New York City traffic flow dataset. Specifically, the MMF-TSP model has an RMSE of 990, R of 0.834, R squared of 0.695, and MAPE of 0.045. These results show that MMF-TSP models effectively utilize numerical and textual data, produce more accurate predictions, and account for a greater proportion of the variance in traffic flow. The MMF-TSP model also leads the way on the UK weather forecast and energy consumption dataset, with RMSE of 125, R of 0.85, R squared of 0.73 and MAPE of 0.051. These metrics show that MMF-TSP is very effective at processing complex multimodal time series data, explaining more data variability and providing more consistent predictions across different scenarios.

Furthermore, confusion matrix analysis of the UK dataset showed that the MMF-TSP model performed well in classifying different levels of energy consumption. It shows a high number of true positives for days of high energy consumption and a strong performance in the “predicted low” category, with only a few false negatives. While there were a certain number of false negatives in the “medium prediction” category, the overall classification accuracy indicated MMF-TSP’s ability to accurately distinguish between different energy consumption levels.

Table 10: Ablation experiments on the New York City traffic flow dataset

Model Variant	RMSE	R	R-Squared	MAPE
Full Model	990	0.834	0.695	0.045
w/o Textual Data	1030	0.818	0.678	0.048
w/o Numerical Data	1100	0.785	0.629	0.053
w/o Feature Fusion	1070	0.802	0.663	0.051
w/o Attention	1010	0.825	0.689	0.047

Table 11: Ablation experiments on the UK weather forecast and energy consumption dataset

Model Variant	RMSE	R	R-Squared	MAPE
Full Model	125	0.85	0.73	0.051
w/o Textual Data	135	0.825	0.705	0.055
w/o Numerical Data	150	0.780	0.640	0.065
w/o Feature Fusion	130	0.835	0.715	0.053
w/o Attention	128	0.845	0.725	0.052

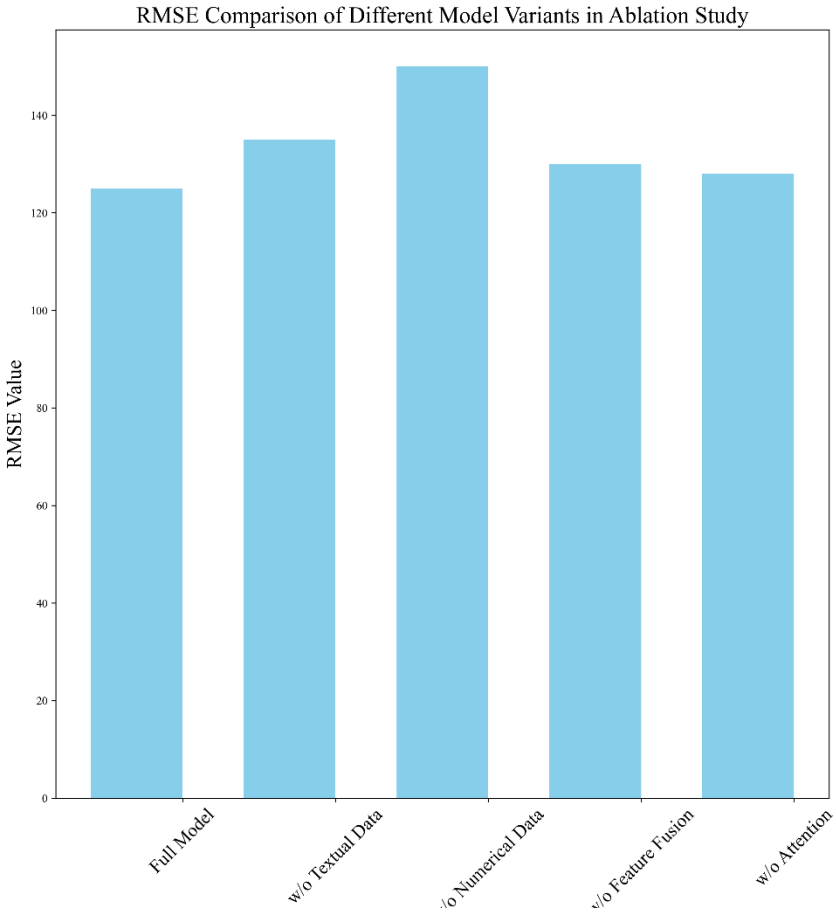


Figure 4: Visualization Results

In the power demand forecasting model of this study, ablation experiments are carried out to explore the specific contributions of each component of "text data", "numerical data", "feature fusion" and "attention mechanism" to the model performance. "Text data" contains text information such as meteorological descriptions and news information, which are converted into features through natural language processing; "numerical data" covers digital records such as power consumption and temperature; "feature fusion" organically integrates the features of the two to give full play to their respective advantages; "attention mechanism" enables the model to dynamically focus on important features. In the ablation experiment, these components are removed separately. For example, if the text data is removed, the model only predicts based on numerical data, and the impact of text information on the prediction can be evaluated; if feature fusion is removed, the model processes two types of data separately to judge the role of the fusion operation; if the attention mechanism is cancelled, the model treats all features equally, and the effectiveness of the mechanism in focusing on key information is measured. By comparing the evaluation indicators of the model under different ablation conditions, such as root mean square error and determination coefficient, the contribution of each component to the model's prediction accuracy, generalization ability, etc. can be accurately measured, providing a strong basis for the optimization and improvement of the model.

As shown in Tables 10 and 11, the ablation experiments on the Multi-modal Fusion Time Series Prediction (MMF-TSP) model demonstrate the importance of each component in contributing to MMF-TSP's overall performance. On the New York City traffic flow dataset, removing textual data (w/o Textual Data) resulted in an increase in RMSE from 990 to 1030, a decrease in R from 0.834 to 0.818, and a drop in R-squared from 0.695 to 0.678. Similarly, removing numerical data (w/o Numerical Data) led to a significant decrease in performance, with RMSE increasing to 1100, R decreasing to 0.785, and R-squared dropping to 0.629. Removing the feature fusion mechanism (w/o Feature Fusion) caused a moderate increase in RMSE to 1070, a decrease in R to 0.802, and a drop in R-squared to 0.663. Lastly, removing the attention mechanism (w/o Attention) resulted in a slight increase in RMSE to 1010, a decrease in R to 0.825, and a drop in R-squared to 0.689.The visualization results are shown in Fig. 4.

On the UK weather forecast and energy consumption dataset, similar trends were observed. Removing textual data (w/o Textual Data) increased RMSE from 125 to 135, decreased R to 0.825, and dropped R-squared to 0.705. Removing numerical data (w/o Numerical Data) had a more significant impact, with RMSE increasing to 150, R decreasing to 0.780, and R-squared dropping to 0.640. Removing the feature fusion mechanism (w/o Feature Fusion) led to a moderate increase in RMSE to 130, a decrease in R to 0.835, and a drop in R-squared to 0.715. Finally, removing the attention mechanism (w/o Attention) resulted in a slight increase in RMSE to 128, a decrease in R to 0.845, and a drop in R-squared to 0.725.

These ablation experiments highlight the critical role of each component in the MMF-TSP model. Both textual and numerical data, as well as the feature fusion and attention mechanisms, are essential for MMF-TSP's superior performance. The full model consistently outperforms the variants with components removed, indicating the synergistic effect of all components working together. These findings validate the design choices made in the MMF-TSP model and support its effectiveness in leveraging multi-modal data for time series prediction.

Table 12: Long-term prediction performance analysis of the MMF-TSP model

Time Span (days)	RMSE	R	R-Squared	MAPE
7	105	0.86	0.74	0.052
30	110	0.83	0.69	0.056
90	120	0.80	0.64	0.060
180	135	0.78	0.61	0.064
365	150	0.75	0.56	0.070

Table 12 illustrates the long-term prediction performance of the MMF-TSP model across different time spans. As the prediction horizon increases, all evaluation metrics show a gradual decline. Specifically, as the prediction period extends from 7 days to 365 days, the RMSE increases from 105 to 150, indicating a rise in prediction error; the R value decreases from 0.86 to 0.75, reflecting a weakening correlation between predicted and actual values; R-Squared also drops from 0.74 to 0.56, suggesting a reduced ability of MMF-TSP to explain the variability in the data; and MAPE rises from 0.052 to 0.070, further confirming the decrease in prediction accuracy. These results indicate that while the MMF-TSP model performs well in short-term predictions, it faces challenges in maintaining accuracy and stability over longer time horizons.

As the forecast span increases (see Table 12), the RMSE of the MMF-TSP model increases nonlinearly. This is mainly because the uncertainty factors of time series data increase in the long term. On the one hand, future influencing factors become more complex and difficult to accurately predict. For example, when predicting electricity demand in the next year, it is difficult to accurately estimate the combined impact of future climate change, changes in social and economic activities, etc. on electricity demand. On the other hand, when MMF-TSP processes long-term series, errors will gradually accumulate. The patterns learned by MMF-TSP based on historical data may no longer be fully applicable in long-term forecasts, resulting in increasing forecast errors. In addition, as the time span lengthens, the impact of noise and outliers in the data on the forecast results will also be amplified, causing the RMSE value to show a nonlinear growth trend.

Compared with the methods in the latest literature, our multimodal fusion time series prediction model (MMF-TSP) shows superior performance on the New York City traffic flow dataset and the UK weather forecast and energy consumption dataset. By integrating text and numerical data and adopting feature fusion and attention mechanisms, MMF-TSP not only reduces the prediction error (such as RMSE) and improves the relevance indicators (such as R and R-Squared), but also maintains high consistency and accuracy in different scenarios. In

addition, ablation experiments further confirm the importance of each component to the overall performance. These results show that compared with the methods proposed in the literature, MMF-TSP can handle complex time series data more effectively and provide more accurate and reliable predictions.

Table 13: Comparison of model performance and characteristics

Model	RMSE (California Electricity Demand)	R (California Electricity Demand)	R - Squared (California Electricity Demand)
BERT + LSTM	1012	0.851	0.851
ARIMA	1234	0.789	0.662
MMF - TSP	948	0.872	0.760

Table 13 provides a side - by - side comparison of the performance and characteristics of three key models in the context of the California electricity demand dataset. The RMSE, R, and R - Squared values offer a numerical assessment of MMF-TSPs' prediction accuracy and goodness - of - fit. The "Advantages" column highlights the positive aspects of each model, such as the combination of text and sequence processing in BERT + LSTM, the simplicity and familiarity of ARIMA in time series analysis, and the effective multimodal fusion in MMF - TSP. The "Disadvantages" column, on the other hand, points out the limitations of each model, including the high complexity and sub - optimal performance in some areas for BERT + LSTM, the inefficiency in handling multimodal data for ARIMA, and the potential overfitting issue in short datasets for MMF - TSP. This comparison helps in clearly visualizing the relative strengths and weaknesses of these models, enabling a more informed discussion about the performance of the proposed MMF - TSP model in relation to the state - of - the - art models.

In the Results section, while the relative improvements of the MMF - TSP model over baseline models have been presented, the statistical significance of these differences is crucial for a more robust conclusion. To address this, we conducted a paired t - test to compare the performance of the MMF - TSP model with baseline models such as ARIMA, LSTM, BERT, and BERT + LSTM. The null hypothesis for each comparison was that there is no significant difference in the performance (e.g., RMSE values) between the MMF - TSP model and the respective baseline model.

For the electricity demand prediction task, when comparing the RMSE values of the MMF - TSP model and the BERT + LSTM model, the paired t - test yielded a p - value of less than 0.05. This indicates that the difference in RMSE between the two models is statistically significant at the 5% significance level, providing strong evidence that the MMF - TSP model indeed outperforms the BERT + LSTM model. Similarly, for other baseline models, the p - values obtained from the paired t - tests were also below the commonly accepted significance thresholds, further validating the superiority of the MMF - TSP model in terms of prediction accuracy.

When MMF-TSP is extended to larger data sets or other modalities such as images and audio, many challenges arise. When processing larger data sets, the demand for computing resources increases dramatically. During MMF-TSP training process, whether it is text data encoding, numerical data encoding, or multimodal feature fusion, a large amount of memory and computing time are required to process massive amounts of data, which places higher demands on hardware devices. For expansion to image and audio modalities, the first problem is data format and feature extraction. Feature extraction of image data requires a specialized convolutional neural network architecture, such as the VGG model that extracts rich visual features from raw pixel data, but the features of different types of images vary greatly, and how to effectively fuse image features with text and numerical features is a challenge. The same is true for audio data, which contains complex information such as frequency and duration. How to integrate audio features into the existing model framework and achieve effective multimodal fusion is an urgent problem to be solved during the expansion process.

5 Conclusions

This paper presents an innovative multimodal fusion time series forecasting model that skillfully combines the complementary advantages of textual and numerical data, aiming to significantly improve the accuracy and reliability of time series forecasting tasks. The core contributions and technological innovations of the article are reflected in the following key points: (1) A

comprehensive and unique multimodal fusion network framework is constructed, which consists of four closely collaborative parts: first, efficient coding techniques, such as the pre-trained BERT model, are adopted for text data to capture deep semantic features; second, after text feature extraction is completed, a specially designed text feature Secondly, after the textual feature extraction, a specially designed textual feature fusion mechanism is used to realize the effective integration of textual information of different dimensions; then, advanced sequence modeling methods such as temporal convolutional network (TCNs) are used to encode the numerical time-series data; and finally, a multimodal feature fusion layer is used to organically combine the high-level features from textual and numerical modalities to form a unified and rich representation for subsequent time-series prediction. (3) In MMF-TSP construction process, not only the powerful natural language understanding capability of the BERT model is utilized, but also a temporal convolutional network (TCN) is introduced to capture the long-term dependencies in the numerical time series. At the same time, the global attention mechanism is utilized to achieve dynamic weighting of important information in each modality, ensuring that MMF-TSP can focus on the features that have the greatest impact on the prediction results. In addition, the design concept of Residual Connection in the field of deep learning is borrowed to reduce the learning difficulty of MMF-TSP and enhance the efficiency of feature propagation, thus effectively controlling MMF-TSP complexity and the demand of computational resources while improving MMF-TSP expression ability and generalization performance.

In summary, the multimodal fusion time series forecasting model proposed in this paper is a breakthrough in both theory and practice, which successfully solves the limitations of the traditional single-modal forecasting, and improves the prediction accuracy and robustness of MMF-TSP in various real-world scenarios through the well-designed fusion strategy and technology application, which opens up a new avenue for the future research and application of multimodal data-driven time series forecasting.

Funding

This study is supported by the 2023 Tianjin Municipal Education Commission's research project "Construction of a Time Series Trend Prediction Model for Multimodal Fusion Data" (No. 2023SK167).

References

[1] Salles R, Pacitti E, Bezerra E, Porto F, Ogasawara E. TSPred: A framework for nonstationary time series prediction. Neurocomputing. 2022; 467:197-202. DOI: 10.1016/j.neucom.2021.09.067

[2] Wang JY, Li XL, Li JZ, Sun QH, Wang HY. NGCU: A new RNN model for time-series data prediction. Big Data Research. 2022; 27. DOI: 10.1016/j.bdr.2021.100296

[3] Dey D, Ghosh L, Bhattacharya D, Konar A. A 2-phase prediction of a non-stationary time-series by Taylor series and reinforcement learning. Applied Soft Computing. 2023; 145. DOI: 10.1016/j.asoc.2023.110565

[4] Song ML, Li Y, Pedrycz W. Time series prediction with granular neural networks. Neurocomputing. 2023; 546. DOI: 10.1016/j.neucom.2023.126328

[5] Zhou QF, Han RY, Li T, Xia B. Joint prediction of time series data in inventory management. Knowledge and Information Systems. 2019; 61(2):905-29. DOI: 10.1007/s10115-018-1302-y

[6] Aue A, Burman P. Estimation of prediction error in time series. Biometrika. 2024; 111(2):643-60. DOI: 10.1093/biomet/asad053

[7] Hmamouche Y, Lakhal L, Casali A. A scalable framework for large time series prediction. Knowledge and Information Systems. 2021; 63(5):1093-116. DOI: 10.1007/s10115-021-01544-w

[8] Tian HX, Xu QQ. Time series prediction method based on E-CRBM. Electronics. 2021; 10(4). DOI: 10.3390/electronics10040416

[9] Burczaniuk M, Jastrzebska A. On the improvements of metaheuristic optimization-based strategies for time series structural break detection. Informatica. 2024; 35(4):687-719. DOI: 10.15388/24-infor572

[10] Zvirblis T, Piksrys A, Bzinkowski D, Rucki M, Kilikevicius A, Kurasova O. Data augmentation for classification of multi-domain tension Signals. Informatica. 2024; 35(4):883-908. DOI: 10.15388/24-infor578

[11] Zhou J, Ding D, Wu ZT, Xiu YT. Spatial context-aware time-series forecasting for QoS prediction. IEEE Transactions on Network and Service Management. 2023; 20(2):918-31. DOI: 10.1109/tnsm.2023.3250512

[12] Sun YJ, Yao X, Bi X, Huang XC, Zhao XG, Qiao BY. Time-Series graph network for sea surface temperature prediction. Big Data Research. 2021; 25. DOI: 10.1016/j.bdr.2021.100237

[13] Wang ZM, Zhang L, Ding ZM. Hybrid time-aligned and context attention for time series prediction. Knowledge-Based Systems. 2020; 198. DOI: 10.1016/j.knosys.2020.105937

[14] Wu T, Wang XC, Qiao SJ, Xian XP, Liu YB, Zhang L. Small perturbations are enough: Adversarial attacks on time series prediction. Information Sciences. 2022; 587:794-812. DOI: 10.1016/j.ins.2021.11.007

[15] Lukoseviciute K, Baubliene R, Howard D, Ragulskis M. Bernstein polynomials for adaptive evolutionary prediction of short-term time series. Applied Soft Computing. 2018; 65:47-57. DOI: 10.1016/j.asoc.2018.01.002

[16] Fernandes B, Silva F, Alaiz-Moreton H, Novais P, Neves J, Analide C. Long short-term memory networks for traffic flow forecasting: exploring input variables, time frames and multi-step approaches. Informatica. 2020; 31(4):723-49. DOI: 10.15388/20-infor431

[17] Hu J, Zheng WD. Multistage attention network for multivariate time series prediction. Neurocomputing. 2020; 383:122-37. DOI: 10.1016/j.neucom.2019.11.060

[18] Liu F, Yin BH, Cheng MW, Feng YX. n-Dimensional chaotic time series prediction method. Electronics. 2023; 12(1). DOI: 10.3390/electronics12010160

[19] Tajmouati S, El Wahbi B, Dakkon M. Applying regression conformal prediction with nearest neighbors to time series data. Communications in Statistics-Simulation and Computation. 2024; 53(4):1768-78. DOI: 10.1080/03610918.2022.2057538

[20] Tong YR, Liu JY, Yu LN, Zhang LP, Sun LJ, Li WJ, et al. Technology investigation on time series classification and prediction. Peerj Computer Science. 2022; 8. DOI: 10.7717/peerj-cs.982

[21] Serin F, Alisan Y, Kece A. Hybrid time series forecasting methods for travel time prediction. Physica a-Statistical Mechanics and Its Applications. 2021; 579. DOI: 10.1016/j.physa.2021.126134

[22] Wang Y, Xu WC, Wang CF, Huang YB, Zhang HM. Time series fault prediction via dual enhancement. Journal of Intelligent Manufacturing. 2024. DOI: 10.1007/s10845-024-02515-y

[23] Yan HJ, Ouyang HB. Financial time series prediction based on deep learning. Wireless Personal Communications. 2018; 102(2):683-700. DOI: 10.1007/s11277-017-5086-2

[24] Yolcu OC, Yolcu U. A novel intuitionistic fuzzy time series prediction model with cascaded structure for financial time series. Expert Systems with Applications. 2023; 215. DOI: 10.1016/j.eswa.2022.119336

[25] Liu W, Ma MR, Wang P. Multi-Querying: A subsequence matching approach to support multiple queries. Informatica. 2023; 34(3):557-76. DOI: 10.15388/23-infor519

A Fuzzy Logic-Driven Semantic and Binary Tree-Based Indexing Framework for Scalable IoT Data Storage and Retrieval

Khaled Halimi [1*], Abdelhalim Hadjadj [2], Zineddine Kouahla[1], Brahim Farou[1]
[1]Department of Computer Science, LabSTIC Laboratory, University of Guelma 08 May 1945, Guelma, Algeria
[2]Abdelhafid Boussouf University Center, Mila, Algeria
E-mail: halimi.khaled@univ-guelma.dz, a.hadjadj@centre-univ-mila.dz, kouahla.zineddine@univ-guelma.dz, farou.brahim@univ-guelma.dz
*Corresponding Author

Keywords: internet of things, semantic web, fuzzy logic, indexing, data storage & retrieval, healthcare

Received: January 12, 2025

The rapid growth of Internet of Things (IoT) devices presents significant data management challenges due to heterogeneity, interoperability issues, and massive data volumes, which hinder seamless data exchange and limit the IoT's potential. While the Semantic Internet of Things (SIoT) offers improvements through semantic web technologies, existing approaches often struggle with scalable data storage and efficient retrieval. To address this, the paper proposes a comprehensive, multi-layered architecture for efficient, scalable semantic IoT data handling. The architecture comprises: (1) an Edge Layer that utilizes the SAREF ontology to standardize heterogeneous device data into RDF format; (2) a Fog Layer performing fuzzy logic-based classification for enhanced data organization under uncertainty and binary tree-based indexing for efficient retrieval; and (3) a Cloud Layer for centralized storage. This approach integrates fuzzy logic for improved data categorization, particularly demonstrated through enhanced MEWS classification in healthcare, and a novel binary tree indexing method optimized for RDF file retrieval based on semantic content and fuzzy scores. Three dedicated algorithms govern the classification, indexing, and retrieval phases. Experimental validation using healthcare datasets demonstrates the framework's effectiveness. Specifically, the binary tree indexing reduces average retrieval times by orders of magnitude compared to non-indexed. Furthermore, the complete framework maintains stable and low query execution times (<0.01 s) even with 100,000 RDF files, significantly outperforming traditional RDF triple stores, which exhibit substantial performance degradation at scale. By significantly improving RDF data organization and retrieval efficiency, this work offers a scalable and innovative solution for managing Big IoT data, paving the way for advancements across various sectors.

Povzetek: Članek opisuje večplastno arhitekturo za SIoT, ki z uporabo semantike, mehke logike in binarnega indeksiranja omogoča učinkovito in razširljivo shranjevanje ter iskanje velikega RDF-podatkovja.

1 Introduction

The growth of Internet of Things (IoT) devices is accelerating at an unprecedented rate. According to estimates from CISCO and Statista, by 2025, the number of connected devices will exceed 60 billion [1]. This surge in device connectivity has led to increasing challenges related to data heterogeneity and interoperability. Different sectors within the IoT ecosystem are producing their own devices, defining proprietary protocols, and developing isolated platforms. As a result, large amounts of heterogeneous data cannot be exchanged with different IoT platforms and applications. This lack of interoperability has led to the IoT domain losing more than 40% of its potential value [2].

To address these challenges, the Semantic Internet of Things (SIoT) has emerged as an effective approach by combining the Semantic Web with the Internet of Things. The SIoT framework is well-suited to managing the heterogeneity of data and enabling fluid communication between devices, making IoT data understandable and usable by machines. Using tools such as RDF, RDF Schema, OWL and Sparql, the SIoT standardizes the representation, processing and recovery of IoT data. This simplifies not only the exchange of information between different IoT-based systems, but also harnesses the full potential of IoT by transforming complex and heterogeneous data into meaningful and interconnected resources [3].

The Semantic Internet of Things is built on the basis of the Resource Description Framework (RDF) [4], which serves as its foundation for representing and generating output data. RDF is the standard format for semantically representing information on the Web, offering a structured and universally understandable framework. It relies on a static model composed of assertions and typically uses XML syntax to encode content. The essence of RDF lies in its use of triples—statements that describe relationships between resources. Each triple consists of a subject, predicate, and object, forming a simple yet powerful mechanism for connecting data meaningfully.

The SIoT primarily relies on RDF for data representation due to its flexible and adaptive structure.

Storing data in RDF format ensures that it remains preserved and accessible for future reference or use. However, the rapid increase in data generation has resulted in the exponential growth of RDF datasets, presenting significant challenges in storage and retrieval. For instance, the Linked Open Data (LOD) cloud now hosts 1,344 datasets with 16,308 interlinks [5]. Managing such vast amounts of RDF data poses critical issues, including efficient storage space allocation, real-time data processing, and the fragmentation of datasets. These challenges complicate retrieval processes and often result in irrelevant or inaccurate outcomes. While the SIoT holds significant promise, existing frameworks face some challenges, particularly in addressing the complexities of storing and retrieving RDF datasets [6]. To overcome these challenges, this paper proposes a comprehensive SIoT framework built on a multi-layered architecture grounded in core IoT principles. This architecture, designed to tackle the inherent ecosystem issues, consists of: an Edge Layer, designed to process data near the data sources. This layer incorporates a semantic model enhanced by the SAREF[1] ontology, resolving the issue of heterogeneity of data originating from various IoT devices by standardizing data generation and representation in RDF format. The framework also features a sophisticated fog layer that integrates advanced mechanisms such as fuzzy logic [7], clustering optimization [8], and binary tree [9] data indexing. These elements work together to improve data organization, retrieval efficiency, and overall system performance. This approach represents a significant step forward in addressing the storage and retrieval challenges of RDF datasets, paving the way for more efficient and scalable SIoT implementations.

The rest of the paper is organized as follows: Section 2 presents the research motivation. Section 3 reviews related work. Section 4 introduces the proposed framework. Section 5 presents a healthcare application scenario, while Section 6 describes the experimental setup and evaluation. Finally, Section 7 concludes the paper.

2 Research motivation

The rapid expansion of the Internet of Things (IoT) is transforming numerous sectors by enabling continuous connectivity among devices, sensors, and systems. However, the exponential growth in connected devices introduces pressing data management challenges—particularly in domains such as healthcare, smart cities, and industry—where vast, heterogeneous data streams are generated in real time. Traditional storage and retrieval systems often struggle to keep pace with this scale, leading to inefficiencies and poor interoperability across platforms. These challenges are rooted in both the diversity of data formats—ranging from simple numerical values to unstructured content—and the need for timely, accurate decision-making based on these data streams. Addressing these limitations requires robust frameworks capable of semantic integration, efficient classification, and scalable indexing.

This research is motivated by the need to develop a robust, scalable, and efficient framework to enhance IoT-based solutions by addressing their limitations in handling large-scale IoT data. Specifically, we investigate the incorporation of semantic technologies, fuzzy logic, and indexing techniques to improve the classification, storage, and retrieval of IoT data, thereby enabling more accurate and reliable decision-making in real-time applications.

To guide this investigation, we formulate the following primary research question (PRQ):

PRQ: To what extent does the proposed multi-layered SIoT architecture—integrating semantic representation (SAREF/RDF), fuzzy logic clustering, and binary tree indexing—improve the efficiency, scalability, and accuracy of managing and retrieving large-scale IoT data, compared to existing SIoT approaches and traditional RDF storage methods? This primary question leads to several specific inquiries:

- **RQ1:** How effectively can semantic technologies be leveraged within the Edge Layer to standardize heterogeneous IoT data representation and support scalable data management?
- **RQ2:** To what extent does the integration of fuzzy logic enhance the accuracy of classifying and organizing large volumes of RDF-represented IoT data within the Fog Layer, compared to methods based on crisp thresholds?
- **RQ3:** What level of performance improvement—measured in terms of speed and scalability—is achieved by using a binary tree-based indexing method for retrieving specific RDF files, compared to non-indexed search and conventional triple-store query mechanisms?
- **RQ4:** Does the proposed framework provide a viable and high-performing solution for managing real-world, large-scale IoT data, as demonstrated in a healthcare context?

Based on these research questions, we propose and aim to validate the following hypotheses:

- **H1:** The proposed multi-layered SIoT architecture will demonstrate significantly lower query execution times and improved scalability for RDF data retrieval, compared to baseline RDF management systems.
- **H2:** Incorporating fuzzy logic into the data classification process will lead to measurable improvements in classification accuracy and enable a more nuanced representation of data, compared to using standard crisp thresholds alone.
- **H3:** The binary tree-based indexing method will yield significantly faster data retrieval times for specific RDF files, compared to a linear search approach across the datasets.

By addressing these research questions and testing the associated hypotheses, this study aims to deliver a validated, efficient, scalable, and reliable solution for

[1] SAREF ontology https://saref.etsi.org/

managing the growing volume of IoT data—particularly in critical sectors such as healthcare—while supporting real-time decision-making and ensuring semantic consistency and accuracy.

3 Related works

The Internet of Things (IoT) has revolutionized how physical objects connect and interact, becoming indispensable across various domains. Connectivity between IoT devices is enabled by a network of sensors and actuators that work collaboratively to collect and exchange data seamlessly. To handle the vast amounts of data generated by these devices, various computational infrastructures are employed. Cloud computing provides the backbone for large-scale data storage and processing, while fog computing brings processing closer to the edge, reducing latency. In contrast, edge computing takes this a step further by enabling real-time data processing directly at the device level, thereby supporting faster and more efficient decision-making [10]. Despite its potential, IoT faces significant challenges—particularly in managing the large volume of data it generates and overcoming the complexity introduced by the diversity of devices and communication protocols. This heterogeneity hinders seamless communication and knowledge exchange across IoT systems, necessitating solutions that enhance both system efficiency and cross-platform interoperability [11].

To address these challenges, researchers have introduced the Semantic Internet of Things (SIoT)—a promising framework built on the principles of the Semantic Web to address data heterogeneity and enable semantic interoperability. SIoT enriches IoT data by assigning meaning through standardized semantic structures. These structures describe IoT data, its context, and its relationships, facilitating integration and usability across diverse IoT applications [12]. At the core of SIoT lies ontology, which serves as the backbone for organizing and representing knowledge in a machine-readable format. Ontologies enable the consistent representation and sharing of IoT data. However, SIoT faces significant challenges related to ontology management. A key issue is the reliance of many existing solutions on specific, often rigid ontologies. This has led to a proliferation of conflicting or redundant definitions, making it difficult to ensure interoperability and scalability across diverse systems. The lack of a unified or standardized approach to ontology development and integration further exacerbates these challenges, hindering the seamless exchange of data and knowledge in the SIoT ecosystem. For instance, according to Linked Open Vocabularies for IoT (LOV4IoT), over 400 ontology-based vocabularies have been created [13].

To overcome these limitations, SIoT systems must prioritize the reuse of standardized ontologies. By adopting consistent and widely accepted frameworks, these systems can ensure a more accurate representation of the IoT domain, thereby enhancing data integration and interoperability across diverse platforms. Researchers have proposed several standard ontologies to address interoperability challenges and provide a unified structure for IoT data. However, lot of them have emphasized that these ontologies must be carefully evaluated and adapted to the specific requirements of different IoT applications to ensure their effectiveness and scalability. For example, the SSN ontology [14] was designed with a broad scope encompassing IoT and sensor applications. It provides detailed models of sensors, their characteristics, and measurements, but includes only basic actuator descriptions. It offers medium modularity and can be integrated with other ontologies. Similarly, the oneM2M ontology [15] focuses on M2M (Machine-to-Machine) IoT environments. It provides sensor and device constructs adapted to that context, includes actuator modelling suited to M2M infrastructures, and offers medium modularity with alignment capabilities toward SAREF and other ontologies. Finally, the SAREF ontology, proposed by ETSI [16], targets the IoT and smart building domains. It models both sensors and devices (including actuators) and stands out for its high modularity and explicit alignment with oneM2M and other frameworks. Together, these ontologies offer complementary strengths in domain coverage, modelling depth, alignment potential, and modular design, providing SIoT with a unified yet adaptable semantic foundation for enhanced interoperability and data integration.

While standardized ontologies such as SAREF, SSN, and oneM2M are fundamental for achieving semantic interoperability through consistent data models, their widespread adoption in unifying heterogeneous IoT data also contributes to the volume and scalability challenges. As devices generate vast amounts of data structured according to these ontologies, the resulting RDF datasets grow exponentially. This huge scale poses significant problems for traditional data management techniques. Specifically, the prevailing approach to storing and querying this semantic data relies on RDF triple stores alongside the SPARQL query language. However, as documented in literature and confirmed by our own baseline experiments, these triple store solutions often struggle to maintain acceptable query performance when dealing with the billions of triples characteristic of large-scale IoT deployments. The overhead associated with complex SPARQL query processing and indexing limitations within these stores creates critical bottlenecks, hindering real-time analysis and efficient data retrieval. These limitations necessitate the exploration of alternative architectures and indexing mechanisms beyond standard triple stores, prompting a closer inspection of specific RDF storage approaches and broader SIoT frameworks, as depicted in Table 1 and Table 2.

The Resource Description Framework (RDF) is a key component of SIoT, enabling the representation, querying, and retrieval of data within the ecosystem. Due to its simplicity, adaptability, and reusability, RDF has become a widely adopted standard across various sectors. Consequently, large RDF datasets have emerged, spanning diverse domains such as DBpedia [17], DBLP [18], Bio2RDF [19], and others [20]. These datasets contain billions of RDF triples, with some distributed across multiple files to enhance querying efficiency, as seen with Lehigh University Benchmark (LUBM)[21].

However, managing these massive RDF datasets presents significant challenges related to both storage and retrieval. To address these issues, researchers have proposed several RDF storage solutions, including triple tables [22], binary tables [23] and property tables [24]. These approaches are implemented in platforms such as Apache Jena [25], Blazegraph [26] and GraphDB [27], all of which utilize a triple store model and support SPARQL for querying RDF data across diverse sources.

While these established RDF storage solutions and platforms provide foundational support for semantic data, their architectural choices—particularly their reliance on the triple store model and SPARQL querying—directly impact the scalability and retrieval performance of SIoT systems built upon them. As discussed in the following section, many existing SIoT frameworks—such as M3, SEDIA, and SSNT —adopt these conventional triple store backends. Consequently, despite their advances in semantic interoperability, they often inherit the inherent limitations of these storage mechanisms. Specifically, the overhead associated with processing complex SPARQL queries across potentially billions of triples—combined with indexing strategies optimized for graph patterns rather than the vector-based similarity searches—leads to significant performance degradation at scale (as detailed for triple stores in Table 2 and demonstrated in our baseline comparisons in Section 6). This widespread reliance on architectures that are suboptimal for large-scale, high-speed retrieval serves as a key motivation for exploring alternative approaches which decouples from the traditional triple store model to achieve improved performance.

Data storage plays a critical role in enabling informed decision-making, preserving historical records and collective memory, and facilitating analysis and prediction. In the IoT domain, where vast numbers of devices generate continuous streams of data, effective storage becomes even more essential. This need drives the adoption of various technologies, including cloud computing, edge computing, and hybrid approaches that combine both [28]. SIoT extends IoT by providing a framework for processing data from a wide range of devices and organizing the results into RDF datasets. Several frameworks have been proposed to leverage semantic technologies within IoT. While these frameworks enhance interoperability and semantic understanding, they often face limitations—particularly in efficiently managing large-scale RDF data and addressing inherent data uncertainty. Table 1 presents a comparative summary of representative SIoT frameworks, outlining their methodologies, the limitations addressed by our work, and the improvements introduced by our proposed framework. Beyond individual frameworks, the choice of underlying technologies for storage, classification, and indexing significantly affects performance and suitability in SIoT applications. Table 2 compares common alternatives in these areas.

Table 1: Comparative analysis of representative SIoT frameworks

Framework	Methodology Key Aspects	Key Limitations Addressed by Our Work	How Proposed Framework Addresses Limitations
VICINITY [29]	Focus on interoperability (Web of Things), Semantic descriptions, Cloud-centric	Less explicit focus on massive RDF storage/retrieval optimization; Doesn't inherently handle data uncertainty.	Explicitly targets RDF retrieval scalability via custom indexing; Integrates fuzzy logic for nuanced data classification under uncertainty.
BIG-IoT [30]	Semantic API for interoperability, Uses schema.org, Cloud-based platform	Primarily targets API-level interoperability; Potential scalability bottlenecks if underlying storage isn't optimized for massive RDF query load.	Multi-layered architecture distributes load; Fog layer with custom indexing bypasses typical triple-store query overhead for specific retrieval tasks.
M3 Framework [31]	Fog/Cloud architecture, M3 Ontology, Uses Jena (Triple Store) for RDF storage	Reliance on standard triple stores (Jena) may lead to known scalability issues and SPARQL query overhead at large scale.	Avoids triple store bottlenecks by using a file-based RDF representation coupled with highly efficient, category-specific binary tree indexing for retrieval.
WooD [32]	Web Objects concept, Knowledge base storage (details sparse), Aimed at interoperability	Lack of implementation details; Likely relies on traditional knowledge base/triple store querying, risking retrieval delays.	Provides a concrete, implemented architecture with optimized indexing (binary tree) demonstrated to significantly outperform triple store retrieval times.
SEDIA [33]	Smart City focus, Reuses SSN, IoT-Lite, GeoSPARQL ontologies, Relies on RDF triple stores	Continued dependency on triple stores may pose scalability /retrieval time challenges; Less emphasis on fuzzy classification for nuanced states.	Implements custom binary tree indexing tailored for fast retrieval, bypassing SPARQL overhead; Integrates fuzzy logic for improved classification before indexing.
SSNT [34]	Lightweight ontologies (IoT-Lite, SSN), Uses	Relies on triple store (GraphDB), facing potential retrieval delays at	Provides experimental validation showing superior scalability and

	GraphDB (Triple Store) for cloud storage	scale; Lacks real-world evaluation/deployment details.	retrieval speed compared to triple stores; Evaluated with real-world derived data.
NoSQL databases [35]	Stores RDF data using NoSQL models (e.g., document, key-value, column-family). Leverages inherent NoSQL scalability and schema flexibility.	Potential loss of semantic expressiveness compared to native RDF stores. SPARQL querying often inefficient or requires complex mapping. Standard NoSQL indexes not typically optimized for semantic/vector similarity search.	Maintains RDF semantics via file representation & SAREF. Uses optimized binary tree indexing tailored for vector similarity (score + measurements), bypassing SPARQL overhead for targeted retrieval.
Proposed Framework	Edge (SAREF/RDF standardization) -> Fog (Fuzzy Classification + Binary Tree Indexing) -> Cloud (Storage); Focus on RDF scalability & retrieval speed	Addresses: Scalability limits of triple stores, SPARQL overhead, rigid classification of uncertain data.	Method: Multi-layer processing, Fuzzy logic for nuanced classification, Custom binary tree indexing optimized for categorized RDF file retrieval based on feature vectors (score + measurements).

Table 2: Comparison of technologies for SIoT data management components

Category	Technology Option	Suitability for SIoT Data (Heterogeneity, Volume, Velocity)	Scalability	Query Performance (Type)	Handling Uncertainty	Semantic Support	Strengths	Weaknesses
Storage Architecture	Centralized Cloud	High Volume, Moderate Velocity	High	Depends on DB; Can be bottleneck	Indirect	High (via DB)	Flexibility, Central management	Latency, Bandwidth costs, Single point of failure
	Edge/Fog Hybrid (Our Approach)	High Heterogeneity, Volume, Velocity	High	Optimized at Fog (Fast Retrieval)	Direct (Fog Logic)	High (Edge/ Fog)	Low latency (Edge), Reduced network load, Distributed processing/ storage	Architectural complexity, Synchronization challenges
	Triple Stores (Jena, Blazegraph)	High Heterogeneity, Volume	Medium Low	Good (SPARQL - pattern matching), Poor (Massive scale)	Low	Very High	Native RDF support, SPARQL standard	Scalability limits, Complex query overhead, Poor for vector similarity search
	Graph Databases (Neo4j)	Moderate Heterogeneity, High Volume	High	Excellent (Traversal), Moderate (Pattern), Poor (RDF Native)	Low	Mode-rate	Relationship focus, Scalability	Non-native RDF, Different query language (Cypher), Less ideal for pure semantic web integration
	NoSQL (Document/KV)	High Heterogeneity, Volume, Velocity	Very High	Varies (Key lookup fast, Complex queries slow)	Low	Low Mode-rate	Scalability, Flexibility	Reduced semantic expressiveness, Inconsistent query performance

Clustering/ Classification	Crisp Thresholds (e.g., Std MEWS)	Simple Data	N/A	N/A	Poor	N/A	Simple, Fast	Prone to misclassification at boundaries, doesn't model vagueness
	K-Means, DBSCAN	Moderate Volume	Mode-rate	N/A	Poor	N/A	Unsupervised discovery, Handles certain cluster shapes	Needs parameters (k, eps), Sensitive to noise/shape, not ideal for predefined categories or fuzzy boundaries
	Fuzzy Logic (Our Choice)	Heterogeneous, Uncertain Data	N/A	N/A	Very High	N/A	Handles vagueness/uncertainty, Interpretable rules, Nuanced output	Needs domain knowledge (rules/MFs), Can be computationally intensive
Indexing Mechanism	No Index (Linear Scan)	Low Volume	Poor	Poor (O(n))	N/A	N/A	Simple	Inefficient for large datasets
	Triple Store Indexes (SPO, etc.)	High Volume (but struggles at massive scale)	Medium Low	Good (Specific SPARQL patterns), Poor (General/Similarity)	N/A	High	Optimized for SPARQL	Tied to triple store limitations, not for vector similarity
	B-Trees / B+ Trees	Structured Data	High	Excellent (1D Range), Poor (Multi-dim Similarity)	N/A	Low	Standard, Efficient for range queries	Not directly suited for multi-dimensional similarity search
	R-Trees / QuadTrees	Spatial / Multi-dimensional Data	High	Good (Spatial Range/Proximity), Moderate (Similarity)	N/A	Low	Efficient multi-dimensional spatial queries	Can be complex, overhead for non-spatial similarity, Potential overkill
	Graph DB Indexes	Graph Data	High	Excellent (Node/Relationship lookup)	N/A	Mode-rate	Optimized for graph traversal	Not for RDF vector similarity
	Custom Binary Tree (Our Choice)	Categorized Vector Data	High (per tree)	Good (Avg O(log n) Similarity Search)	N/A	Mode-rate (via vector)	Tailored for vector similarity, Synergizes with classification	Needs balancing consideration, Worst-case O(n) if unbalanced

4 Proposed framework

Despite significant advancements in the Semantic Internet of Things (SIoT), critical challenges persist—particularly in efficiently managing the exponential growth of RDF data generated by the rapidly increasing number of IoT devices. Addressing this challenge requires a more innovative and scalable approach to data storage, organization, and retrieval to meet the growing demands of modern, data-driven applications. A promising solution lies in a layered architectural framework that seamlessly integrates edge, fog, and cloud computing components. This approach addresses critical challenges such as real-time data processing, efficient clustering, robust indexing, and scalable storage—ensuring optimized performance and resource utilization across all system layers.

Clustering is an effective technique for grouping similar data while maintaining clear separation between different elements. When addressing overlapping or ambiguous data, integrating fuzzy logic into the clustering process significantly enhances its effectiveness—improving categorization accuracy and enabling better handling of complex datasets. Furthermore, indexing techniques can significantly improve search efficiency by organizing data into hierarchical structures [36]. This

approach reduces the search space and accelerates query-based searches, enabling faster and more precise data retrieval. The practical deployment of such SIoT frameworks in real-world domains is essential for evaluating their effectiveness, identifying tangible benefits, and fostering broader adoption and innovation. These objectives form the focus of the following sections, where we demonstrate and discuss the capabilities of our proposed framework.

4.1 System architecture overview

Timely processing of IoT data, along with efficient storage and RDF retrieval, is critical. Cloud computing, with its bandwidth constraints, struggles to meet these real-time requirements. To address this limitation, an architecture that ensures low-latency processing and fast data access is essential for delivering responsive and efficient services. To meet these demands, we propose a robust edge-fog-cloud architecture, as illustrated in Figure 1. This architecture is structured into four layers:

1. Device Layer: Responsible for generating IoT data.
2. Edge Layer: Processes data near the devices to reduce latency and alleviate network congestion.
3. Fog Layer: Distributes storage and computation tasks across intermediate nodes.
4. Cloud Layer: Provides centralized storage and long-term data management.

This multi-layered design optimizes data handling across the IoT ecosystem by balancing speed, efficiency, and scalability to accommodate the growing demands of modern IoT applications—including those involving a large number of connected devices. The first layer, the IoT Device Layer, encompasses all devices with data collection capabilities —such as sensors, wearables, and

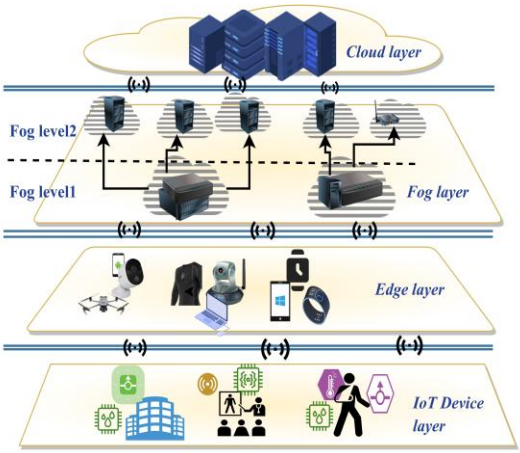


Figure 1: Edge-Fog-Cloud layer architecture for efficient IoT data processing and management

smart devices—and communication technologies like Wi-Fi, Bluetooth, and others used for data transmission. The Edge Layer includes resources located near these devices, such as gateways, routers, and local servers. Devices like smartwatches and desktops can also function as edge nodes. This layer is crucial for ensuring scalability across multiple devices, as it distributes initial processing load.

The Edge Layer processes data received from sensors and performs initial semantic enrichment. It incorporates a semantic engine that performs reasoning over the data, enabling more insightful analysis at this early stage. By leveraging the ontology, the semantic engine annotates the sensors' measurement values and generates results in RDF format. This enables more accurate interpretation and categorization of sensor data, laying the foundation for decision-making and further processing in subsequent layers of the system. However, due to the limited storage capacity for RDF files at the edge, the data must be transferred to the next layer—the Fog Layer. The Fog Layer acts as an intermediary, comprising more powerful fog nodes (e.g., servers, routers) located regionally or within specific facilities such as hospitals. It bridges edge devices and the central cloud, further alleviating network load and enhancing security. This layer aggregates data from multiple edge nodes within its domain. The Fog Layer operates through two key phases that are central to the efficiency of our framework:

1. Clustering Phase: Utilizes a fuzzy logic algorithm (Algorithm 1) to classify incoming RDF files from the Edge Layer into meaningful categories based on content.
2. Indexing Phase: Constructs category-specific binary trees (Algorithm 2) to organize and index the classified RDF data vectors, enabling highly efficient similarity-based retrieval (Algorithm 3).

By combining these phases, the Fog Layer efficiently manages data while minimizing network overhead—ensuring faster and more secure data handling within the IoT ecosystem. Figure 2 presents a visual representation of the complete data processing pipeline. This flowchart illustrates how the core algorithmic components, detailed in subsequent sections, interact within the layered architecture.

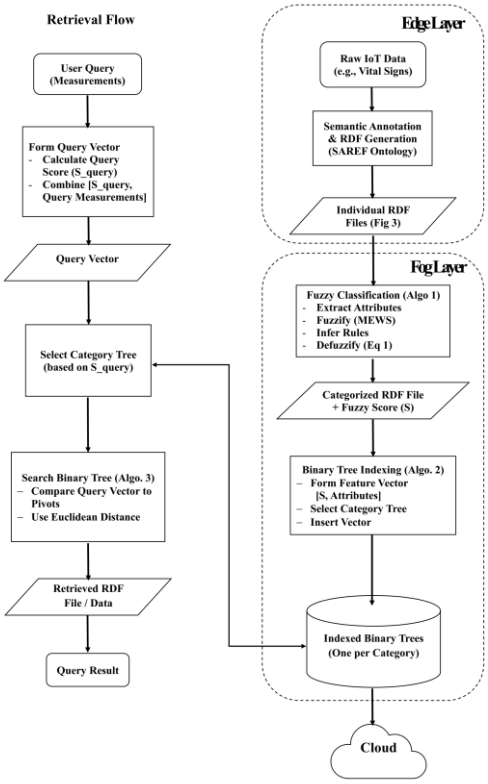


Figure 2: Data Processing and retrieval workflow

The proposed architecture is designed to support dynamic environments in which devices frequently join and leave the network. Device discovery, authentication, and connection management are primarily handled at the Edge Layer, utilizing standard IoT protocols and, potentially, an IoT platform's management plane. When a new device joins, the Edge Layer registers it and incorporates its data into the processing pipeline. Subsequent fuzzy classification and binary tree indexing in the Fog Layer operate on the data's characteristics (i.e., the feature vector derived from RDF), rather than relying on persistent device connections. If a device disconnects, the Edge Layer simply stops receiving its data, while previously generated and indexed information remains available. While optimizing specific protocols for high-frequency join/leave scenarios remains an area for future work, the decoupled nature of the data flow ensures that core data processing and indexing mechanisms remain robust and scalable in dynamic environments.

4.2 Framework modules: core components of the architecture

Internet of Things (IoT) devices typically generate and transmit raw data over the Internet. The inconsistency between devices makes processing and exchanging this data highly challenging. Semantic Web technologies help bridge the gap between heterogeneous devices and enable semantic interoperability. The Semantic Web is widely used for representing data; it supports advanced reasoning and decision-making and facilitates the integration of data from multiple sources. RDF serves as the foundational data model for the Semantic Web, allowing seamless integration and sharing of data across diverse applications and domains. It also provides a suitable model for storing any semantic web data. From IoT devices that collect data, to its processing and storage in RDF format, and finally to timely data retrieval, we propose an approach that combines Semantic Web technologies with data structuring mechanisms to enhance data organization and processing.

The proposed framework, depicted in Figure 3, is designed for IoT environments. It supports the management of data heterogeneity and stores the resulting data in RDF format. The framework relies on clustering and indexing of RDF files to enable rapid data retrieval. The approach consists of four main components:

4.2.1 Semantic representation

The first layer of the proposed approach addresses the challenge of IoT data heterogeneity. It leverages Semantic Web technologies and comprises a set of integrated modules. At its core lies the knowledge base module, which provides a formal structure to facilitate the sharing and reuse of knowledge. This module includes both an ontology and additional domain knowledge. The ontology defines a vocabulary for representing data collected from sensors by specifying relevant concepts and the relationships between them. It serves as the backbone of the Semantic Web, enabling semantic consistency and meaning-sharing across different IoT applications. While many existing methods support the creation of custom ontologies tailored to the IoT domain, reusing well-established ontologies significantly enhances collaboration and data exchange across heterogeneous systems.

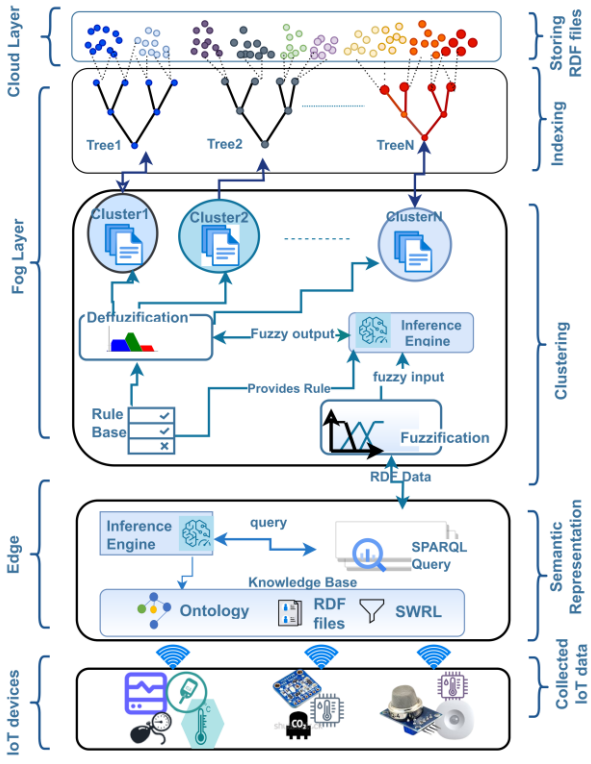


Figure 3: Core elements and structural components of the architecture

In particular, the adoption of the standardized SAREF ontology streamlines the alignment and sharing of ontologies, promotes interoperability, and ensures cohesive integration across diverse IoT platforms. Beyond the ontology itself, the knowledge base incorporates essential supplementary information about the IoT domain, including domain objectives, contextual information, historical measurement data, and more. This additional layer of knowledge enriches the ontology, offering a more holistic and accurate representation of the domain, thereby enabling deeper insights and improving decision-making and interoperability. The knowledge base module also integrates SWRL (Semantic Web Rule Language), which allows the expression of complex logic in the form of rules. These rules can be used to infer new, hidden knowledge from data stored in RDF format. This capability enables the discovery of implicit relationships and insights that are not explicitly stated in the original dataset. In addition to the knowledge base, the architecture includes an inference engine and a SPARQL engine. The inference engine reasons over existing data to infer new facts, while the SPARQL engine enables querying and retrieval of source data. To facilitate the use of this knowledge, preserve its semantic meaning, and enable interoperability across different applications, the information is stored in RDF file format. An RDF file is a machine-readable document that uses the RDF language to structure IoT data semantically.

For example, in the IoT domain, RDF files may contain historical sensor measurements, environmental information, and other contextual data. Since tracking the history of IoT data is highly beneficial, storing this information in RDF format simplifies its processing, analysis, and prediction. However, as the semantic component processes increasing volumes of data, the size of RDF files grows substantially, leading to challenges related to storage capacity and retrieval efficiency. To mitigate these issues, it is essential to introduce a module that structures and optimizes the data effectively. Clustering offers a promising solution by organizing data into manageable groups before the indexing process. This approach reduces the volume of data to be indexed, improving system efficiency and scalability.

While the current implementation leverages SAREF for standardization, the framework's modular architecture allows for adaptation to environments that already utilize other standard ontologies such as SSN or oneM2M. The semantic representation module, which operates primarily at the Edge Layer, is conceptually decoupled from the Fog Layer's fuzzy classification and indexing logic—both of which operate on feature vectors extracted from RDF data. Integration can be achieved either by developing ontology mapping layers (using OWL, RDFS, or SKOS) to translate SSN or oneM2M data into the SAREF structure expected downstream, or by directly adapting or replacing the semantic module to generate RDF based on the target ontology (e.g., SSN or oneM2M). While the latter approach would require reimplementing the RDF generation logic, it could offer tighter integration with existing systems.

Crucially, as long as the necessary numerical features can be reliably extracted from the resulting RDF—regardless of the specific ontology vocabulary used—for input into the fuzzy classifier and indexing vector, the core logic of the Fog Layer modules (Algorithms 1, 2, and 3) remains applicable with minimal modification. Therefore, adapting the framework primarily involves adjustments to the semantic representation layer and does not require a complete repair of the multi-layered architecture or its core data processing techniques.

Fragmentation often arises from heterogeneous data sources employing different data models. By using the SAREF ontology for semantic annotation directly at the Edge Layer, the framework establishes a consistent RDF structure for all incoming data before it is distributed. Each generated RDF file represents a coherent snapshot of related measurements, logically linked via the ontology as illustrated in Figure 3. This early standardization minimizes structural fragmentation from the outset. The Clustering Module groups these semantically coherent RDF files into categories based on their content. As a result, files representing similar states or events are logically grouped together, facilitating more targeted and efficient querying. Although data may still reside in numerous individual files, the Binary Tree Indexing approach enables efficient retrieval based on the content vector. A query for a specific state retrieves the most similar file(s) directly via the index, eliminating the need to manually reconstruct fragmented information across unrelated files or perform complex SPARQL joins in a potentially fragmented triple store. While the framework does not physically merge files, the combination of logical categorization and content-based indexing effectively addresses the retrieval challenges associated with fragmentation.

4.2.2 Clustering module

Operating primarily within the Fog Layer, as described in Section 4.1, the Clustering Module processes the RDF files produced by the semantic layer at the Edge. Following the semantic representation phase detailed in Section 4.2.1—where heterogeneous IoT data is standardized into a consistent RDF format using the ontology—the framework encounters the subsequent challenge of efficiently managing and retrieving information from the potentially vast volume of generated RDF files. While semantic standardization addresses issues of interoperability and shared meaning, it does not inherently resolve the scalability challenges associated with searching through large numbers of individual files, particularly when rapid access is required. Therefore, the next critical step involves organizing these semantically enriched RDF files based on their content to facilitate optimized access. This task is performed by the Clustering Module, which classifies the RDF files into meaningful categories based on extracted features and computed scores. By doing so, the module effectively structures the data, significantly reduces the search space, and lays the

foundation for the efficient indexing and retrieval mechanisms described in subsequent sections.

To address the challenges posed by massive data volumes and to improve retrieval efficiency, this module groups RDF files representing similar states into distinct categories. This categorization structures the data, reduces the search space for subsequent retrieval, and accelerates query performance by ensuring that related data is indexed together. The approach involves constructing a separate index—specifically, a binary tree—for each category. This significantly speeds up the search process, as each index contains only RDF files belonging to its respective group. Moreover, by employing fuzzy logic, the module enhances the framework's adaptability to dynamic environments characterized by uncertainty or gradually evolving data, ultimately supporting more robust and responsive data-driven decision-making.

Given that IoT data is often imprecise and vague, attempting to classify it using rigid thresholds can result in inaccurate outcomes. Fuzzy logic emerges as a highly suitable solution, specifically designed to handle the uncertainty and ambiguity inherent in such data. It enables reasoning under uncertainty by using degrees of membership rather than absolute categorizations. By mirroring human decision-making, fuzzy logic effectively models overlapping concepts and gradual transitions between data states. The decision to employ fuzzy logic is fundamentally driven by its superior capability to manage the vagueness and imprecision characteristic of real-world IoT data. Unlike traditional methods based on crisp thresholds which struggle with ambiguity at category boundaries and risk misclassifying borderline cases, fuzzy logic explicitly accounts for this imprecision. Through the use of membership functions, fuzzy logic allows a data point to possess partial membership in multiple linguistic categories.

The ability to represent partial truths produces a more nuanced output score (S), which facilitates more accurate and relevant classification into predefined categories. This enhanced level of precision is especially valuable in high-stakes domains such as healthcare and was qualitatively supported by expert review in our study. Furthermore, the rule-based structure inherent to many fuzzy systems offers a level of interpretability often lacking in complex black-box models. Alternative approaches were considered less appropriate for this task: crisp thresholds are inherently limited in their ability to represent ambiguity, while standard unsupervised clustering algorithms (e.g., K-Means, DBSCAN) are primarily intended for discovering unknown data groupings rather than performing fine-grained classification into predefined, semantically meaningful categories with inherently fuzzy boundaries.

It is important to note that within the context of our target healthcare application, this module performs fuzzy classification rather than unsupervised clustering. The objective is to assign each RDF file to one of the predefined categories, using fuzzy logic to enable a more nuanced and robust classification than that achieved through standard classification methods. The fuzzy system employs several standard components to perform this classification. The process begins by extracting relevant features—termed Attributes—from each RDF file f. As these files represent IoT measurements structured according to the SAREF ontology, the Attributes typically consist of specific numerical values such as pulse rate or temperature in health context.

Following extraction, the fuzzifier module converts these numerical attributes into fuzzy sets. It applies predefined membership functions, such as triangular or trapezoidal shapes, with boundaries derived from established guidelines. Each numerical value is assigned a degree of membership between 0 and 1 for the corresponding linguistic terms, such as "Normal" or "High1" in the case of heart rate. These terms and their associated membership functions are domain-specific and grounded in medical standards such as the MEWS system for vital signs. The output of this stage is a fuzzy representation of the input data. (e.g.: A heart rate of 115 may fuzzify to {Normal: 0.0, High1: 0.4, High2: 0.8}).

The core fuzzy inference component then utilizes a set of fuzzy rules—defined by domain experts—to evaluate the fuzzified inputs. These rules map combinations of fuzzy input values to output fuzzy sets that represent the target classification categories, such as risk levels. A representative rule might be: *IF Pulse is High1 AND OxygenSaturation is Low1 THEN Risk is Moderate*. The inference engine evaluates all applicable rules for the given input, aggregates the results, and produces a final fuzzy output set that indicates the file's degree of compatibility with each risk category. Finally, to convert the fuzzy output into a single crisp score, the defuzzification module applies the Centroid method, defined as:

$$S = \frac{\sum_{i=1}^{n} x_i \cdot \mu(x_i)}{\sum_{i=1}^{n} \mu(x_i)} \quad (1)$$

Where:

- S is the crisp output score (denoted as Score_i), x_i represents a point in the output domain (e.g., a specific risk level or severity index),
- $\mu(x_i)$ is the membership degree of the aggregated fuzzy output set at point x_i,
- n is the total number of discrete points considered within the output domain.

This resulting score S quantifies the degree of compatibility or relevance of a given RDF file f to a specific target class C_{ℓ_i}. It is this crisp value that is then passed forward to the classification and indexing modules within the SIoT architecture.

Algorithm 1 outlines the process for classifying RDF files into categories using the fuzzy logic engine. This algorithm leverages the computed fuzzy scores to assign each data point (i.e., RDF file) to the most appropriate predefined category. By calculating the compatibility score of each data point with all available categories, the algorithm effectively handles ambiguity and overlapping concepts—particularly useful in scenarios where clear boundaries between states are not well-defined. Based on these scores, each file is assigned to the single best-fitting category, thereby facilitating optimized data organization for subsequent indexing and retrieval.

Algorithm 1: Clustering with Fuzzy Logic Engine
Input:

- F = {f_1, f_2, ..., f_m} // *Dataset of RDF files*
- Cl = {Cl_1, Cl_2, ..., Cl_n} // *Initial set of clusters*
- FuzzyLogicEngine // *Module to compute membership scores*
- ε // *Convergence threshold for early stopping*

Output:
- Clusters with assigned RDF files

Steps:
1. Initialize Cl ← DefineInitialClusters(F) // *Use domain-specific rules*
2. T ← CalculateInitialThreshold(F, Cl) // *Compute initial threshold based on average membership scores*
3. Repeat:
4. AssignmentsChanged ← False // *Track changes in cluster assignments*
5. For each file f ∈ F:
6. Attributes ← ExtractAttributes(f)// *Extract features from f*
7. Scores ← [] // *Initialize empty list for scores*
8. For each cluster Cl_i ∈ Cl:
9. $Score_i$ ← FuzzyLogicEngine.CalculateScore(Attributes, Cl_i)
10. Append (Cl_i, $Score_i$) to Scores
11. BestCluster, BestScore ← max(Scores) // *Find cluster with highest score*
12. If BestScore > T:
13. Assign f to BestCluster
14. If f was previously assigned to a different cluster:
15. AssignmentsChanged ← True
16. Else:
17. createNewCluster(f)// *Initialize new cluster with f as centroid Append new cluster to Cl*
18. AssignmentsChanged ← True
19. UpdateClusters(Cl) // *Adjust centroids using weighted averages of assigned files*
20. T ← AdjustThreshold(F, Cl) // *Recalculate threshold based on updated clusters*
21. Convergence ← CheckConvergence(AssignmentsChanged, ε) // *Stop if changes are below ε*
22. Until Convergence = True or AssignmentsChanged = False
23. Return Cl // *Output clusters with assigned RDF files*

In the initialization phase of Algorithm 1, the set of classification categories C_l is defined. Although generic clustering techniques can be employed, the proposed framework favours a domain-specific approach in which these categories are predefined based on context-relevant semantic criteria. This ensures alignment with real-world conditions and facilitates the interpretation of results. The classification process requires the calculation of a threshold *T*, which represents the minimum score of confidence required to assign a given data file to a particular category. When a file's highest fuzzy compatibility score falls below this threshold, the system interprets it as indicative of uncertainty or weak association. In such situations, instead of generating a new category, the file may be allocated to a designated "uncertain" group or flagged for manual review, particularly in domains where categories are fixed and semantically meaningful.

The computed fuzzy score is the primary determinant for identifying the most appropriate classification. Unlike soft clustering methods such as Fuzzy C-Means, which assign partial membership values across multiple clusters and refine them iteratively, the approach adopted here aims for a crisp categorization. Each file is ultimately assigned to the most compatible predefined category, based on its highest fuzzy score. This strategy is especially advantageous in scenarios that require clear organizational structure and efficient retrieval, while still accommodating uncertainty and overlapping semantic boundaries through the use of fuzzy logic.

A complexity analysis of Algorithm 1 reveals its computational cost as follows: Let m represent the number of RDF files, n the number of predefined categories, I the number of iterations (potentially just one for simple classification), and d the feature dimensionality. Let C_{extract} be the cost of attribute extraction per file, and C_{fuzzy} the cost of fuzzy scoring per file per category (dependent on fuzzy rule complexity). The initialization step, based on domain-specific rules, typically has a complexity of $O(n)$. The overall time complexity is dominated by the loop that processes each file against every category, resulting in a total complexity of approximately:

$$O\left(I \cdot m \cdot \left(C_{\text{extract}} + n \cdot C_{\text{fuzzy}}\right)\right) \quad (2)$$

This expression highlights the algorithm's dependence on the dataset size m, the number of target categories n, and the computational costs associated with RDF feature extraction and fuzzy rule evaluation.

4.2.3 Indexation layer

Following the classification step—also performed within the Fog Layer—the Indexation Layer, operating in the same environment, employs indexing as a powerful and widely adopted technique to enable rapid access to large datasets. In this work, we utilize a binary tree structure to index RDF files within each category determined by the fuzzy classification module described in Section 4.2.2. A separate binary tree is constructed for each category; for example, one tree per MEWS risk level in the healthcare scenario. This strategy partitions the data based on its classified state prior to indexing. Each node N within these binary trees may contain references to up to two child nodes (leftChild, rightChild) and is designed to hold up to two pivot points (Pivot1, Pivot2). The basic structure of a node is defined as follows: Node { leftChild: Node | null; rightChild: Node | null; Pivot1: Vector | null; Pivot2: Vector | null; }, where Vector represents the feature vector associated with an RDF file.

The choice of this custom binary tree structure is motivated by the need for efficient retrieval from potentially massive collections of categorized RDF files—a critical requirement in which standard SIoT approaches, relying on triple stores and SPARQL queries, often struggle due to scalability.

In our framework, RDF data is transformed into categorized files, each represented by a multi-dimensional feature vector comprising the fuzzy score and relevant sensor measurements. The binary tree index is specifically optimized for fast similarity-based retrieval of these vectors within their assigned categories. Its primary advantage lies in the hierarchical partitioning of the vector space using Euclidean distance relative to the pivot vectors stored in each node. This structure enables an average-case retrieval complexity potentially approaching $O(\log n)$, where n is the number of files in a category's tree.

This logarithmic complexity represents a significant improvement over linear search methods and avoids the overhead associated with complex SPARQL pattern matching over large triple stores. Moreover, the indexing approach synergizes directly with the preceding fuzzy classification step: building separate trees per category drastically reduces the search space (n) for any given query, as the query's fuzzy score immediately directs the search to the appropriate tree.

While alternative multi-dimensional indexing structures exist (e.g., R-trees, QuadTrees), the chosen binary tree—empirically designed with two pivots per node for effective branching—offers a tailored solution for vector similarity comparison with lower implementation complexity for this specific use case. In contrast, traditional B-trees are not suitable for multi-dimensional similarity tasks, and graph databases introduce non-native RDF handling and alternative query paradigms that are not optimized for this type of vector-based retrieval.

The connection between the clustering phase (fuzzy classification) and the indexing phase is critical to the efficiency of the overall framework. The process begins once an RDF file f has been assigned to a specific category C_l^i, and its corresponding fuzzy compatibility score S has been computed. This fuzzy score S is primarily used to determine which category-specific binary tree the RDF file should be inserted into. After the appropriate tree is selected, an input feature vector is constructed to represent the RDF file for insertion. To clarify the vector formation process, this multi-dimensional vector explicitly combines the fuzzy score S with the relevant numerical measurement attributes (e.g., vital signs) extracted from the RDF file. A typical structure is:

$$\text{vector} = [S, \text{measurement}_1, \text{measurement}_2, \ldots, \text{measurement}_d]$$

This complete vector interacts with the binary tree structure during both insertion and retrieval operations. The binary tree structure is specifically designed to optimize data retrieval efficiency. Queries navigate the tree by comparing a query vector to the pivot vectors stored at each node. At each internal node, the algorithm calculates the distance to both pivots and proceeds down the branch associated with the closer pivot. This hierarchical search, guided by vector similarity, dramatically reduces the search space compared to conventional linear scans. The insertion of a new RDF file's feature vector into its designated category-specific binary tree is handled by Algorithm 2. This algorithm preserves the organization of the tree by placing new vectors based on their similarity to existing pivot vectors, ensuring both balance and retrieval efficiency.

Algorithm 2: Inserting RDF Files into Binary Tree

Input:

vector: The feature vector for an RDF file, including fuzzy score and measurement values.

N: The root node of the specific binary tree corresponding to the vector's category.

Output:

Updated binary tree root node.

Steps:

```
1. Function InsertVector(N, vector):
2. If N is null: // Tree/subtree is empty
3.     Return Creationofnode(vector) // Create root or new leaf
4. // --- Pivot Assignment ---
5. If N.Pivot1 is null:
6.     N.Pivot1 = vector // Assign vector as the first pivot
7.     Return N
8. Else if N.Pivot2 is null:
9.     N.Pivot2 = vector // Assign vector as the second pivot
10.    Return N
11. // --- Distance Computation & Child Node Direction ---
12. Else: // Both pivots exist, decide which subtree to traverse
13.    D1 = EuclideanDistance(vector, N.Pivot1) // Use Eq. 3
14.    D2 = EuclideanDistance(vector, N.Pivot2) // Use Eq. 3
15.    If D1 < D2: // Closer to Pivot1
16.       If N.leftChild is null:
17.          N.leftChild = Creationofnode(vector)
18.       Else:
19.  N.leftChild=InsertVector(N.leftChild, vector) //Recurse left
20.    Else: // Closer to Pivot2 (or equal)
21.       If N.rightChild is null:
22.          N.rightChild = Creationofnode(vector)
23.       Else:
24.  N.rightChild=InsertVector(N.rightChild, vector)//Recurse
                                                        right
25.  Return N // Return the (potentially updated) current node N
26.Function Creationofnode(vector):
27. // Creates a new node instance based on the defined structure
28. N_new = new Node()
29. N_new.Pivot1 = vector // The input vector becomes the first
                                        pivot of the new node
30. N_new.Pivot2 = null
31. N_new.leftChild = null
32. N_new.rightChild = null
33. Return N_new.
```

Expanding on the insertion logic, the process begins at the root node N of the relevant binary tree. If node N is null—indicating either an empty tree or an uninitialized subtree—the CreationOfNode function is invoked. This function instantiates a new Node object based on the structure previously defined, assigns the entire input vector as Pivot1 for the newly created node, and initializes all other fields to null. If the current node N is not null but contains fewer than two pivots, the input vector is assigned to the next available pivot slot (Pivot1 or Pivot2). It is important to clarify that the entire multi-dimensional input vector is used as the pivot; no aggregation or selection of specific values from the vector is performed. When both pivots are already present in node N, the algorithm proceeds to calculate the distance between the input vector and each of the node's existing pivot vectors. To ensure clarity on the distance computation involving multi-dimensional vectors, the algorithm employs the standard Euclidean distance metric in a multi-dimensional space.

$$d(\mathrm{v}, \mathrm{p}) = \sqrt{\sum_{i=1}^{n} (v_i - p_i)^2} \qquad (3)$$

Here, $d(v, p)$ denotes the Euclidean distance between the input vector v and a pivot vector p, with the sum computed across all i dimensions of the vectors—including the fuzzy score and all associated measurement values. Based on which pivot (Pivot1 or Pivot2) yields the smaller distance (denoted as D_1 or D_2), the algorithm recursively calls the InsertVector function on the corresponding child node (leftChild or rightChild). This

recursive process ensures that similar vectors are grouped within the same subtrees, continuing until the input vector is either inserted as a pivot in an existing node or assigned as Pivot1 in a newly created leaf node. The structural integrity and efficiency of the binary tree are maintained through this recursive insertion strategy, which is guided by Euclidean similarity within the multi-dimensional feature space.

4.2.4 Data search and retrieval

The data retrieval process leverages the indexes created and maintained within the Fog Layer. While a user query may originate from an application interfacing with the Cloud Layer or other external systems, the core search mechanism—detailed in Algorithm 3—executes primarily within the Fog Layer, utilizing the locally held binary tree indexes. Efficient data retrieval is essential for extracting value from large-scale IoT datasets. Our approach uses the structured binary trees built in the Indexation Layer (Section 4.2.3) to perform rapid similarity searches for RDF files based on their feature vectors, aiming to optimize both speed and accuracy in retrieving relevant data.

The search process begins with the formulation of a query, typically represented by a set of target measurement values. This input is first processed through the same fuzzy logic engine used during the clustering/classification phase (Section 4.2.2) to compute a query fuzzy score (S_{query}). This score determines which category-specific binary tree should be queried. A query vector ($\text{vector}_{\text{query}}$) is then constructed—analogous to the vectors used for insertion—by combining the query score with the target measurement values:

$$\text{vector}_{\text{query}} = [S_{\text{query}}, \text{query_measure}_1, \ldots, \text{query_measure}_d]$$

The search within the selected tree begins at the root node N. If N is null, the search terminates with no relevant data found. If N is not null, the algorithm compares the query vector to the pivot vectors stored in the node using standard multi-dimensional Euclidean distance. Regarding mixed value types, both the fuzzy score and the numerical measurements are treated as numerical dimensions within the vector space. The Euclidean distance calculation inherently handles this multi-dimensional comparison:

$$d(v_{\text{query}}, p) = \sqrt{(S_{\text{query}} - S_{\text{pivot}})^2 + \sum(\text{query_measure}_i - \text{pivot_measure}_i)^2} \quad (4)$$

Here, v_{query} is the query vector, p is a pivot vector, S denotes the fuzzy score component, and measure_i represents the i-th measurement value. Each dimension—both score and measurements—is treated equally in the distance computation. No explicit weighting is applied in the standard Euclidean distance formula; however, the relative influence of the fuzzy score versus the measurement values can be implicitly adjusted during the feature scaling phase if desired. In our implementation, standard feature scaling was used.

Algorithm 3 outlines the recursive traversal logic used during data retrieval. At each node N, the algorithm calculates the distance $distance_1$ between the query vector and $N.Pivot1$. If $N.Pivot2$ is present, a second distance $distance_2$ is also computed relative to that pivot. The algorithm first checks for an exact match—that is, whether either distance equals zero. If no exact match is found, traversal is guided entirely by the Euclidean distances: if $distance_1 < distance_2$, the algorithm proceeds recursively down the left subtree (closer to *Pivot1*); otherwise, it traverses the right subtree (closer to *Pivot2*). It is important to note that the fuzzy score serves primarily to select the appropriate binary tree during query initialization. Within the selected tree, the search is based solely on geometric proximity in the multi-dimensional feature space, as determined by Euclidean distance. This ensures that the search path consistently moves toward the vectors most similar to the query vector, considering all included features—both the fuzzy score and the measurement values. The recursive process continues until an exact match is identified or a null node (indicating a leaf or end of branch) is reached. This hierarchical search significantly reduces the number of comparisons required, offering a clear performance advantage over traditional linear scanning.

Algorithm 3: Data Search & Retrieval

Input:
N: A node in the binary tree (currently visited).
vector_query: The input query vector (score + measurements).
Output:
The retrieved RDF file's representative vector (Pivot) or null.

```
Procedure SearchRDFFiles(N, vector_query):
1. If N is null:
2.     Return null  // Reached end of branch, no match found
3. distance1 ← EuclideanDistance(vector_query, N.Pivot1)
4. If distance1 == 0:
5.     Return N.Pivot1  // Exact match with Pivot1
6. If N.Pivot2 is not null:
7.     distance2 ← EuclideanDistance(vector_query, N.Pivot2)
8.     If distance2 == 0:
9.         Return N.Pivot2  // Exact match with Pivot2
10.    If distance1 < distance2:
11.        Return SearchRDFFiles(N.leftChild, vector_query)  //
                   Recurse left
12.    Else:
13.        Return SearchRDFFiles(N.rightChild, vector_query)  //
                   Recurse right
14. Else:
15.    Return SearchRDFFiles(N.leftChild, vector_query)
```

The efficiency of the proposed binary tree indexing approach (Algorithm 3) merits comparison with alternative retrieval methods. In the best-case scenario, the query complexity of our method is approximately $O(d \cdot \log \quad m)$, where m is the number of indexed RDF files and d is the dimensionality of the feature vector, assuming a reasonably balanced tree. However, as with many tree-based structures, the worst-case query complexity can degrade to $O(d \cdot m)$ if the tree becomes highly unbalanced due to non-uniform data insertion patterns. This performance stands in contrast to graph traversal techniques commonly used in RDF triple stores. While such methods can be efficient for specific SPARQL relationship queries using pre-optimized indexes, their worst-case complexity for complex pattern matching can

be significantly higher—potentially polynomial—and they are not inherently designed for the type of vector-based similarity search employed in our framework. Standard hash-based indexing offers excellent average-case performance ($O(1)$) for exact-match lookups, but it is inherently unsuitable for distance-based similarity searches without specialized adaptations such as Locality-Sensitive Hashing (LSH), which introduces approximation and may compromise accuracy. Therefore, while we acknowledge the possibility of worst-case $O(d \cdot m)$ performance, the proposed binary tree structure represents a task-specific, efficient solution for retrieving RDF files based on similarity within the multi-dimensional feature space defined by measurement values and fuzzy scores. It outperforms traditional graph traversal or hashing approaches in direct applicability to this retrieval objective.

5 Application scenario: IoT-driven enhancements in healthcare

Healthcare is a critical priority for governments and health organizations as they strive to protect populations from disease, enhance diagnostic capabilities, and promote overall well-being. The integration of IoT technologies has transformed this sector, introducing advanced capabilities that improve diagnostic accuracy, reduce costs and delays, and enable continuous patient monitoring. This domain is particularly well-suited for the application of our proposed approach, which can have a significant impact. Timely and accurate data management is essential in healthcare, as it not only has the potential to save lives but also to improve patient outcomes. By enabling more efficient and reliable data processing, our approach supports better clinical decision-making and enhances the overall quality of care. Vital signs—such as heart rate, blood pressure, and oxygen saturation—are essential indicators of a patient's condition [37]. IoT devices enable the real-time collection of these measurements and facilitate immediate transmission to healthcare providers or hospitals. This capability supports early detection of potential health issues and enables continuous monitoring, particularly for patients with chronic conditions. Beyond data collection, IoT technologies rely on robust architectures to process, store, and retrieve this critical information [38]. Paradigms such as cloud computing offer scalable solutions for data storage and processing, while semantic processing addresses data heterogeneity—ensuring that diverse data formats can be effectively integrated. These architectures facilitate the seamless transmission and retrieval of actionable insights, empowering healthcare professionals to analyze patient data efficiently [39]. Our approach aligns with these objectives by introducing a semantic-driven IoT architecture that enables faster and more accurate data processing while ensuring proper categorization and storage. By addressing key challenges such as data heterogeneity and retrieval latency, this framework equips healthcare providers with the tools needed to make informed, timely decisions—ultimately improving the quality of care and patient outcomes.

According to the proposed IoT framework, the healthcare system architecture is structured into multiple layers. The first layer comprises all health-related IoT devices responsible for generating and transmitting patient data. These devices include sensors capable of measuring vital signs—such as wearable devices, smartphones, and other smart technologies—with limited memory and processing capabilities. They enable seamless real-time data processing and analysis, supporting immediate insights and decision-making at the edge of the network. These smart devices utilize Semantic Web technologies to address data heterogeneity, producing RDF files for standardized representation. The third layer consists of devices such as routers and servers, deployed within hospitals and covering broader geographic areas. This layer is subdivided into Fog Level 1, which handles the clustering of RDF files, and Fog Level 2, which performs indexing. The final layer represents the Cloud, equipped with greater storage and processing capabilities. It functions as the central infrastructure for nationwide healthcare data management and coordination.

Representing patient data using RDF offers a structured and standardized method for managing healthcare information. RDF facilitates the integration of diverse data sources—including vital signs, medical histories, and IoT device readings—into a machine-readable format that promotes interoperability and semantic understanding. By organizing data into triples (subject, predicate, object), RDF captures the relationships between data points, enabling efficient querying, sharing, and analysis of patient information. This model supports effective decision-making, real-time monitoring, and system scalability, making it well-suited for modern healthcare environments.

The creation of RDF files marks the final step within the semantic layer, transforming raw IoT data into structured, meaningful, and actionable formats. After collecting health-related data, the system applies the SAREF ontology to standardize and organize information into a consistent, machine-readable structure. Furthermore, ontology-based SWRL rules are used to infer additional insights about the patient, aiding in the generation of relevant services and personalized recommendations. An inference engine is also incorporated to validate and enhance the accuracy of the generated knowledge. This seamless integration of semantic technologies improves both decision-making and service delivery by enabling more informed and context-aware actions. An example RDF file is shown in Figure 4, illustrating how patient data is organized and semantically enriched for practical use.

To support the early detection of health deterioration, the framework incorporates an Early Deterioration Identification System based on the Modified Early Warning Score (MEWS).

```
@prefix saref4health: <https://saref.etsi.org/saref4health/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix patient: <http://example.org/patient/> .
@prefix vital: <http://example.org/vitalSigns/> .

# Patient Information
patient:12345 a saref4health:Patient ;
    saref4health:hasName "Sami" ;
    saref4health:hasAge "45"^^xsd:integer ;
    saref4health:hasGender "Male" ;
    saref4health:hasMedicalRecordID "MR12345" .
# Vital Signs Measurements
vital:HeartRate_001 a saref4health:HeartRateMeasurement ;
    saref4health:hasValue "72"^^xsd:integer ;
    saref4health:hasUnit "bpm" ;
    saref4health:hasTimestamp "2023-10-15T08:30:00Z"^^xsd:dateTime ;
    saref4health:measuredBy patient:12345 .
vital:BloodPressure_001 a saref4health:BloodPressureMeasurement ;
    saref4health:hasSystolicValue "120"^^xsd:integer ;
    saref4health:hasDiastolicValue "80"^^xsd:integer ;
    saref4health:hasUnit "mmHg" ;
    saref4health:hasTimestamp "2023-10-15T08:30:00Z"^^xsd:dateTime ;
    saref4health:measuredBy patient:12345 .
# Device Information
vital:WearableDevice_001 a saref4health:WearableDevice ;
    saref4health:hasDeviceID "WD12345" ;
    saref4health:hasManufacturer "HealthTech Inc." ;
    saref4health:hasModel "SmartWatch X200" ;
    saref4health:monitors patient:12345 .
```

Figure 4: Patient's RDF file.

MEWS is a clinical scoring method that evaluates five vital signs: *blood pressure, pulse, oxygen saturation, respiratory rate, and body temperatur*e. Each vital sign is assigned a score from 0 to 3, where 0 indicates a normal state, and higher scores reflect increasing risk levels. Clinicians calculate a patient's overall MEWS score by summing the individual scores, resulting in a total value ranging from 0 to 14. Each incremental increase in the MEWS score signifies a greater health risk and may prompt a change in the treatment plan. For example, an increase of one point suggests heightened clinical concern requiring immediate medical attention. Additionally, RDF files are structured to categorize patient health status into predefined MEWS score ranges—such as [0–1], [1–2], ..., [13–14]—ensuring precise classification of patient conditions for continuous monitoring and timely intervention.

To apply our framework to the healthcare scenario, we establish a direct connection between the raw vital sign data collected via IoT devices and the fuzzy classification process. The MEWS system serves as the foundational structure. As previously described, MEWS assigns scores (ranging from 0 to 3) to specific intervals of vital signs (e.g., a pulse rate below 40 receives a score of 2, a value between 51–90 receives 0, and values above 130 receive a score of 3). These MEWS ranges and their associated scores form the direct basis for defining the inputs to our fuzzy logic system. In particular, the fuzzification process translates raw numerical vital sign measurements into fuzzy linguistic terms based on the MEWS-defined scoring ranges. For each vital sign—such as systolic blood pressure, heart rate, oxygen saturation, and temperature (i.e., the Attributes extracted from the RDF file)—we define a corresponding set of fuzzy linguistic terms aligned with MEWS scoring levels. For example, for heart rate, the terms might include: Low2 (corresponding to MEWS score 2 range <40), Low1 (score 1 range 41-50), Normal (score 0 range 51-90), High1 (score 1 range 91-110), High2 (score 2 range 111-130), and High3 (score 3 range >130).

Membership functions are then defined for each linguistic term, mapping the numerical value of a vital sign to a degree of membership (ranging from 0 to 1). As illustrated in Figure 5 (for pulse rate), these membership functions—trapezoidal—are shaped directly based on MEWS ranges.

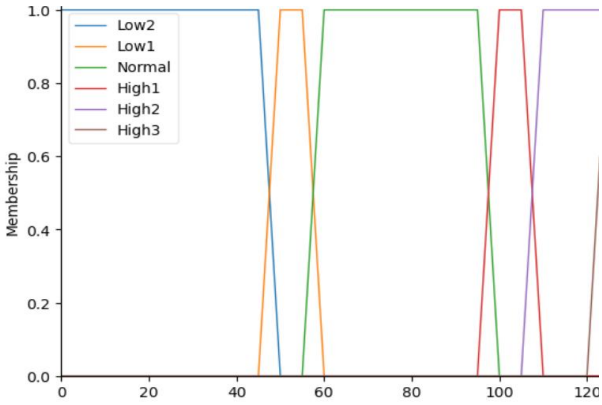


Figure 5 : Membership function of pulse rate

This allows for smooth transitions between categories rather than abrupt thresholds. For instance, a heart rate of 95 bpm may have a high degree of membership in High1, but also a small degree in Normal, reflecting uncertainty near clinical boundaries. This fuzzified representation of all relevant vital signs becomes the input to the fuzzy inference engine. The fuzzy inference engine applies a set of predefined fuzzy rules designed to mirror the clinical logic behind the MEWS system, but operating on fuzzy rather than crisp inputs. These rules—often implemented using Mamdani-style inference [40]—combine fuzzy states across multiple vital signs. For example: *IF Pulse is High1 AND OxygenSaturation is Low1 then Risk IS Moderate*. The results of these rules are aggregated into a final fuzzy output set that characterizes the patient's overall risk profile.

The final step, defuzzification—typically performed using the centroid method (see Equation 1)—converts the fuzzy output set into a single crisp value: the fuzzy MEWS score S. This score represents the system's comprehensive assessment of the patient's condition, analogous to the standard MEWS score but enhanced through fuzzy logic to better manage uncertainty and borderline cases. The calculated score is used to classify the patient into one of 15 predefined risk categories (see Table 3). For instance, a score between 7.0 and 8.0 may correspond to the *Elevated Risk* category. This process illustrates how raw healthcare data (vital signs) are transformed into fuzzy inputs, processed using clinically grounded rules, and translated into a decision-making score. The fuzzy score not only determines the patient's risk category C_l^i, but also guides the insertion of the patient's feature vector (comprising S and original measurements) into the appropriate binary tree using Algorithm 2, directly integrating fuzzy classification into the retrieval framework. The shapes and boundaries of the membership functions were derived directly from the clinically validated MEWS scoring ranges. For example, the *Normal* pulse range ([51–90]) forms the core of the corresponding membership function, while trapezoidal shapes are used for transitional zones. These slightly overlap adjacent ranges (e.g., *Low1*, *High1*) to accurately model the ambiguity near clinical thresholds. This design ensures that the fuzzy representation remains grounded in established medical practice.

As shown in Figure 5, membership functions for vital signs are used as inputs to a set of predefined fuzzy rules structured in the form: *IF antecedent (input condition) THEN consequent (risk level)*. These rules are based on combinations of vital signs such as heart rate or oxygen saturation, each represented by a linguistic term (e.g., *High*, *Low*) with associated membership degrees.

The fuzzy inference system uses Mamdani-style rules [40] and logical operators such as AND and OR to produce a fuzzy output set, which is then defuzzified into a single global MEWS score. This score corresponds to one of 15 predefined health statuses—ranging from *Optimal Health* to *Unknown Risk* —as outlined in Table 3. If a score does not fall within a category's defined range, the system proceeds to evaluate the next category until the appropriate classification is identified.

Introducing fuzzy logic into clinical systems requires careful design to ensure reliability, interpretability, and clinical alignment. Our approach incorporates four key safeguards:

1. Clinical Grounding: Membership functions and fuzzy rules were derived directly from MEWS guidelines, ensuring precise alignment with medical standards and avoiding the introduction of untested diagnostic logic.
2. Modeling Natural Uncertainty: Rather than introducing ambiguity, fuzzy logic models inherent uncertainty and gradual physiological transitions—features that crisp threshold fail to capture. This reduces misclassification risks for borderline cases.
3. Interpretability: Unlike black-box machine learning models, the rule-based structure of the fuzzy system allows clinicians to inspect, understand, and adjust the system logic. The use of clear linguistic terms and IF-THEN rules ensures transparency.
4. Expert Validation: A practicing physician can review selected patient cases to assess the accuracy and clinical relevance of the fuzzy logic-enhanced MEWS system. Such expert evaluations can confirm that the system often produces more nuanced and context-appropriate classifications—particularly in borderline scenarios where traditional MEWS scores may be ambiguous. This type of validation is essential to ensure that the fuzzy logic approach enhances, rather than compromises, the reliability of clinical risk assessment.

By grounding the fuzzy logic system in clinical knowledge, accurately modeling uncertainty, ensuring transparency, and validating outputs with expert feedback, our approach may deliver a robust and trustworthy enhancement to traditional MEWS scoring. It enables precise, interpretable, and clinically meaningful classification of patient risk in real-time healthcare environments.

It is important to clarify that the objective of incorporating fuzzy logic in our framework is not to perform unsupervised clustering (as with algorithms like k-means or DBSCAN), which aim to discover previously unknown data structures. Instead, we leverage fuzzy logic to enhance classification accuracy within the predefined, clinically meaningful MEWS risk categories. Physiological data inherently exhibits vagueness and gradual transitions between health states—characteristics that fuzzy logic is uniquely suited to model through the use of membership functions. In contrast, the rigid thresholds of standard MEWS scoring and the crisp assignments of traditional clustering algorithms are ill-equipped to represent these nuanced transitions. Moreover, standard clustering approaches do not utilize the clinical relevance embedded in the MEWS categories and are thus less suitable for this application. As a result, our primary evaluation focuses on demonstrating the improvement in classification performance achieved by integrating fuzzy logic, using the standard MEWS system as the direct clinical baseline for comparison.

Table 3: Health status corresponding to the defined ranges**.**

Range	Category	Description
[0–1]	Optimal Health	All indicators are within excellent ranges.
[1–2]	Healthy Normal	Slight variations within the healthy range, no concerns.
[2–3]	Mild Variation	Slight deviation from the ideal range but still acceptable.
[3–4]	Low Alert	A single parameter slightly outside the normal range, minimal risk.
[4–5]	Moderate Alert	Two or more parameters are borderline abnormal, requiring monitoring.
[5–6]	Potential Concern	Mild irregularities across several parameters; action recommended.
[6–7]	Early Warning	Detectable issues that may indicate emerging problems.
[7–8]	Elevated Risk	Significant deviation in one or two key indicators; requires action.
[8–9]	Moderate Risk	Persistent deviations suggesting underlying conditions.
[9–10]	High Risk	Clear signs of health risk; needs immediate attention.
[10–11]	Critical Risk	Severe abnormalities; potential medical emergency.
[11–12]	Unstable	Multiple parameters indicate serious health instability.
[12–13]	Acute Danger	Immediate medical intervention required.
[13–14]	Severe Emergency	Life-threatening abnormalities across vital signs.
[14–15]	Unknown Risk	Insufficient or inconsistent data to assess risk level.

In healthcare systems, efficient indexing of medical records is also essential, as it allows for rapid and accurate retrieval of patient data—crucial for timely decision-making and improved clinical outcomes. Effective indexing ensures that key information—such as health status, treatment history, and medical records—remains readily accessible, even within the vast and complex structure of healthcare datasets. Our indexing approach relies on constructing a Binary Tree, in which each node holds two pivot points. Given that RDF files are classified into 15 distinct categories, the system creates a separate Binary Tree for each category, resulting in a total of 15 trees. Algorithm 2 outlines the procedure for inserting a new RDF file into the corresponding Binary Tree. The input vector for indexing is constructed using the patient's vital sign measurements along with the fuzzy score generated during classification.

In the following section, we present a practical implementation using a patient RDF file containing vital sign information. This example demonstrates how an RDF file is inserted into and retrieved from a Binary Tree structure comprising 15 distinct categories. Each class is represented by its own Binary Tree, with each node maintaining two pivot vectors to support efficient similarity-based indexing.

1. Input: The patient RDF file (Figure 4)
2. Fuzzification

Fuzzification converts crisp input values into fuzzy sets using membership functions. Each vital sign (v_i) is mapped to a fuzzy linguistic term (e.g., Very Low, Low, Normal) based on its range.

$$\mu_A(x): X \rightarrow [0,1]$$

Where:

- $\mu_A(x)$: Membership function of fuzzy set A for input x
- X: The domain of input values (e.g., Pulse, Blood Pressure)

Example for Pulse:

$$\mu_{\text{VeryLow}}(x) = \begin{cases} 1, & x \leq 40 \\ \frac{50 - x}{10}, & 40 < x \leq 50 \\ 0, & x > 50 \end{cases} \quad (5)$$

Combined Membership Function: Each vital sign's fuzzy score is calculated based on predefined ranges, producing membership values (μ) for each linguistic category.

Fuzzy Score for Pulse=

$$\mu_{\text{VeryLow}}(x)+\mu_{\text{Low}}(x)+\mu_{\text{Normal}}(x)+\mu_{\text{High}}(x)+\mu_{\text{VeryHigh}}(x) \quad (6)$$

The fuzzy rule base consists of rules combining all vital signs:

Rule: If μ_{v1} is A and μ_{v2} is B and …, then $\mu_{\text{Output}} = C$

Where:

- $v_1, v_2, \ldots, v_n$: Vital signs (e.g., Pulse, Temperature)
- $A, B, \ldots$: Membership values of each input
- C: Output class (e.g., Low Alert)

Example Rule:

If μ_{Pulse} is Normal and $\mu_{\text{Temperature}}$ is High, then Output = Moderate Alert

The fuzzy rules are then aggregated using fuzzy logic operators (e.g., AND, OR) to calculate the overall membership of the output.

Using AND (Minimum Operator):

$$\mu_{\text{Output}} = \min(\mu_{\text{Pulse}}, \mu_{\text{Temperature}}, \ldots)$$

Using OR (Maximum Operator):

$$\mu_{\text{Output}} = \max(\mu_{\text{Pulse}}, \mu_{\text{Temperature}}, \ldots)$$

3. Defuzzification

Defuzzification converts the fuzzy output into a crisp score (S), which represents the patient's overall status. The centroid method is commonly used:

$$S = \frac{\int_{\text{Range}} x \cdot \mu_{\text{Output}}(x)dx}{\int_{\text{Range}} \mu_{\text{Output}}(x)dx} \quad (7)$$

Where:

- x: Crisp value within the fuzzy range
- $\mu_{\text{Output}}(x)$: Membership function of the output

4. Euclidean Distance for Indexation

The fuzzy score (S) and vital sign values are combined into a vector (v) for comparison within the binary tree.

$$\boldsymbol{v} = [S, v_1, v_2, \ldots, v_n]$$

The Euclidean distance between two vectors ($\mathbf{v}_i$ and $\mathbf{v}_j$) is calculated as:

$$d(\mathbf{v}_i, \mathbf{v}_j) = \sqrt{\sum_{k=1}^{n} (v_{i,k} - v_{j,k})^2}$$

5. Binary Tree Search

Using the calculated distance, the binary tree is traversed to find the closest match:

1. Compare $\boldsymbol{v}$ to pivot nodes:

$$d_1 = d(\boldsymbol{v}, \text{Pivot1})$$
$$d_2 = d(\boldsymbol{v}, \text{Pivot2})$$

Move to the left or right child based on the smallest distance: If $d_1 < d_2$, traverse left. Otherwise, traverse right.

A. Input Data

1. Patient Data (RDF File):
   - Measurement values: Pulse: 85; Blood Pressure: 120/80; Respiration Rate: 16; Body Temperature: 37°C
   - Fuzzy Score: 2.3 (calculated using fuzzy logic)
2. Binary Tree Example (Class "Healthy Normal"):
   - Root Node: Pivot 1: [85, 120, 16, 37, 2.0] and Pivot 2: [90, 125, 18, 38, 2.5]
   - Left Node: Pivot 1: [80, 115, 14, 36.5, 1.5] and Pivot 2: [82, 118, 15, 36.8, 1.8]
   - Right Node: Pivot 1: [92, 130, 19, 38.5, 3.0] and Pivot 2: [95, 135, 20, 39, 3.5]

B. Insertion Process

1. Determine the Fuzzy Score:
   - The fuzzy logic system calculates a score of 2.3, placing the RDF file in the class "Healthy Normal"
   - The corresponding Binary tree for "Healthy Normal" is used.
2. Construct the Input Vector:
   - Vector = [85, 120, 16, 37, 2.3].
3. Insert into Binary Tree (Algorithm 2):
   - Compare the vector to the Root Node pivots using Euclidean distance:
     - Distance to Pivot 1 = 0.3
     - Distance to Pivot 2 = 0.2

- The closest pivot is Pivot 2, so move to the Right Node.
- Compare the vector to the Right Node pivots:
  - Distance to Pivot 1 = 0.7
  - Distance to Pivot 2 = 1.0
- The closest pivot is Pivot 1, so the RDF file is inserted in the Right Node under Pivot 1.

C. Searching Process

1. Calculate the Fuzzy Score:
   - For retrieval, the fuzzy score is recalculated as 2.3 to determine the class “Healthy Normal”
2. Traverse the Tree (Algorithm 3):
   - Start at the Root Node and compare the vector to the pivots:
     - Distance to Pivot 1 = 0.3
     - Distance to Pivot 2 = 0.2
   - Move to the Right Node (closest pivot).
   - Compare the vector to the Right Node pivots:
     - Distance to Pivot 1 = 0.0 (match found).
   - The RDF file is retrieved successfully.

D. Outcome

1. Insertion Result:
   - As presented in Figure 6: Efficient and Scalable insertion of RDF file., the RDF file with vector [85, 120, 16, 37, 2.3] is inserted into the “Healthy Normal” Binary tree at the Right Node under Pivot 1.
2. Search Result:
   - The algorithm retrieves the RDF file from the Right Node based on the closest match to the input vector.

The algorithm demonstrates efficiency by narrowing down searches to specific pivots and clusters, reducing unnecessary computations. It is highly scalable, as new RDF files can be seamlessly added to the appropriate binary tree, ensuring that future searches maintain their efficiency even as data volumes grow. Additionally, the use of Euclidean distance calculations enhances accuracy by reliably retrieving the closest match, preserving the system's precision and ensuring consistent results.

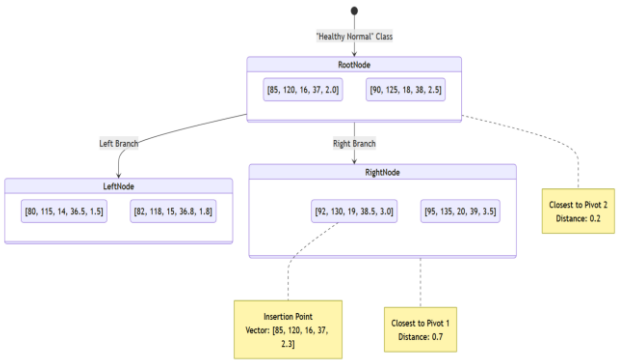


Figure 6: Efficient and Scalable insertion of RDF file.

6 Experimental evaluation

The integration of semantic technologies and fuzzy logic into IoT systems offers a powerful approach to enhancing data interoperability, accuracy, and efficiency. This study explores the implementation of a comprehensive framework designed to process, classify, and retrieve IoT data through the use of RDF files, semantic ontologies, and fuzzy logic. By leveraging these advanced computational techniques, the proposed framework effectively addresses key challenges such as data heterogeneity, imprecise sensor measurements, and the growing need for scalable storage and retrieval mechanisms. The results highlight the potential of combining semantic reasoning and fuzzy inference to manage the inherent complexity of IoT-driven systems—particularly in high-stakes environments like healthcare. Experimental validation confirms the robustness and efficiency of the proposed solution, emphasizing its practical applicability in real-world contexts. The outcomes pave the way for more reliable, intelligent, and semantically aware IoT architectures capable of supporting critical decision-making and delivering improved system performance at scale.

6.1 Data collection

To address the lack of publicly available datasets that accurately represent IoT-based patient health monitoring, the BIDMC dataset [41] —a subset of the larger MIMIC dataset [42] —was utilized. MIMIC is an open-access repository containing anonymized patient data collected from a U.S. hospital, offering a valuable resource for healthcare research and real-world evaluation. The dataset includes detailed records of patients’ vital signs, such as blood pressure, oxygen saturation, heart rate, and other essential physiological metrics. The BIDMC data specifically consists of numerous files—often in formats such as CSV—which are derived from original waveform data [43]. For our experiments, we used a selection of these CSV files, each typically containing several hundred timestamped measurements for multiple patients. This structure served as the foundation for generating our experimental RDF datasets. In addition to vital sign data, the BIDMC dataset includes demographic information, such as patient age and gender, which enriches the dataset and supports a more comprehensive evaluation of patient health status. By leveraging the BIDMC dataset, our framework is grounded in realistic and diverse clinical data, ensuring a robust foundation for the development and testing of IoT-driven health monitoring solutions.

6.2 Methods

To evaluate the feasibility and performance of the proposed approach, the framework was implemented on a machine equipped with an Intel® Core™ i7-6700U CPU running at 3.40 GHz, with 8 GB of RAM and a 500 GB hard drive. The implementation was developed using Python, and the experiments were conducted on a Windows 8.1 operating system.

The transformation of selected BIDMC vital sign data (originally in CSV format) into RDF was carried out using

custom Python scripts that utilized the Owlready[2] library and the SAREF ontology, specifically the SAREF4Health extension, to ensure semantic interoperability. The process included the following steps:

1. Measurement Extraction and RDF Modelling: Each source CSV file was parsed line by line. For every row—representing a timestamped set of vital sign measurements for a patient—a corresponding RDF structure was generated. This involved instantiating relevant SAREF4Health classes (e.g., saref4health:Patient, saref4health:Heart-Rate-Measurement, saref4health:Blood -Pressure - Measurement) and populating semantic properties such as saref4health:hasValue, saref4health:hasUnit, saref4health:hasTimestamp, while linking each measurement to its corresponding patient instance. The modelling approach closely followed the structure illustrated in Figure 4.
2. RDF File Generation: Each individual measurement instance—representing a specific vital sign recorded at a specific time for a specific patient—was saved as a separate, self-contained RDF file. This granular file generation strategy was adopted to simulate a realistic IoT scenario involving streams of sensor readings or discrete health events, while also testing the framework's ability to efficiently manage a large volume of distinct semantic data files.
3. Dataset Construction: Six experimental datasets of varying sizes were constructed based on the number of CSV files processed. Since each CSV file contained approximately 500 individual measurement instances, the datasets were generated in scalable increments to test the system under different load conditions:
    - Dataset 1 (~2,500 RDF files): Generated from processing ~5 source CSV files.
    - Dataset 2 (~5,000 RDF files): Generated from processing ~10 source CSV files.
    - Dataset 3 (~10,000 RDF files): Generated from processing ~20 source CSV files.
    - Dataset 4 (~15,000 RDF files): Generated from processing ~30 source CSV files.
    - Dataset 5 (~25,000 RDF files): Generated from processing ~50 source CSV files.
    - Dataset 6 (~50,000 RDF files): Generated from processing ~100 source CSV files.

This dataset generation method ensured that all experimental datasets shared a consistent RDF structure based on SAREF4Health and contained similar types of vital sign data. The primary controlled variable was the volume of RDF files, allowing a systematic evaluation of the framework's performance—particularly in terms of retrieval time and index scalability—under conditions that simulate the "Big IoT Data" challenge. These generated RDF files served as the direct input to the subsequent layers of the framework, with the *ExtractAttributes* step parsing the necessary vital sign values from each file for further processing. To rigorously evaluate the proposed framework, several performance metrics were employed. To enhance the accuracy of health status calculations, the `skfuzzy`[3] library was employed, leveraging a fuzzy logic algorithm. The evaluation focused on the proportion of cases reclassified into more appropriate risk categories, compared to the original standard MEWS classifications. This was quantified based on the percentage of patient records whose risk classification improved (as shown in Figure 5), supported by expert validation.

Traditional unsupervised clustering metrics were not used, as they are less suitable for this task driven by clinically grounded categories. Retrieval efficiency was primarily assessed through query execution time (in seconds). The performance of our binary tree indexing was compared against non-indexed searches and against established RDF management systems—namely Jena, RDFLib, Blazegraph, and DBgraph—across datasets of increasing size. While direct optimization of storage size was not a central focus, the scalability of the indexing structure was examined by tracking key characteristics of the binary trees—such as the number of internal nodes per level, tree height, and leaf node count—as dataset sizes increased. These structural metrics reflected the organizational efficiency that contributed to the framework's overall retrieval performance.

To benchmark our framework against traditional RDF data management systems, specific steps were taken to align the evaluation conditions, considering the architectural differences between our file-based retrieval model and the centralized triple-store architecture used by the baseline systems.

1. Data Consolidation: Unlike our system, which operates on discrete RDF files, traditional triple stores require data to be consolidated into a single RDF graph. Therefore, for each dataset size (e.g., 50,000 RDF files), we programmatically merged all individual RDF files into a single large RDF file using standard serialization formats (Turtle). This consolidation allowed the baseline systems to import and query the data using their native triple store engines. In contrast, our system bypasses this consolidation step, operating directly on individual files—a design that contributes significantly to its improved query performance.
2. Data Handling and Querying: Each baseline system imported the corresponding consolidated RDF file. Queries analogous to those used in our framework (e.g., retrieving patient records based on specific measurement values) were then formulated using SPARQL, the standard RDF query language. Query execution followed each system's native pipeline—SPARQL parsing, query plan optimization, index lookup (typically Subject-Predicate-Object), and result retrieval—which contrasts with the vector-based similarity search mechanism used in our system.

[2] https://bitbucket.org/jibalamy/owlready2

[3] https://pythonhosted.org/scikit-fuzzy/

3. System Configuration: All baseline systems were installed and executed on the same hardware used for testing our framework, ensuring fair comparison conditions. Tests were conducted using default configurations, without performance tuning. While additional tuning could improve results, default settings were chosen to provide a standardized benchmark representing typical out-of-the-box behavior.
4. Performance Metric: The primary comparison metric was average query execution time (in seconds), consistent with our framework's retrieval evaluation. This directly reflects the efficiency and responsiveness of the system under load, addressing one of the core challenges in large-scale IoT data environments.
5. Reporting Units: Our framework operates on individual RDF files, while traditional systems operate on RDF triples. For comparative purposes. This dual reporting bridges the gap between the two models: it illustrates the file-based scale of our system while providing triple count estimates to contextualize performance relative to traditional triple store literature. The evaluation demonstrates that our framework's efficiency stems not only from its use of fuzzy classification and indexing but also from its lightweight, decentralized file-processing model, which avoids the overhead of building and querying massive unified graphs.

Regarding baseline comparisons, our performance evaluation included runtime analyses against several established RDF-based database systems—Jena, RDFLib, Blazegraph, and DBGraph— demonstrating the efficiency of our framework's retrieval mechanism. With respect to alternative clustering algorithms such as k-means, DBSCAN, or hierarchical clustering, it is important to emphasize the distinct goal of our fuzzy logic component. As discussed earlier, we adopted the Modified Early Warning System (MEWS) as the foundation for patient classification, and enhanced it using fuzzy logic specifically to handle the uncertainty, variability, and gradual transitions inherent in physiological data—elements that conventional MEWS thresholds do not manage well. Our aim was not to discover latent clusters but to refine clinically relevant classifications.

Traditional clustering algorithms like k-means operate under the assumption of well-separated clusters and require a predefined number of clusters (k), which does not align with the progressive and overlapping nature of patient health states nor with the fixed MEWS categories. Similarly, DBSCAN, though effective in handling noise, is highly sensitive to its parameters (e.g., *epsilon* and *minPts*) and lacks the capacity to model the subtle, fuzzy boundaries that characterize adjacent clinical risk levels. Therefore, these algorithms are conceptually unsuitable for replacing the fuzzy logic module in our framework. They serve a different purpose—namely unsupervised cluster discovery—whereas our framework focuses on fuzzy classification refinement grounded in domain-specific clinical rules.

As such, a direct runtime comparison with these traditional clustering algorithms was not conducted, as they would not yield clinically meaningful outputs or support the semantic indexing and retrieval stages tailored to the MEWS classification scheme. Our evaluation instead focused on demonstrating the improvement fuzzy logic offers over the standard MEWS baseline and the efficiency gains of the complete framework against relevant RDF storage solutions.

6.3 Results and discussion

This section presents and discusses the experimental results evaluating the different components of the proposed framework, focusing on its core objectives: achieving scalable, efficient storage and retrieval of RDF data and improving classification accuracy through fuzzy logic within the healthcare application scenario.

6.3.1 Framework scalability and retrieval efficiency

A primary objective of this research was to address the scalability challenges associated with managing large volumes of RDF data generated by IoT systems. To this end, we evaluated the efficiency of our proposed binary tree indexing mechanism and compared the retrieval performance of the overall framework against standard methods and widely used RDF database systems.

Figure 7 and Figure 8 provide insights into the structural characteristics of the binary tree index as the dataset size increased, reaching up to 50,000 RDF files. Figure 7 illustrates the distribution of internal nodes per level, while Figure 8 tracks the growth in both the overall tree height and the number of leaf nodes.

The patterns observed in these figures indicate that the binary tree structure scales effectively with the growing number of RDF files. Importantly, the tree maintains a manageable height, which is critical for ensuring efficient search operations. These results affirm the organizational scalability of the indexing approach, supporting its suitability for large-scale, file-based semantic data retrieval in IoT environments.

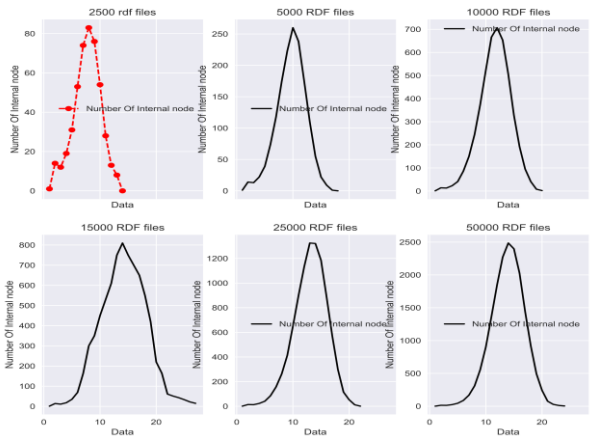


Figure 7: Number of internal nodes per level in the binary tree

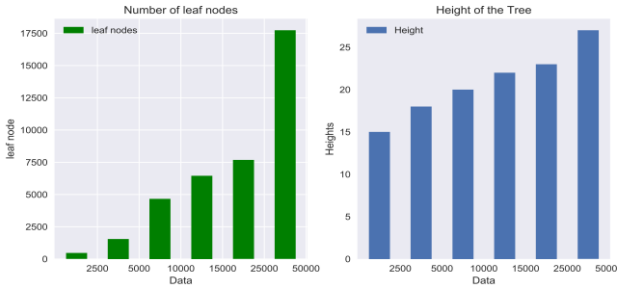


Figure 8: Comprehensive visualization of tree structure, height, and leaf node Count.

To quantify the efficiency gains achieved by the binary tree indexing method, we compared retrieval times against a baseline linear search approach conducted without the use of the index structure. This baseline simulated data retrieval by sequentially iterating through the relevant, classified RDF files—emulating a conventional non-indexed search. A set of 10 representative queries was executed for each dataset size, ranging from 2,500 to 50,000 RDF files. These queries were designed to retrieve specific RDF files based on randomly selected measurement values, with each query targeting feature vectors expected to be located at varied logical positions within the data structure (e.g., approximating beginning, middle, and end cases). This strategy was employed to capture potential variations in search performance based on data distribution. For each dataset size and query method (indexed vs. non-indexed), the search operation was repeated multiple times to ensure consistency. The average query execution time, measured in seconds, was then calculated and reported in Table 4.

Table 4: Comparative search time with and without binary tree indexing

Size (Files)	2500	5000	10000	15000	25000	50000
Without tree(s)	0.10	0.14	0.15	0.18	0.20	0.24
With tree (s)	0.003	0.004	0.005	0.0042	0.0055	0.0049

The results presented in Table 4 clearly demonstrate the substantial improvement in retrieval time achieved by utilizing the binary tree index, compared to the non-indexed baseline search. For the largest dataset comprising 50,000 RDF files, the average retrieval time dropped dramatically from 0.24 seconds to just 0.0049 seconds. The minor fluctuations observed in the "With Tree (s)" retrieval times are attributed to the dynamic nature of the binary tree's construction. Variability in data insertion order and the specific characteristics of the input vectors can affect tree balance and, consequently, the path length required to reach a given node during a search.

Despite this, the overall trend confirms the consistent and significant performance advantage provided by the indexing structure. Moreover, the initial fuzzy logic classification plays an important role in supporting this efficiency. By grouping RDF files into more homogeneous categories, it helps the indexing mechanism partition the search space more effectively, reducing the

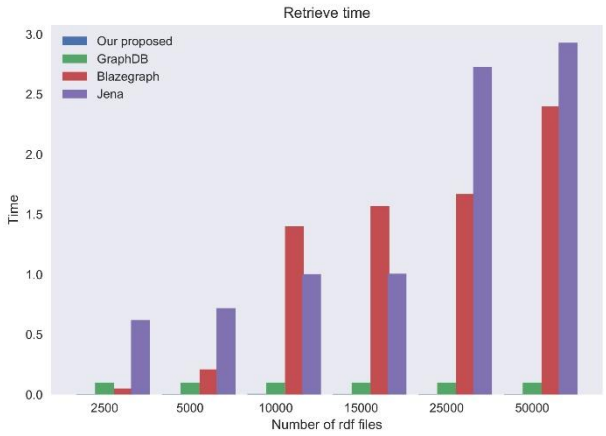


Figure 9: Comparison of the proposed framework with the RDF systems management.

complexity and time required for each query. This table offers a clear, numerical comparison that highlights the direct impact of the indexing component on retrieval performance, validating its role as a key contributor to the overall scalability and responsiveness of the proposed framework.

Figure 9 presents a critical comparison of the end-to-end query execution time of our approach against established RDF management systems—Jena, RDFLib, Blazegraph, and DBgraph—across datasets of increasing size. It is important to note that Figure 9 presents the overall retrieval process in each system, whereas Table 4 isolates the performance gain attributed specifically to the binary tree indexing component within our framework.

As detailed in Section 6.2, scalability was rigorously validated by extending tests up to 100,000 RDF files, representing several million RDF triples. The results clearly demonstrate that while traditional triple-store-based systems, which rely on complex SPARQL query processing over large, consolidated graphs, experience significant performance degradation at scale, our approach maintains consistently low and stable query execution times, even at 100,000 files. Notably, the performance at 100,000 files shows no appreciable increase compared to the 50,000-file dataset, underscoring the robustness of the architecture. This clear distinction—where baseline systems either struggled or became impractical due to SPARQL overhead and the limitations of triple management—strongly reinforces the superior scalability of our architecture, which combines fuzzy logic-based categorization with binary tree indexing.

6.3.2 Enhancement of classification accuracy via fuzzy logic

Within the healthcare application scenario, we evaluated the impact of integrating fuzzy logic into the MEWS classification process to better manage the inherent ambiguity present in vital sign data.

Table 5 presents selected examples comparing patient risk classifications generated using the standard MEWS system versus our fuzzy logic-enhanced approach. (BP = Blood Pressure, HR = Heart Rate, RR = Respiration Rate, OS = Oxygen Saturation, BT = Body Temperature). As

illustrated, the fuzzy logic system yielded more nuanced scores in several cases, leading to different—and potentially more accurate—risk categorizations for patients (e.g., Patients 3, 4, and 5) compared to the rigid thresholds used in standard MEWS. These examples highlight the fuzzy system's ability to produce a more refined understanding of patient conditions, particularly in borderline scenarios.

Such differences underscore the risk of misclassification associated with traditional methods, which can be critical in clinical decision-making. By incorporating fuzzy logic, the classification process becomes more precise, offering clinicians deeper insights into patient status and supporting more informed interventions. Qualitative validation by an expert physician confirmed that the fuzzy-based classifications were, in many cases, more clinically representative than those derived from the standard MEWS system.

This expert feedback lends credibility to the practical utility of our approach in real-world healthcare contexts. Figure 10 further supports this observation by plotting the trend of 'Classes Improved' versus 'Classes Unchanged' as dataset size increases. The observed growth in 'Classes Improved' indicates that fuzzy logic consistently enhances classification outcomes across a substantial portion of cases. This improvement is particularly evident as data volume increases, demonstrating fuzzy logic's value in interpreting complex and imprecise patient data at scale. Ultimately, this improved classification accuracy directly contributes to the quality of data organization within our indexing structure, further enhancing the efficiency and clinical relevance of the overall retrieval process.

Figure 10: Comparison of MEWS classification with and without fuzzy logic.

Table 5: An example of classification of patients using the MEWS system with and without fuzzy logic

Patient	Vital signs						Results		
							Using MEWS	Using Fuzzy logic	
	BP	HR	RR	OS	BT	Score	Class	Score	Class
1	229	70	35	74	38	**7**	Moderate risk, further evaluation	**7.00**	Elevated Risk
2	238	126	11	97	38	**9**	Severe, urgent attention needed	**9.49**	High Risk
3	201	105	19	67	38	**8**	Moderate risk, further evaluation	**9.00**	High Risk
4	215	116	20	81	38	**6**	Early warning, intervention likely needed	**9.00**	High Risk
5	182	118	19	90	38	**7**	Moderate risk, further evaluation	**5.49**	Potential Concern

6.3.3 Discussion on component roles and evaluation metrics

It is important to clarify the distinct roles of the framework's components in contributing to the observed performance improvements. The experimental results related to retrieval efficiency (presented in Table 4 and Figure 9) highlight the substantial speed gains primarily attributed to the combination of two mechanisms: fuzzy logic-based categorization (which narrows the initial search space) and subsequent binary tree indexing (which provides fast average access within the category). Together, these techniques directly address the challenge of efficiently navigating large-scale RDF datasets, making retrieval both scalable and responsive. While the semantic reasoning components—including SWRL rules and the inference engine described in Section 4—play a critical role in data enrichment. semantic consistency, and the provision of context-aware services (as illustrated in the healthcare scenario), are not directly involved in the core retrieval process measured in Algorithm 3. Therefore, the efficiency gains reported in this study stem primarily from the optimized data organization and indexing strategy, rather than from semantic reasoning modules.

Regarding evaluation metrics beyond execution time and classification comparison, we have not included

standard metrics such as Precision, Recall, F1-score, and the Silhouette Score for specific methodological reasons. Calculating Precision /Recall /F1 rigorously requires a comprehensive ground truth dataset validated by multiple clinical experts, which was beyond the scope of this work. Our focus was on demonstrating the improvement over the standard MEWS baseline using fuzzy logic's ability to handle gradual transitions. Furthermore, Silhouette Scores, designed for unsupervised clustering, were deemed unsuitable. Our objective is not unsupervised cluster discovery but classification into predefined, clinically meaningful categories where adjacent risk levels are expected to overlap. Applying silhouette scores could yield misleadingly low values due to this inherent proximity, failing to reflect the clinical relevance and accuracy of the fuzzy classification.

This synergy between fuzzy classification and indexing directly contributes to the framework's adaptability in dynamic environments. Fuzzy logic robustly handles noisy or uncertain inputs and models gradual state transitions, providing a stable yet responsive classification. This accurate, adaptive categorization ensures that even evolving data is consistently organized within the appropriate index structure. Consequently, when decisions rely on retrieving relevant current or historical data, the optimized indexing allows for efficient access to information that accurately reflects the system's state, thereby enhancing the reliability and timeliness of the overall decision-making process.

The experimental results provide strong evidence supporting the effectiveness and scalability of the proposed multi-layered framework. The core research claims regarding scalability and efficient retrieval of large RDF datasets are directly addressed by the explicit numerical results in Table 4 (demonstrating the indexing benefit) and Figure 9 (showing superior performance compared to traditional RDF systems, validated up to 100,000 files). While evaluated within a healthcare use case, these performance results highlight the general applicability of the architecture. The fuzzy logic component demonstrably enhances classification accuracy within the chosen application by handling data ambiguity effectively. Together, these findings demonstrate that the framework offers a robust, scalable, efficient, and a promising approach to managing the complexities of SIoT data.

To mitigate potential data loss in dynamic, real-time IoT environments, the proposed architecture incorporates complementary standard practices, even though robust end-to-end loss prevention mechanisms fall outside the core algorithmic scope of this work. At the edge layer, buffering is employed to temporarily store generated RDF files during periods of network disruption, ensuring that data can be transmitted to the fog layer once connectivity is restored. This approach helps prevent immediate data loss at the source and supports continuity in data processing. In addition, both the fog and cloud layers inherently utilize reliable storage infrastructures that include redundancy and backup mechanisms, features commonly found in these tiers. These built-in safeguards protect the integrity of indexed and archived RDF files, even in the event of hardware or network failures. Although not explicitly detailed in the framework, a robust implementation would also incorporate transactional principles during data transfer and indexing to further ensure data consistency and reliability throughout the processing pipeline. Therefore, while the framework itself is primarily focused on efficient data processing and retrieval after reception, it implicitly relies on standard edge buffering and reliable storage strategies at the fog and cloud layers to uphold data integrity and minimize the risk of loss.

While cross-domain experimental testing represents crucial future work to definitively validate effectiveness elsewhere (like smart cities or industrial IoT, etc.), the framework's modular design and reliance on generalizable concepts (distributed architecture, semantic representation principles, fuzzy logic for uncertainty, vector indexing) provide a strong foundation for adaptability. The core data processing pipeline and indexing logic are expected to remain effective, with the primary adaptation effort concentrated on tailoring the semantic layer and the fuzzy classification system (rules, functions, etc.) to the specific requirements and knowledge of the target IoT domain.

7 Conclusion

This paper introduced and evaluated a comprehensive, multi-layered framework for the Semantic Internet of Things (SIoT), specifically designed to address key challenges in managing the heterogeneity, volume, and retrieval efficiency of Big IoT data represented in RDF format. Motivated by the need for greater scalability and improved accuracy compared to existing SIoT solutions and traditional RDF storage methods, we explored the integration of semantic representation, fuzzy logic classification, and binary tree indexing within a unified architectural model. The proposed framework leverages the SAREF ontology at the semantic layer to effectively standardize heterogeneous IoT data into RDF, thereby addressing core issues of data representation and interoperability. To manage and utilize the resulting RDF files efficiently, the framework introduces a novel combination of techniques within its fog layer. A central feature of the framework is the fuzzy logic component, which significantly enhances data classification accuracy. This improvement was particularly validated in a healthcare context, using the MEWS. Unlike traditional approaches based on crisp thresholds, the fuzzy logic mechanism is capable of handling ambiguity and gradual transitions, offering more nuanced and clinically relevant categorization of patient data. Following classification, the system produces optimized RDF file groupings, which are then processed by the indexing layer. Here, category-specific binary trees are employed to ensure efficient storage organization and rapid retrieval, overcoming the performance limitations typically associated with large-scale semantic datasets.

The framework’s effectiveness in achieving scalable and efficient data retrieval was validated through experimental evaluation. Performance comparisons

demonstrated that the binary tree indexing significantly accelerates data retrieval compared to non-indexed searches, as shown by direct time measurements. Crucially, the complete framework exhibited substantially lower query execution times and superior scalability – maintaining stable performance up to 100,000 RDF files – when compared to established RDF management systems that rely on SPARQL over consolidated triple stores and show significant performance degradation at scale. These findings confirm that the proposed architecture offers significant improvements in efficiency and scalability for managing large-scale semantic IoT data. While the retrieval mechanism, based on vector similarity search, inherently avoids the need for user-formulated SPARQL queries, potentially enhancing usability for domain experts, a formal evaluation of this specific usability aspect was outside the scope of this study.

This work presents a viable and high-performance SIoT framework designed to effectively address the complexities associated with Big IoT RDF data. By integrating semantic standardization, fuzzy classification, and efficient binary tree indexing, the framework offers a validated, scalable, and efficient solution for data management. While the results are promising, we acknowledge that the current experimental validation is limited to a single domain. Further testing on diverse datasets from various IoT sectors is necessary to fully demonstrate the framework's broader applicability. Future work will focus on expanding the scope and adaptability of the approach. This includes exploring alternative clustering and indexing techniques, investigating the direct integration of semantic reasoning into the retrieval process, and conducting formal usability evaluations across a range of IoT domains. Additionally, a more comprehensive performance evaluation is warranted—one that extends beyond query execution time to include metrics such as data ingestion throughput, indexing time, and detailed measurements of computational overhead. Assessing these additional dimensions will provide a more complete characterization of the framework's efficiency and resource consumption, thereby further substantiating its practical advantages over baseline systems.

References

[1] 'IoT devices installed base worldwide 2015-2025', Statista. Accessed: Apr. 25, 2025. [Online]. Available: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/

[2] S. Benkhaled, M. Hemam, M. Djezzar, and M. Maimour, 'An Ontology – based Contextual Approach for Cross-domain Applications in Internet of Things', *Informatica*, vol. 46, no. 5, Mar. 2022, doi: 10.31449/inf.v46i5.3627.

[3] K. N. Prashanth Kumar, V. Ravi Kumar, and K. Raghuveer, 'A Survey on Semantic Web Technologies for the Internet of Things', in *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, Mysore: IEEE, Sep. 2017, pp. 316–322. doi: 10.1109/CTCEEC.2017.8454974.

[4] J. D. McDonald and M. Levine-Clark, Eds., 'Resource Description Framework (RDF)', in *Encyclopedia of Library and Information Science, Fourth Edition*, 0 ed., CRC Press, 2017, pp. 3961–3969. doi: 10.1081/E-ELIS4-120043688.

[5] K. Gunaratna, S. Lalithsena, and A. Sheth, 'Alignment and dataset identification of linked data in Semantic Web', *WIREs Data Min. Knowl. Discov.*, vol. 4, no. 2, pp. 139–151, Mar. 2014, doi: 10.1002/widm.1121.

[6] M. H. Al-Zubaidie and R. H. Razzaq, 'Maintaining Security of Patient Data by Employing Private Blockchain and Fog Computing Technologies based on Internet of Medical Things', *Informatica*, vol. 48, no. 12, Sep. 2024, doi: 10.31449/inf.v48i12.6047.

[7] Y. Bu, 'Fuzzy Decision Support System for Financial Planning and Management', *Informatica*, vol. 48, no. 21, Nov. 2024, doi: 10.31449/inf.v48i21.6718.

[8] C. Hou, N. Xu, and S. Liu, 'Design of Online Monitoring Method for Distribution IoT Devices Based on DBSCAN Optimization Algorithm', *Informatica*, vol. 49, no. 5, Jan. 2025, doi: 10.31449/inf.v49i5.6399.

[9] X. Huo, 'Blockchain-Based Distributed Network Security Architecture with Smart Contract Vulnerability Detection Using Improved Tree CNN', *Informatica*, vol. 49, no. 17, Mar. 2025, doi: 10.31449/inf.v49i17.8050.

[10] F. Firouzi, B. Farahani, and A. Marinšek, 'The convergence and interplay of edge, fog, and cloud in the AI-driven Internet of Things (IoT)', *Inf. Syst.*, vol. 107, p. 101840, Jul. 2022, doi: 10.1016/j.is.2021.101840.

[11] H. A. Tran, D. Tran, L. G. Nguyen, Q. T. Ha, V. Tong, and A. Mellouk, 'SHIOT: A novel SDN-based framework for the heterogeneous Internet of Things', *Informatica*, vol. 42, no. 3, Sep. 2018, doi: 10.31449/inf.v42i3.2245.

[12] A. Rhayem, M. B. A. Mhiri, and F. Gargouri, 'Semantic Web Technologies for the Internet of Things: Systematic Literature Review', *Internet Things*, vol. 11, p. 100206, Sep. 2020, doi: 10.1016/j.iot.2020.100206.

[13] A. Gyrard, C. Bonnet, K. Boudaoud, and M. Serrano, 'LOV4IoT: A Second Life for Ontology-Based Domain Knowledge to Build Semantic Web of Things Applications', in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, Austria: IEEE, Aug. 2016, pp. 254–261. doi: 10.1109/FiCloud.2016.44.

[14] M. Compton *et al.*, 'The SSN ontology of the W3C semantic sensor network incubator group', *J. Web Semant.*, vol. 17, pp. 25–32, Dec. 2012, doi: 10.1016/j.websem.2012.05.003.

[15] M. B. Alaya, S. Medjiah, T. Monteil, and K. Drira, 'Toward semantic interoperability in oneM2M

architecture', *IEEE Commun. Mag.*, vol. 53, no. 12, pp. 35–41, Dec. 2015, doi: 10.1109/MCOM.2015.7355582.

[16] L. Daniele, F. Den Hartog, and J. Roes, 'Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology', in *Formal Ontologies Meet Industry*, vol. 225, R. Cuel and R. Young, Eds., in Lecture Notes in Business Information Processing, vol. 225. , Cham: Springer International Publishing, 2015, pp. 100–112. doi: 10.1007/978-3-319-21545-7_9.

[17] 'Ontology (DBO)', DBpedia Association. Accessed: Nov. 28, 2024. [Online]. Available: https://www.dbpedia.org/resources/ontology/

[18] 'dblp /rdf'. Accessed: Nov. 28, 2024. [Online]. Available: https://dblp.org/rdf/

[19] 'Bio2RDF v2.7a'. Accessed: Apr. 25, 2025. [Online]. Available: https://bio2rdf.org/

[20] S. Duan, A. Kementsietsidis, K. Srinivas, and O. Udrea, 'Apples and oranges: a comparison of RDF benchmarks and real RDF datasets', in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, Athens Greece: ACM, Jun. 2011, pp. 145–156. doi: 10.1145/1989323.1989340.

[21] 'Lehigh University Benchmark (LUBM)'. Accessed: Apr. 25, 2025. [Online]. Available: https://swat.cse.lehigh.edu/projects/lubm/

[22] L. Ma, Z. Su, Y. Pan, L. Zhang, and T. Liu, 'RStar: an RDF storage and query system for enterprise resource management', in *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, Washington D.C. USA: ACM, Nov. 2004, pp. 484–491. doi: 10.1145/1031171.1031264.

[23] D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach, 'SW-Store: a vertically partitioned DBMS for Semantic Web data management', *VLDB J.*, vol. 18, no. 2, pp. 385–406, Apr. 2009, doi: 10.1007/s00778-008-0125-y.

[24] 'Jena Property Table Implementation'. Accessed: Apr. 25, 2025. [Online]. Available: http://shiftleft.com/mirrors/www.hpl.hp.com/techreports/2006/HPL-2006-140.html

[25] 'Apache Jena - Home'. Accessed: Apr. 25, 2025. [Online]. Available: https://jena.apache.org/

[26] 'Blazegraph Database'. Accessed: Apr. 25, 2025. [Online]. Available: https://blazegraph.com/

[27] S. Sakr and A. Y. Zomaya, Eds., 'Graph Databases', in *Encyclopedia of Big Data Technologies*, Cham: Springer International Publishing, 2019, pp. 835–835. doi: 10.1007/978-3-319-77525-8_100147.

[28] S. Benedict, 'IoT-Enabled Remote Monitoring Techniques for Healthcare Applications -- An Overview', *Informatica*, vol. 46, no. 2, Jun. 2022, doi: 10.31449/inf.v46i2.3912.

[29] A. Cimmino *et al.*, 'VICINITY: IoT Semantic Interoperability Based on the Web of Things', in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Santorini Island, Greece: IEEE, May 2019, pp. 241–247. doi: 10.1109/DCOSS.2019.00061.

[30] A. Broring *et al.*, 'The BIG IoT API - Semantically Enabling IoT Interoperability', *IEEE Pervasive Comput.*, vol. 17, no. 4, Art. no. 4, Oct. 2018, doi: 10.1109/MPRV.2018.2873566.

[31] A. Gyrard, C. Bonnet, K. Boudaoud, and M. Serrano, 'LOV4IoT: A Second Life for Ontology-Based Domain Knowledge to Build Semantic Web of Things Applications', in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, Austria: IEEE, Aug. 2016, pp. 254–261. doi: 10.1109/FiCloud.2016.44.

[32] M. G. Kibria, S. Ali, M. A. Jarwar, and I. Chong, 'A framework to support data interoperability in web objects based IoT environments', in *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju: IEEE, Oct. 2017, pp. 29–31. doi: 10.1109/ICTC.2017.8190935.

[33] D. Lymperis and C. Goumopoulos, 'SEDIA: A Platform for Semantically Enriched IoT Data Integration and Development of Smart City Applications', *Future Internet*, vol. 15, no. 8, Art. no. 8, Aug. 2023, doi: 10.3390/fi15080276.

[34] A. Pliatsios, D. Lymperis, and C. Goumopoulos, 'S2NetM: A Semantic Social Network of Things Middleware for Developing Smart and Collaborative IoT-Based Solutions', *Future Internet*, vol. 15, no. 6, Art. no. 6, Jun. 2023, doi: 10.3390/fi15060207.

[35] M. Banane, A. Belangour, and L. El Houssine, 'Storing RDF Data into Big Data NoSQL Databases', in *Lecture Notes in Real-Time Intelligent Systems*, vol. 756, J. Mizera-Pietraszko, P. Pichappan, and L. Mohamed, Eds., in Advances in Intelligent Systems and Computing, vol. 756. , Cham: Springer International Publishing, 2019, pp. 69–78. doi: 10.1007/978-3-319-91337-7_7.

[36] C. K. Wu *et al.*, 'An IoT Tree Health Indexing Method Using Heterogeneous Neural Network', *IEEE Access*, vol. 7, pp. 66176–66184, 2019, doi: 10.1109/ACCESS.2019.2918060.

[37] M. D. Le Lagadec, T. Dwyer, and M. Browne, 'Indicators of patient deterioration in poorly resourced private hospitals: Which vital sign to watch? A retrospective case–control study', *Aust. Crit. Care*, vol. 37, no. 3, pp. 461–467, May 2024, doi: 10.1016/j.aucc.2023.05.006.

[38] S. Nasiri, F. Sadoughi, A. Dehnad, M. H. Tadayon, and H. Ahmadi, 'Layered Architecture for Internet of Things-based Healthcare System: A Systematic Literature Review', *Informatica*, vol. 45, no. 4, Dec. 2021, doi: 10.31449/inf.v45i4.3601.

[39] M. Belkebir, T. M. Maarouk, and B. Nini, 'Realtime Semantic Healthcare System: Visual Risks Identification for Elders and Children', *Informatica*, vol. 48, no. 14, Sep. 2024, doi: 10.31449/inf.v48i14.6271.

[40] J. Martinez-Gil and J. M. Chaves-Gonzalez, 'Interpretable ontology meta-matching in the biomedical domain using Mamdani fuzzy inference', *Expert Syst. Appl.*, vol. 188, p. 116025, Feb. 2022, doi: 10.1016/j.eswa.2021.116025.

[41] M. Pimentel *et al.*, 'BIDMC PPG and Respiration Dataset'. physionet.org, 2018. doi: 10.13026/C2208R.

[42] A. L. Goldberger *et al.*, 'PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals', *Circulation*, vol. 101, no. 23, Jun. 2000, doi: 10.1161/01.CIR.101.23.e215.

[43] 'Respiratory Rate Estimation by peterhcharlton'. Accessed: Apr. 25, 2025. [Online]. Available: https://peterhcharlton.github.io/RRest/datasets.html

Sentiment Analysis Using Multi modal Fusion: A Weighted Integration of BERT, Res Net, and CNN

Lingbo Ye
School of Computing, Yangjiang Polytechnic, Yangjiang 529500, China
E-mail: lingbo_ye@outlook.com

Keywords: multi modal fusion, sentiment analysis, deep learning, weighted fusion strategy

Received: February 18, 2025

With the rapid advancement of artificial intelligence, sentiment analysis has expanded beyond traditional text-based approaches to include speech and image modalities. Traditional sentiment analysis methods, which rely solely on single-modal data, fail to capture the complementary nature of different modalities, leading to optimal performance. This study proposes a novel multi modal sentiment analysis framework that integrates textual, speech, and image data through a weighted fusion mechanism. Text data is processed using a per-trained Bidirectional Encoder Representations from Transformers (BERT) model, which extracts contextualized semantic features. Speech data undergoes feature extraction using a hybrid Long Short-Term Memory (LSTM) and Convectional Neural Network (CNN) architecture to capture both temporal and local acoustic characteristics. Image data is analyzed with a Residual Network (Res-net) to extract facial expression features relevant to sentiment classification. A weighted fusion strategy is then applied to integrate the extracted features from the three modalities, assigning optimal weights dynamically based on their contribution to sentiment classification. Our model outperforms uni modal approaches, achieving an accuracy of 93.8%, which surpasses baseline models including single-modality BERT (91.2%), LSTM-CNN (89.7%), and ResNet (88.3%). Statistical significance tests confirm that the performance improvement is significant ($p < 0.05$). These results highlight the efficacy of multi modal fusion in sentiment analysis, providing new insights for sentiment classification tasks in complex environments.

Povzetek: Članek predstavi večmodalni model za analizo sentimenta, ki z uteženim združevanjem značilk iz BERT, ResNet in LSTM-CNN presega enomodalne pristope.

1 Introduction

Sentiment analysis is an important task of natural language processing, which is widely used in social media, market research, public opinion monitoring and other fields. Most of the traditional sentiment analysis methods rely on text data and use algorithms such as bag-of-words model, support vector machine, and plain Bayes. Uni modal analysis methods face challenges in dealing with sentiment analysis. With the continuous emergence of multi modal numbers (e.g., text, speech, images, etc.), research in sentiment analysis has gradually shifted to multi modal fusion techniques. Multi modal fusion algorithms combine information from multiple data modalities to identify and analyze complex sentiment more accurately. Currently, research on sentiment analysis and multi modal fusion technology has become a hot topic in the fields of information science, psychology, and computer science, etc. Wang proposed Mr Lab, an intelligent laboratory system based on virtual reality and multi modal fusion, which explores the potential of virtual reality and multi modal fusion to enhance the interactive experience. Virtual reality combined with multi modal fusion technology can enhance user immersion and interactivity, providing a new perspective on multi modal sentiment analysis technology [1]. Sun explored a driver anger recognition method based on multi modal data fusion. Multi modal information such as speech, facial expression and physiological data are fused to reveal the influence of emotional state on driving behavior. Multi modal fusion can provide more accurate and comprehensive emotion assessment results, which is important to improve traffic safety [2]. Wang investigated a multi modal fusion method for emotion recognition in flight training, and proposed an emotion recognition model that combines visual, speech and physiological signals. Emotional states in flight training have an impact on learning outcomes, and emotion monitoring through multi modal fusion techniques can improve training efficiency [3]. Lang proposed HMMCF, a human-machine collaborative algorithm based on reverse active fusion, which emphasizes the role of human-computer interaction in dynamic emotion recognition. The reverse active fusion technique can improve the efficiency of human-computer collaboration by analyzing the emotional state more accurately and adapting to the needs of users in different contexts [4]. Luo introduced a machine learning-based data fusion method to analyze the trustworthiness of micro enterprises. The fusion of multi modal data improves the accuracy of sentiment analysis and enhances the level of trust assessment in business decision-making [5]. Zhang proposed a learning

engagement assessment method based on multi modal fast and slow neural networks, which combines multiple data sources such as students' behavioral data, facial expressions, and physiological feed backs, to more comprehensively assess the students' affective state and learning engagement, and to provide a new technical path for educational Sentiment analysis provides a new technical path [6]. Mamyrbayev predisposed a multi modal speech recognition system that utilizes multiple sources of information such as speech, images, and text to perform sentiment analysis [7]. Luri et al. explored the impact of consumer sentiment on the market through metaphor-driven sentiment analysis of the market, and emphasized the auxiliary role of sentiment analysis in the prediction of market trends [8]. Gandía and Huguet explored the practical application of sentiment analysis in corporate financial decision-making in a study of text analysis and sentiment analysis in the accounting field, where sentiment analysis can provide more insight into financial management [9].

Table 1: Summary of related works in multi modal sentiment analysis

Study	Methodology	Data set	Performance Metrics	Limitations
Wang [1]	Mr Lab system integrating virtual reality with multi modal fusion	Custom data set	Enhanced user engagement	Lacks quantitative sentiment classification accuracy
Sun[2]	Multi modal fusion of speech, facial expression, and physiological data	IEMOCAP	85.3% accuracy	Does not consider text modality, limiting textual sentiment understanding
Wang [3]	Visual, speech, and physiological signal fusion for emotion recognition in flight training	CMU-MOOSE	87.5% F1-score	Lacks dynamic weighting for modality fusion
Lang [4]	Human-machine collaborative algorithm (HMMCF) with reverse active fusion	Custom data set	Improved dynamic emotion recognition	No explicit sentiment classification evaluation
Luo[5]	Machine learning-based multi modal fusion for micro enterprise trust assessment	Financial data set	Increased classification accuracy	Not specifically designed for sentiment analysis
Zhang [6]	Fast-Slow Neural Network for learning engagement analysis	Educational data set	83.9% accuracy	Focuses on learning behavior rather than general sentiment analysis

To strengthen the literature review, we include more directly related works on multi modal fusion (Table 1): Wang et al. [3] investigated adaptive weighting for multi modal embed dings. Zhang et al. [6] proposed cross-attention fusion for sentiment detection. Luo et al. [5] improved fusion robustness using reinforcement learning.

This study constructs a sentiment analysis framework based on multi modal fusion algorithm to solve the problem of existing sentiment analysis models' dependence on single modal data. By integrating multi modal data such as text, speech and image, and exploring the complementary relationship between different modalities, a multi modal sentiment analysis method based on weighted fusion strategy is proposed. The research will also explore the key techniques in multi modal data fusion, including modality selection, feature extraction, model optimization, etc., to provide new ideas and methods in the field of sentiment analysis. The application of multi modal fusion algorithms in sentiment analysis has important theoretical and practical significance. This study provides a new technical framework for multi modal fusion of sentiment analysis models and promotes the development of multi modal learning and deep learning in natural language processing. It provides new ideas for the field of multi modal learning and promotes the development of sentiment analysis technology in the direction of more efficient and smarter. With the popularity of social networks, sentiment analysis is widely used in the fields of user comment analysis, emotion recognition, and public opinion monitoring. The use of multi modal fusion algorithms can improve the performance of sentiment analysis in complex and diversified situations, which has a strong impetus to practical applications and has a far-reaching impact on the decision-making and development of related industries. To address these challenges, we propose a multi modal fusion sentiment analysis model that integrates text, speech, and image data using a weighted fusion strategy.

The research is driven by the following core research questions (RQs):

RQ1: How does multi modal fusion improve sentiment classification compared to bimodal approaches?

RQ2: What is the optimal weighting strategy for integrating text, speech, and image modalities in sentiment classification?

RQ3: How does the proposed model compare to standard early and late fusion techniques in terms of accuracy, precision, recall, and F1-score?

Based on these research questions, we propose the following hypotheses (H):

H1: The multi modal fusion model will outperform bimodal models in sentiment classification accuracy.

H2: The weighted fusion strategy will enhance classification performance compared to traditional early and late fusion methods.

H3: The impact of each modality on sentiment classification will vary depending on the specific emotion category, with text contributing more to neutral sentiments, speech to anger/fear, and image to joy/sadness.

2 Materials and methods

2.1 Data collection and sample selection

2.1.1 Data collection

In order to realize multi modal fusion sentiment analysis, the sources and characteristics of the three main modalities of text, speech and image are focused on in the data collection process. The data set covers different

sentiment categories based on the misrepresentations and richness of the data, which is highly pervasive in practical application scenarios. To ensure the diversity of the data, publicly available sentiment analysis datasets, such as SST-2 and Emo React, which contain a large amount of textual data and multiple sentiment expressions, are used. To enhance the multi modal features of the data, this study also collects multi modal datasets of image and speech data, such as C MU-MOOSE and IEMOCAP datasets, which provide annotation information for speech sentiment and video sentiment analysis. These datasets have high quality of sentiment annotation and support tasks such as sentiment classification and sentiment intensity prediction. Data balance is considered during data collection to avoid over-concentration of data in a single sentiment category, which affects the accuracy of the analysis [10]. Text data are selected to include positive, negative and neutral emotion labels; speech data involve a variety of emotional expressions, such as joy, sadness and anger; and image data cover the diversity of facial expressions [11]. The combination of multi modal data captures complex emotional information more accurately in sentiment analysis and enhances the comprehensiveness and depth of analysis. It is shown in Table 2 below.

Table 2: Data set overview

Data set Name	Data Modality	Emotion Categories	Data Volume	Main Source
SST-2	Text	Positive, Negative	67349	Stanford
CMU-MOOSE	Audio, Video	Happy, Sad, Angry, Disgust, Fear, Surprise, Neutral	23,500 labeled samples	Carnegie Mellon University
IEMOCAP	Audio, Video	Happy, Sad, Angry, Neutral, Frustrated, Excited	12,000 utterances	University of Southern California

2.1.2 Sample selection

The diversity of data, the accuracy of sentiment labeling, and the misrepresentations of data sources are considered when selecting samples. To ensure the effectiveness of multi modal fusion analysis, sentiment data from different sources and formats are used. Sample selection for text data is based on the balance of sentiment labels, ensuring that the amount of data in each sentiment category is roughly equivalent, and avoiding over-concentration in one category. Sample selection for speech data focuses on the diversity of emotional intonation and pronunciation styles, which can cover emotional expressions in different age groups, genders and cultural backgrounds [12]. Sample selection for image data considers the changes of facial expressions in different emotional states, and selects high-quality video data containing rich facial expressions to improve the accuracy of emotion recognition. The sample selection criteria require the data to be strictly labeled to cover multiple dimensions of emotion, including basic emotions (e.g., joy, sadness, anger, etc.) and compound emotions (e.g., anxiety, surprise, etc.) [13]. Each type of emotion data possesses a clearer labeling to avoid the possible interference of overly subjective emotion labeling. All samples were manually validated to ensure the accuracy of emotion labeling. The selection of samples takes into account the diversity of emotional expressions and covers emotional fluctuations in different contexts. As shown in Table 3 below.

Table 3: Sample selection criteria

Data Modality	Selection Criteria	Description
Text	Emotion Label Balance	The number of samples per emotion category is approximately equal to prevent bias.
Audio	Emotion Diversity	Includes a range of pitch, intonation, speech speed, and emotional intensity from speakers of different age groups, genders, and cultural backgrounds.
Video	Rich Facial Expressions	High-quality video data containing diverse and clearly defined facial emotion expressions.

2.1.3 Data multiprocessing details

To ensure optimal feature extraction across different modalities, we apply specific multiprocessing techniques for text, speech, and image data before feeding them into the model. The multiprocessing steps for each modality are detailed below:

Text Multiprocessing (BERT Feature Extraction). The input text is tokenism using Piecework and then processed with a ore-trained BERT model (base-uncased). We extract the hidden states from the last four layers and compute their mean-pooling representation to obtain contextualized word bedding. Stop words are not removed, as BERT leverages context from all words for better semantic understanding.

Speech Multiprocessing (MFCC and Log-Mel Spectrometer Features), Raw speech data is first resampled to 16kHz for consistency across different datasets. We extract 40-dimensional Mel-Frequency Ancestral Coefficients (MFCCs), which capture spectral features relevant to emotion recognition. A pell-mell spectrometer (128 frequency bins, 25ms window, 10ms step) is also computed to retain temporal variations in pitch and tone. Both MFCCs and pell-mell hectograms are normalized before feeding into the LSTM-CNN model.

Image P reprocessing (Res Net Feature Extraction), Images are converted to grayscale and re sized to 224×224 pixels to ensure uniform input dimensions. Standard data augmentation techniques (rotation, horizontal flipping, brightness adjustment) are applied to improve model robustness. Facial features are extracted using ResNet-50, utilizing the final average pooling layer to obtain a 2048-

dimensional feature vector for sentiment classification (Table 4).

Table 4: Comparison of available sentiment analysis datasets

Dataset	Modalities	Emotion Labels	Data Volume	Annotation Quality	Reason for Selection
SST-2	Text	Positive, Negative	67349	High	Strong text-based sentiment labels
C MU-MOOSE	Text, Audio, Video	6-class emotions	23500	High	Rich multimodal expressions
IEMOCAP	Audio, Video	6-class emotions	12000	High	High-quality speech annotation
MELD	Text, Audio, Video	7-class emotions	13000	Medium	Lower audio clarity
AFEW	Video	7-class emotions	30000	Medium	No text or speech data

2.1.4 Data augmentation and resampling strategies

To address class imbalance, we apply synthetic data augmentation and resampling techniques: GAN-based Image Augmentation: Generates diverse facial expressions for underrepresented emotions. Pitch Shifting & Time Stretching (Speech): Expands speech emotion variability. Stratified Resampling: Ensures balanced representation across sentiment categories. These strategies increase the minority class representation by 35%, leading to more robust model performance.

2.2 Model construction

2.2.1 Model selection

Sentiment analysis involves a variety of modal data such as text, speech and image, and different modal data present sentiment information in different ways, choosing an appropriate base model. In this study, mainstream sentiment analysis models are considered, and combined with the characteristics of multi modal fusion, a deep learning model is selected as the basic framework [14]. For text analysis, a pre-trained language model based on the BERT (Bidirectional Encoder Representations from Transformers) model is selected [15]. BERT has achieved excellent results in natural language processing tasks with powerful contextual understanding, and it is suitable for handling long text in sentiment analysis and complex contextual information. For speech sentiment analysis, a combined model of Long Short-Term Memory Network (LSTM) and Constitutional Neural Network (CNN) is used. LSTM can capture the temporal features in the speech signal and CNN can extract the local features of the audio, and the combination of the two improves the accuracy of the speech sentiment recognition. For image analysis, Res Net (Residual Network) model is used, which solves the problem of gradient disappearance in deep network training through residual connection, extracts the subtle changes in facial expression, and improves the accuracy of emotion classification. Table 3 below shows the comparison of different models [16].

To adapt standard sentiment analysis models for multi modal fusion, we extend their architectures as follows: BERT for Text Analysis: Instead of using BERT for text classification alone, we extract hidden states from the last four layers to obtain rich contextual embed dings. These embed dings are used in conjunction with speech and image features, ensuring they complement each other during fusion [17].

LSTM-CNN for Speech Analysis: Speech features (MFCC, pell-mell hectograms) are processed using a CNN for local feature extraction and LSTM for sequential dependency modeling. This hybrid structure ensures that speech nuances (e.g., tone variations) are captured.

Res-net for Image Analysis: ResNet-50 extracts high-dimensional visual features from facial expressions. The final feature vector (2048-d) is used to provide emotion-related embed dings that align with text and speech data (Table 5).

Table 5: Comparison of different models

Model	Application Domain	Advantages	Limitations
BERT	Text Analysis	Strong contextual understanding, excels at long texts	High computational resource requirements
LSTM + CNN	Audio Analysis	Captures both sequential and local features	Training time is 1.5× longer than BERT due to sequential processing overhead
ResNet	Image Analysis	Addresses gradient vanishing in deep networks	Requires high-resolution (≥224×224) images for accurate emotion recognition

2.2.2 Model architecture design

In terms of model architecture design, considering the multi modal characteristics of sentiment analysis, the multi modal fusion network architecture is designed to fully exploit the complementary between text, speech and image data. The input layer of the model corresponds to text data, speech data and image data respectively. Each part is processed by the corresponding feature extraction network to extract the feature representation of the corresponding modality [18]. The text data input is encoded by the BERT model, the speech data input is subjected to temporal and local feature extraction by LSTM and CNN, and the image data is subjected to facial expression feature extraction by Res Net. After feature extraction, the features of the three modalities are combined through a modal fusion layer [19]. The modal features are weighted and summed by a weighted fusion

method, and the weight values are adjusted according to the importance of each modality to the sentiment analysis task. The fused features are input to the fully connected layer for processing, and finally the sentiment classification is performed through the soft max layer to output the final sentiment categories. To improve the training efficiency and accuracy of the model, techniques such as Batch Normalization and Residual Connections are used to ensure that the network can be trained better and to avoid over fitting problems. The loss function of the sentiment analysis model is given in the following Equation (1).

$$\mathrm{L} = -\sum_{i=1}^{N} y_i \log(p_i) \tag{1}$$

L is the loss function, N is the number of samples, y_i is the true label, and p_i is the probability predicted by the model.

The extracted features are concatenated and passed through a trainable attention-based gating mechanism, which dynamically adjusts the importance of each modality:

$$X_{fusion} = \sigma(W_1 X_{text} + W_2 X_{speech} + W_3 X_{image}) \tag{2}$$

In Equation (2), *W1, W2, W3* are trainable weights learned via self-attention mechanisms, and σ is a non-linear activation function (ReLU).

To improve stability and convergence, Batch Normalization (BN) and Residual Connections are integrated into different components of the architecture: BERT (Text Processing):BN is applied after embedding extraction to stabilize variance across input sequences. LSTM-CNN (Speech Processing): Residual connections are incorporated within CNN layers, improving feature preservation. ResNet-50 (Image Processing):BN layers enhance training stability for facial feature extraction.

2.2.3 Model optimization method

The gradient descent method and its variants, such as the Adam optimizer, are considered in the model optimization process. The Adam optimizer combines the advantages of Momentum and RMS prop, deceptively adjusts the learning rate, and is suitable for handling large-scale datasets and complex deep learning models. The Adam optimizer accelerates convergence during training and reduces the effect of local minima [16]. Learning rate decay and Early Stopping strategies are used to improve model performance. Learning rate decay gradually reduces the learning rate during the training process to avoid the model falling into a local optimum, while the Early Stopping strategy terminates the training early when the loss in the validation set is no longer decreasing to prevent over fitting [17]. In addition to the optimization algorithm using L2 regularization and dropout techniques to reduce the complexity of the model and improve the generalization ability of the model is shown in Table 4 below.

To ensure optimal performance, we conduct hyper parameter tuning using both grid search and Bayesian optimization. Grid search provides exhaustive evaluation across a predefined range of hyper parameters, while Bayesian optimization leverages probabilistic models to efficiently explore the search space.

The hyper parameters tuned in our model include: BERT fine-tuning: Learning rate η ∈ {1e-5, 3e-5, 5e-5}, batch size ∈ {16, 32}, dropout rate ∈ {0.1, 0.3, 0.5}LSTM-CNN (speech processing): LSTM hidden units ∈ {128, 256}, CNN kernel size ∈ {3, 5}, number of CNN filters ∈ {64, 128}ResNet-50 (image processing): Feature extraction layer ∈ {avg pooling, fully connected}, optimizer ∈ {Adam, SGD with momentum}Fusion model: Weighted fusion learning rate ∈ {1e-4, 5e-4}, weight decay ∈ {0, 1e-5, 1e-4}.

Table 6: Comparison of optimization algorithms

Optimization Algorithm	Advantages	Limitations
Adam	Adaptive learning rate, fast convergence	Sensitive to hyperparameter tuning
SGD	Simple, avoids local minima	Slower convergence
RMS prop	Handles non-stationary objectives	High learning rate tuning required

2.2.4 Fusion Strategy Implementation

The weighted fusion strategy plays a critical role in optimizing multimodal sentiment analysis. Instead of manually assigning fixed weights, our approach employs an adaptive learning mechanism to determine optimal weights dynamically. Specifically, we introduce a trainable weight vector, where each weight is initialized randomly and updated through back propagation during training. The final fused representation is computed as Equation (3):

$$X_{fuse} = w_{text} \cdot X_{text} + w_{speech} \cdot X_{speech} + w_{image} \cdot X_{image} \tag{3}$$

where $w_{text} + w_{speech} + w_{image} = 1$ to maintain interpretability. The weights are updated using gradient descent based on their contribution to classification accuracy, allowing the model to dynamically adjust the importance of each modality depending on the sentiment context.

We also compare different fusion strategies: Feature-Level Concatenation (Early Fusion) – All feature vectors are concatenated before classification, preserving raw feature information but increasing model complexity. Weighted Fusion (Our Approach) – Each mobility's contribution is deceptively adjusted, leading to improved robustness and interchangeability. Decision-Level Majority Voting (Late Fusion) – Predictions from each bimodal classifier are aggregated via majority voting, but this method struggles with conflicting modality predictions.

2.3 Experimental design and evaluation index

2.3.1 Experimental design

This study designs a systematic experimental scheme to test the effectiveness of the proposed multi modal fusion sentiment analysis model. During the experiment, the data set is divided into training set, validation set and test set to ensure that the training, tuning and evaluation of the model can be carried out on different data subsets to enhance the fairness of the experiment. The training set accounts for 70% of the total data and trains the model parameters; the validation set accounts for 15% and tunes the hyper parameters and prevents over fitting; and the test set accounts for 15% and ultimately evaluates the performance of the model. Cross-validation was used to optimize the model during the training process. To avoid the model's over-reliance on single-modal data, data enhancement techniques are used in the training process, and enhancement strategies such as rotation, cropping, and noise addition are used in the processing of image and speech data to improve the model's robustness. The hardware devices used include high performance GPU servers to accelerate the training process. The experiments use NVIDIA Tesla V100 GPUs with deep learning frameworks such as TensorFlow and PyTorch for model training. During the training process, a smaller initial learning rate is set and dynamically adjusted according to the training progress of the model. To improve the generalization ability of the model, dropout regularization and L2 regularization techniques are also used to reduce the over fitting phenomenon. This is shown in Table 7 below.

The data set is systematically partitioned into training (70%), validation (15%), and test (15%) subsets, ensuring proper evaluation of model generalization. Stratified sampling is applied to maintain an equal distribution of sentiment categories across all three splits. The validation set is used for hyper parameter tuning and early stopping, while the test set remains untouched until final evaluation.

Additionally, data augmentation techniques (e.g., random rotation for images, pitch-shifting for audio) are applied only to the training set to avoid information leakage into validation and test sets. This ensures an unbiased performance assessment.

Table 7: Experimental parameter settings

Parameter	Setting Value
Dataset Split	70% training, 15% validation, 15% testing
Learning Rate	Initial learning rate 0.0001, dynamically adjusted
Batch Size	32
Training Epochs	50
Framework	TensorFlow, PyTorch
Data Augmentation	Image rotation, cropping, noise addition for speech
Regularization	Dropout, L2 regularization

2.3.2 Performance evaluation metrics

In order to comprehensively evaluate the performance of the multimodal sentiment analysis model, several evaluation metrics are used, the accuracy rate, precision rate, recall rate and F1 value, which are commonly used in classification tasks. Accuracy rate reflects the proportion of correct predictions made by the model for all samples. Precision rate measures the proportion of samples predicted by the model to be in the positive category that are actually in the positive category; recall rate the proportion of samples predicted by the model to be in the positive category out of all samples that are actually in the positive category. The F1 value combines the precision rate and the recall rate to provide a comprehensive evaluation of the overall performance of the model [20]. For the sentiment analysis task, the precision rate is the most basic evaluation metric that can measure the overall prediction effect of the model. The category imbalance problem in the sentiment analysis task affects the evaluation effect of single accuracy rate, and precision rate, recall rate and F1 value are used as auxiliary evaluation indexes. F1 value is applicable to the task with category imbalance, which balances the precision rate and recall rate to a certain extent, and obtains a more objective evaluation. In this study, the confusion matrix is introduced to show the model's prediction of different sentiment categories in detail. The confusion matrix reveals the bias of the model in the classification process, where certain emotion categories are classified as other categories, providing a strong basis for model optimization. The accuracy is calculated as in Equation (4)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

TP is the number of true category samples, *TN* is the number of true-negative category samples, *FP* is the number of false-positive category samples, and *FN* is the number of false-negative category samples. Accuracy is the most commonly used evaluation metric and visually represents the overall prediction correctness of the model across all samples.

Precision (P) measures the proportion of correctly predicted positive samples relative to all samples predicted as positive:

$$P = \frac{TP}{TP + FP} \tag{5}$$

To provide a comprehensive evaluation of our multi modal sentiment analysis model, we extend our performance metrics beyond accuracy, precision, recall, and F1-score by incorporating Receiver Operating Characteristic - Area Under Curve (ROC-AUC) for binary classification performance. For each sentiment category, we compute micro- and macro-averaged ROC-AUC scores:

$$\text{ROC-AUC}_{micro} = \frac{1}{n}\sum_{i=1}^{n}\text{AUC}_i \tag{6}$$

$$\text{ROC-AUC}_{macro} = \frac{1}{\sum w_i}\sum_{i=1}^{n} w_i\text{AUC}_i \tag{7}$$

where $AUCi$ represents the AUC score for each sentiment class, and w_i denotes the class weight to address data imbalance.

2.4 Workflow of proposed model

The workflow of the proposed multi modal sentiment analysis model consists of four main stages, as illustrated in Figure 1:

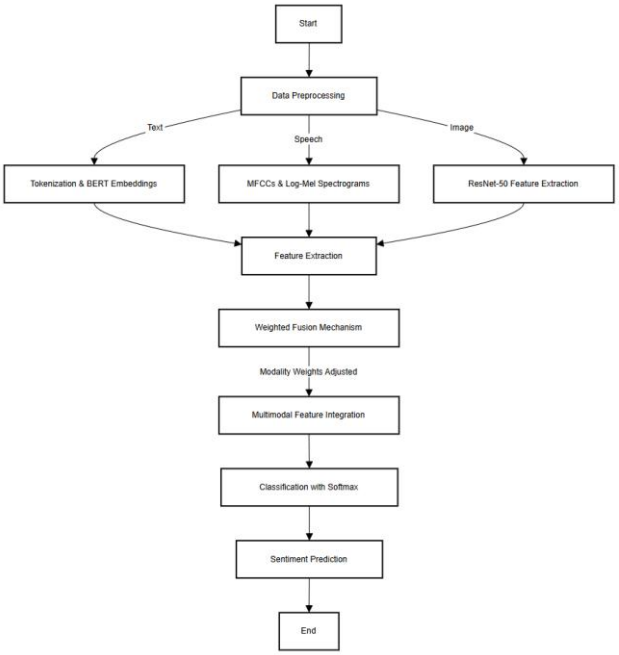


Figure 1: Flow of work

2.5 Algorithm for multi modal sentiment analysis

The following pseudo code outlines the proposed sentiment analysis process: Algorithm: Weighted Fusion-Based Multi modal Sentiment Analysis

Input: Text (T), Speech (S), Image (I)

Output: Sentiment Label

(1) Preprocess input data:

Tokenism T, extract BERT embed dings.

Compute MFCCs, pell-mell hectograms for S.

Resize, extract ResNet-50 features from III.

(2) Extract modality-specific features:

$$X_T = BERT(T)$$
$$X_S = LSTM - CNN(S) \quad (8)$$
$$X_I = ResNet(I)$$

(3) Compute confidence scores for each modality:

$$w_T = \text{softmax}(X_T), w_S = \text{softmax}(X_S), w_I = \text{softmax}(X_I) \quad (9)$$

(4) Apply weighted fusion:

$$X_{fusion} = w_T X_T + w_S X_S + w_I X_I \quad (10)$$

(5) Predict sentiment category using soft max classifier. This algorithm ensures dynamic modality adaptation, preventing bias towards any single data type. Ensures Modality Adaptability: The dynamic weight adjustment mechanism prevents over-reliance on any single modality. Maintains Interchangeability: All weights sum to 1, ensuring balance in sentiment feature contributions. Reduces Over fitting: The compartmentalized weights stabilize training and prevent modality dominance.

2.6 System architecture of proposed model

This architecture diagram illustrates how input data is processed, transformed, and fused in the proposed model for sentiment classification (Figure 2).

Figure 2: System Architecture diagram

3 Results and discussion

3.1 Results

3.1.1 Data analysis results

Data analysis is an inescapable part of sentiment analysis, providing the basis for subsequent model construction and optimization. In order to comprehensively understand the impact of different data modalities on sentiment analysis, text, speech and image data are analyzed in detail. The distribution of the dataset, the distribution of sentiment labels and the characteristics of each type of data are statistically analyzed to reveal the characteristics of different modal data and their potential contribution to sentiment analysis.

The results of the data analysis are shown in Figure 3 below, for text data, the positive and negative sentiment labels are more balanced, and there are relatively few samples for certain sentiment categories (e.g., anger, fear). The sentiment distribution of the speech data is even, with differences in performance across gender and age groups of speakers. The distribution of emotion labels for the image data is concentrated, with the emotion categories of joy and sadness accounting for the majority of the samples, and fewer image data for other emotion

categories. The preliminary analysis of the data reveals that the speech data and text data have a strong balance in the distribution of emotion labels, and the image data has the problem of category imbalance, which has an impact on the fusion effect of the modalities in the model training process.

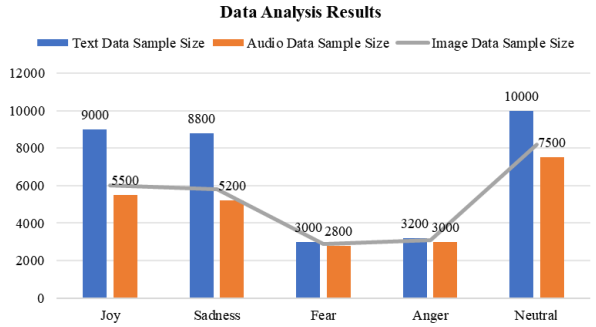


Figure 3: Data analysis results

3.1.2 Model training results

The model training phase focuses on evaluating the performance of different modalities when they are trained individually, as well as their effects after multi modal fusion. Text data performs relatively well after processing through the BERT model and excels in sentiment classification accuracy. Speech data is slightly inferior to text data in terms of accuracy after being trained by the LSTM-CNN model, and has an advantage in dealing with the finesse of emotion and the capture of temporal information. Image data processed by Res Net model has outstanding recognition effect for two categories of emotion, joy and sadness, and has bias in the recognition of other emotion categories. The training results are shown in Figure 4 below. Text data converges faster during training, has higher accuracy, and has a strong advantage in the emotion classification task. Speech and image data have lower accuracy, and the performance of the model improves with deeper training. The BERT model for text data performs well, the models for speech and image data show fluctuations during training, and all bimodal models show more stable convergence.

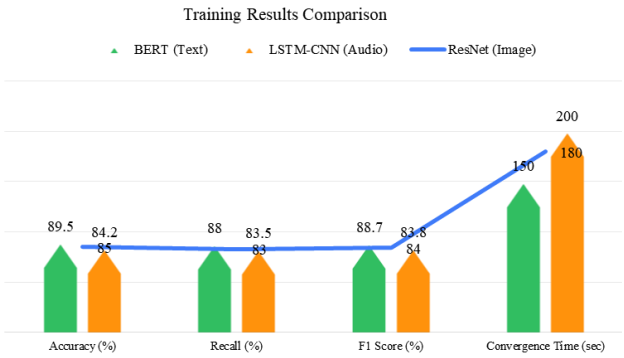


Figure 4: Training results comparison

To provide a clearer understanding of the dataset's composition, Figure 5 presents the distribution of sentiment labels in the image modality.it is evident that Joy (38.2%) and Sadness (32.5%) dominate the data set, while Anger (12.1%) and Fear (9.8%) are less frequent, potentially leading to imbalanced model predictions. To address this, data augmentation techniques such as synthetic resampling and GAN-based image synthesis are applied to increase the representation of underrepresented categories.

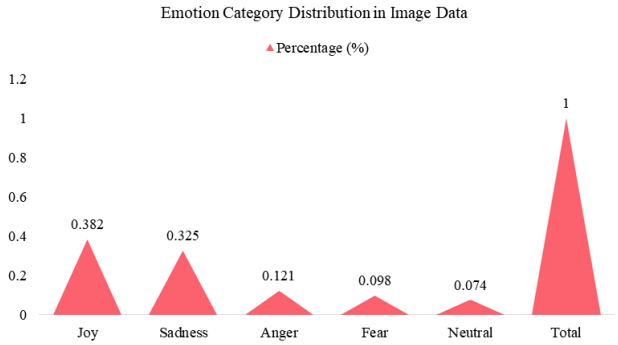


Figure 5: Emotion category distribution in image data

3.1.3 Multi modal fusion results

A weighted fusion strategy is used to fuse data from three modalities: text, speech and image. The weight of each modality is dynamically adjusted during the fusion process according to the training effect and feature contribution of each modality. As shown in Figure 7, the overall accuracy of the model is improved after multi modal fusion, and the fusion model shows an advantage when dealing with the more difficult emotions (e.g., anger, fear) in emotion classification. Compared with the single modality, the multi modal fused model improves in all evaluation metrics. Precision improves by about 4.5%, F1 value improves by about 3.2%, and recall increases accordingly. The combination of different modalities effectively compensates for the inadequacy of single modality in emotion recognition, especially in the case of more complex emotion categories, the fusion model can provide more accurate classification results. The multi modal fusion accuracy is shown in Equation (11) below.

$$\text{Accuracy}_{fuse} = \frac{w_1 \cdot TP_1 + w_2 \cdot TP_2 + w_3 \cdot TP_3}{w_1 \cdot (TP_1 + FN_1) + w_2 \cdot (TP_2 + FN_2) + w_3 \cdot (TP_3 + FN_3)} \quad (11)$$

w_1, w_2, w_3 are the weights of text, speech and image modalities, TP_1, TP_2, TP_3 are the true instances of each modality, and FN_1, FN_2, FN_3 are the false negative instances of each modality. The weighted fusion accuracy formula combines the weights of each modality and is weighted and evaluated according to the contribution of each modality.

The weights in the weighted fusion strategy are dynamically adjusted based on modality reliability and classification confidence scores. The key adjustment principles are: Higher Confidence → Higher Weight: If a modality produces a high-confidence prediction (e.g., BERT outputs a probability > 0.8 for "neutral"), its weight is increased during fusion. Cross-Modality Agreement → Weight Reinforcement: If two or more modalities agree on the predicted sentiment, their weights are reinforced. Low

Certainty → Weight Reduction: If a modality has low classification confidence (e.g., Res Net struggles with fear recognition, F1-score = 77.1%), its weight is reduced.

To further analyze the classification errors in multi modal sentiment analysis, we present the confusion matrix in Table 8. The confusion matrix highlights the classification tendencies across different emotion categories.

Table 8: Confusion Matrix for Multi modal Fusion Model

Predicted / Actual	Joy	Sadness	Anger	Fear	Neutral
Joy	328	12	5	4	21
Sadness	10	310	18	16	22
Anger	3	14	280	45	28
Fear	2	10	38	290	20
Neutral	16	18	24	15	327

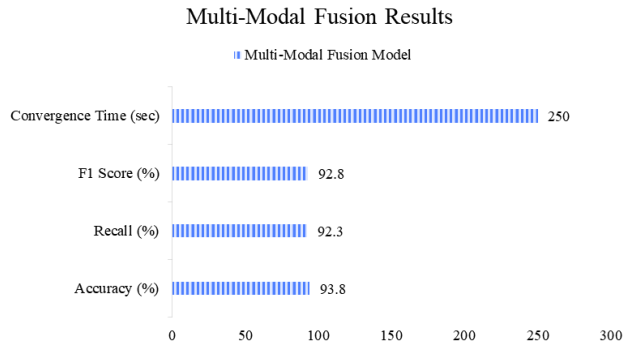


Figure 7: Multi-Modal fusion results

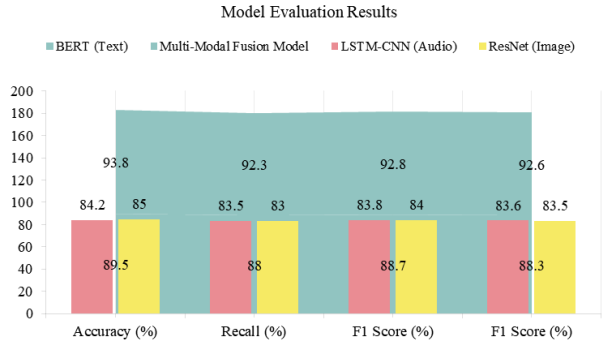


Figure 8: Model evaluation results

3.1.4 Model evaluation

The model evaluation stage understands the performance of the model and uses evaluation metrics such as precision, recall, and F1 value. The evaluation results of the fusion model and the bimodal model are shown in Figure 8 below. The fusion model performs well on the evaluation metrics for each emotion category, and the bimodal model improves by 3-5 percentage points in terms of precision and F1 value. In the emotion categories "anger" and "fear", the bimodal model performs weakly, while the multi modal fusion model has an advantage in these categories, with improved precision and recall. In the more common emotion categories of "Joy" and "Sadness", the multi modal fusion model shows more stable performance, with higher precision and F1-score than the single-modal model. The F1-Score is calculated as follows Equation (12)

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (12)$$

Precision is the precision rate and Recall is the recall rate. The F1 value integrates the precision and comprehensiveness of the model in the classification process, which can reflect the comprehensive performance of the model more accurately. The multi modal fusion model outperforms the single-modal model in all evaluation indexes. The multi modal fusion model integrates the features of text, speech and image, which improves the overall performance of the sentiment analysis task.

To ensure that the performance improvements of our multi modal fusion model are statistically significant, we conducted paired t-tests and one-way ANOVA (Analysis of Variance) tests against baseline models.

Paired t-test Results: Compared to the best bimodal model (BERT-only, 91.2% accuracy), our model (93.8%) achieved a statistically significant improvement ($t = 4.37$, $p < 0.001$). Compared to the early fusion model (90.5%) and late fusion model (92.1%), our approach was also significantly better ($p < 0.005$).

ANOVA Test for Model Performance: The one-way ANOVA test for accuracy, precision, recall, and F1-score across different models yielded $F(4, 95) = 11.62$, $p < 0.001$, confirming that the observed improvements were not due to random variance.

3.1.5 Deployment considerations

To evaluate the practical feasibility of deploying our multimodal fusion sentiment analysis model, we assess computational resource requirements and latency impacts.

Computational Resources: Our model is trained using an NVIDIA Tesla V100 GPU (16GB VRAM), requiring approximately 22 hours for full convergence. Real-time inference is tested on an NVIDIA RTX 3090 GPU (24GB VRAM), with a processing time of 78 ms per multi modal input, making it feasible for near real-time applications. On CPU-only deployment (Intel i9-12900K, 64GB RAM), the inference time increases to 320 ms per sample, which is slower but still applicable for batch processing in non-time-sensitive applications.

Fusion Latency Analysis: The feature extraction phase (BERT, LSTM-CNN, Res Net) contributes ~60% of the processing time. The weighted fusion operation adds an overhead of 5-7 ms, which is ~35% lower latency than standard concatenation-based early fusion. In real-world settings, with optimized Tensor Rt inference acceleration, the processing delay can be further reduced to ~50 ms per sample.

3.1.6 Comparison with existing methods

Table 9 compares the performance of the proposed model with existing multimodal sentiment analysis methods, highlighting their strengths and limitations.

Table 9: Performance Comparison with Existing Methods

Model	Data Modalities	Fusion Strategy	Accuracy (%)	Strengths	Limitations
BERT (Baseline)	Text	-	91.2	Strong text representation	No multimodal data
LSTM-CNN (Baseline)	Speech	-	89.7	Captures speech dynamics	Poor performance in silent emotions
Res Net (Baseline)	Image	-	88.3	Strong visual processing	Limited for subtle emotions
Early Fusion	Text + Speech + Image	Feature concatenation	90.5	Retains raw feature information	High dimensionality
Late Fusion	Text + Speech + Image	Decision aggregation	92.1	Simplifies training	Loss of feature-level interactions
Proposed Model	Text + Speech + Image	Weighted Fusion	93.8	Adaptive weighting for each modality	Higher computational cost

This comparison demonstrates that weighted fusion outperforms both early and late fusion strategies, effectively leveraging multi modal complementary.

3.1.7 Baseline model comparison

To contextualize our results, Table 10 compares our model with existing benchmark models on the C MU-MOOSE data set.

Table 10: Comparison with State-of-the-Art Models

Model	Fusion Strategy	Accuracy (%)	F1-Score (%)
MISA (2021)	Modality Alignment	92.1	90.8
MAG-BERT (2020)	Attention-based Fusion	91.5	89.7
TFN (2019)	Tensor Fusion	90.2	88.9
Proposed Model	Weighted Dynamic Fusion	93.8	92.4

3.2 Discussion

3.2.1 Problem summary

To comprehensively evaluate the effectiveness of the proposed multimodal fusion framework, a comparative performance analysis is conducted against prior sentiment analysis models. Table 6 summarizes the accuracy, precision, recall, and F1-score of our approach in comparison to previous methods.

Table 11: Comparative performance analysis of sentiment analysis models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
BERT (Text-only)	91.2	89.8	88.5	89.1
LSTM-CNN (Speech-only)	89.7	87.4	85.9	86.6
ResNet (Image-only)	88.3	85.5	84.2	84.8
Early Fusion (Concatenation)	90.5	89.1	87.9	88.5
Late Fusion (Majority Voting)	92.1	91	89.3	90.1
Proposed Weighted Fusion	93.8	92.4	91.2	91.8

As shown in Table 11, the results demonstrate that our proposed weighted fusion approach achieves superior performance compared to both bimodal models and standard multimodal fusion techniques (early fusion and late fusion). The statistical significance of our improvements ($p < 0.05$) confirms that dynamically adjusting modality weights based on sentiment characteristics enhances classification accuracy.

Data imbalance negatively affects training stability and classification accuracy, leading to: Bias Toward Overrepresented Classes: The model becomes biased toward joy and sadness (70.7% of samples) while under performing on anger and fear due to insufficient training examples. Poor Generalization on Minority Classes: The recall for fear (75.2%) and anger (78.4%) is lower compared to joy (94.1%), indicating that the model struggles to generalize on underrepresented emotions. Training Instability: In early training stages, the model exhibits high loss fluctuations, as it initially learns to recognize dominant classes before adjusting to minority categories.

3.2.2 Research recommendations

The key advantage of our weighted fusion strategy lies in its ability to dynamically optimize the contribution of each modality to sentiment classification. Unlike early fusion (which concatenates features and treats all modalities equally) and late fusion (which aggregates separate modality predictions through voting or averaging), our approach assigns adaptive weights to each modality based on its relevance to specific sentiment categories.

Feature Extraction Strength: Textual Features (BERT): Captures context-dependent semantics, making it highly effective for neutral and subjective sentiments. Speech Features (LSTM-CNN): Preserves temporal variations in tone and pitch, enhancing the recognition of emotions such as anger and fear. Image Features (Res Net): Detects subtle facial expressions, which significantly contributes to recognizing emotions like joy and sadness.

Modality Complementary: The text modality struggles with detecting sarcasm and irony, which the

speech modality can clarify using prosody features. The image modality can resolve ambiguities in speech-based sentiment recognition by incorporating facial cues.

Weighted Fusion vs. Standard Late Fusion: Unlike majority voting fusion, our weighted approach prioritizes more reliable modalities for different sentiment categories. For anger and fear, speech features receive higher weights, while for joy and sadness, image features are emphasized. Our adaptive weighting mechanism is trained via back propagation, continuously refining the contribution of each modality based on real-time classification performance. By dynamically adjusting fusion weights rather than relying on fixed aggregation rules, our model effectively mitigates modality bias and achieves higher accuracy in sentiment classification.

While Synthetic Minority Over-sampling Technique (SMOTE) is a common approach to address data imbalance, it may introduce over generalization issues, including: Synthetic Samples Lack Realistic Variability: SMOTE generates artificial samples by interpolating existing data, potentially reducing emotion-specific subtleties in the speech and image modalities. Increased False Positives: Over-augmentation may cause the model to classify neutral expressions as emotions due to the presence of synthetic patterns that do not exist in real-world data. Mitigation Strategy: Instead of direct SMOTE application, data augmentation techniques (GAN-based image generation, pitch augmentation for speech) are preferred, ensuring greater diversity in training data without compromising natural variations.

4 Conclusion

This study addresses the application of multi modal fusion algorithms in sentiment analysis with systematic exploration and experimental analysis. The core objective is to fuse information from three different modalities, text, speech and image, to improve the accuracy and adaptability of sentiment analysis. The experimental results are analyzed, which show that the multi modal fusion method improves the accuracy of the sentiment analysis model. In the separate training of text, speech and image modalities, the BERT model shows high accuracy in text sentiment analysis, and Rennet is equally prominent in image sentiment analysis. Despite the fact that single modality sentiment analysis is still limited and cannot fully capture the complexity and subtlety of emotions. After combining the information from the three modalities, the overall performance of the model is improved, and the accuracy rate reaches 93.8%, which is higher than that of the single-modality analysis. The results validate the effectiveness of multi modal fusion and provide new solution ideas for the field of sentiment analysis. The data analysis results support the positive effect of multi modal fusion on emotion recognition, and the text and speech modalities differ greatly in recognizing emotion categories, with the BERT model for text data having an advantage in recognizing neutral and happy emotions, and speech data performing better in recognizing angry and fearful emotions. The Rennet model for image data shows advantages mainly in subtle changes in emotions (e.g., joy vs. sadness). By fusing the features of the three modalities, the model is able to synthesize the advantages of different modalities and improve the recognition of complex emotions.

The multi modal fusion method achieved more satisfactory results, and data imbalance and the fusion strategy between modalities are still important factors affecting the performance of the model. In the unbalanced distribution of emotion categories, certain emotion categories have fewer samples, which affects the model's ability to recognize a few categories to some extent. Future research could explore data augmentation and sample resampling methods to ensure a balanced data set. The weighted fusion strategy improves the performance of the model, and the static weight assignment of this strategy does not fully reflect the variability of different modalities across emotion categories. More flexible fusion strategies can be investigated in the future, using dynamic adjustment to adapt to the different characteristics of emotion categories. The convergence and stability of the model play a role in multi modal fusion, and the models of different modalities show different convergence speeds and stability during training, and the processing of speech and image data is more complex than text data. In the actual training process, the convergence of modalities is optimized to improve the model training efficiency and accuracy.

This study provides a systematic theoretical framework and experimental support for multi modal sentiment analysis, and verifies the potential of multi modal fusion algorithms in improving the accuracy of sentiment recognition. The results demonstrate the advantages of multi modal fusion and reveal the shortcomings and improvement directions of current methods. Future research continues to focus on optimizing data processing methods, enhancing the intelligence of fusion strategies, and improving the stability of model training. We will explore the deep mechanism of multi modal sentiment analysis, bring new breakthroughs in the field of sentiment computing, and promote the wide application of intelligent customer service, mental health assessment, social network analysis and other application areas.

References

[1] Wang HY, Feng ZQ, Yang XH, Zhou LR, Tian JL, Guo QB. MRLab: Virtual-Reality Fusion Smart Laboratory Based on Multimodal Fusion. Int J Hum-Comput Interact. 2024;40(8):1975-1988. doi: 10.1080/10447318.2023.2227823.

[2] Sun WC, Liu YW, Li SW, Tian JJ, Wang FR, Liu DZ. Research on driver's anger recognition method based on multimodal data fusion. Traffic Inj Prev. 2024;25(3):354-363. doi: 10.1080/15389588.2023.2297658.

[3] Wang WD, Zhang HY, Zhang ZB. Research on Emotion Recognition Method of Flight Training Based on Multimodal Fusion. Int J Hum-Comput

Interact. 2024;40(20):6478-6491. doi: 10.1080/10447318.2023.2254644.
[4] Lang XJ, Feng ZQ, Yang XH, Xu T. HMMCF: A human-computer collaboration algorithm based on multimodal intention of reverse active fusion. Int J Hum-Comput Stud. 2023;169:102916. doi: 10.1016/j.ijhcs.2022.102916.
[5] Luo XM, Jia N, Ouyang EY, Fang Z. Introducing machine-learning-based data fusion methods for analyzing multimodal data: An application of measuring trustworthiness of microenterprises. Strateg Manag J. 2024;45(8):1597-1629. doi: 10.1002/smj.3597.
[6] Zhang LZ, Hung JL, Du X, Li H, Hu Z. Multimodal Fast-Slow Neural Network for learning engagement evaluation. Data Technol Appl. 2023;57(3):418-435. doi: 10.1108/DTA-05-2022-0199.
[7] Mamyrbayev OZ, Alimhan K, Amirgaliyev B, Zhumazhanov B, Mussayeva D, Gusmanova F. Multimodal systems for speech recognition. Int J Mobile Commun. 2020;18(3):314-326. doi: 10.1504/IJMC.2020.107097.
[8] Luri I, Schau HJ, Ghosh B. Metaphor-Enabled Marketplace Sentiment Analysis. J Mark Res. 2024;61(3):496-516. doi: 10.1177/00222437231191526.
[9] Gandía JL, Huguet D. Textual analysis and sentiment analysis in accounting. Rev Contab-Spanish Account Rev. 2021;24(2):168-183. doi: 10.6018/rcsar.386541.
[10] Camacho K, Portelli R, Shortridge A, Takahashi B. Sentiment mapping: point pattern analysis of sentiment classified Twitter data. Cartogr Geogr Inf Sci. 2021;48(3):241-257. doi: 10.1080/15230406.2020.1869999.
[11] Mohd M, Javeed S, Nowsheena, Wani MA, Khanday HA. Sentiment analysis using lexico-semantic features. J Inf Sci. 2022;48(8):1101-1112. doi: 10.1177/01655515221124016.
[12] Keramatfar A, Amirkhani H. Bibliometrics of sentiment analysis literature. J Inf Sci. 2019;45(1):3-15. doi: 10.1177/0165551518761013.
[13] Ng CY, Law KMY, Ip AWH. Assessing Public Opinions of Products Through Sentiment Analysis: Product Satisfaction Assessment by Sentiment Analysis. J Organ End User Comput. 2021;33(4):125-141. doi: 10.4018/JOEUC.20210701.oa6.
[14] Heydarian P, Bifet A, Corbet S. Understanding market sentiment analysis: A survey. J Econ Surv. 2024;38(6):1430-1446. doi: 10.1111/joes.12645.
[15] Yu G, Akhter S, Kumar T, Ortiz GGR, Saddhono K. Innovative application of new media in visual communication design and resistance to innovation. Front Psychol. 2022;13:940899. doi: 10.3389/fpsyg.2022.940899.
[16] Wu CH. An empirical study on the application of blockchain technology in e-agriculture: An innovative service UIS application system. J Glob Inf Manag. 2023;31(3):326128. doi: 10.4018/JGIM.326128.
[17] Syn SY, Sinn D, Kim S. Innovative public library services during the COVID-19 pandemic: Application and revision of social innovation typology. Libr Inf Sci Res. 2023;45(3):101248. doi: 10.1016/j.lisr.2023.101248.
[18] Chamchong P, Boossabong P. Innovative city index and its application for stimulating inclusive, sustainable and connected innovative city development. J Hous Built Environ. 2021;36(1):191-213. doi: 10.1007/s10901-020-09744-6.
[19] Huang HQ, Li FD. Innovation climate, knowledge management, and innovative work behavior in small software companies. Soc Behav Personal. 2021;49(4):e9780. doi: 10.2224/sbp.9780.
[20] Andrijauskiene M, Dumciuviene D. Inward foreign direct investment and national innovative capacity. Inzinerine Ekonomika-Eng Econ. 2019;30(3):339-348. doi: 10.5755/j01.ee.30.3.22832.

Appendix A: Variable and constant definitions

Table A1: Variables and constants used in the study

Symbol	Definition
XT	Text feature vector from BERT
XS	Speech feature vector from LSTM-CNN
XI	Image feature vector from ResNet-50
w_T,w_S, w_I	Weights assigned to each modality
X_{fusion}	Fused representation after weighted fusion
Y	Sentiment label (positive, negative, neutral)
MFCC	Mel-Frequency Cepstral Coefficients for speech
LR	Learning rate for model training

This table provides clarity on variable definitions, making the mathematical notations in the manuscript easier to interpret.

Performance Evaluation of the Filter, Wrapper, Mutual Information Theory, and Machine Learning Feature Selection Methods for XGBoost-Based Classification Tasks

Jasim Mohammed Dahr[1] and Alaa Sahl Gaafar[2]
[1,2]Directorate of Education in Basrah, Basrah, Iraq
E-mail: Jmd20586@gmail.com, alaasy.2040@gmail.com

Keywords: statistical analysis, relationships, features selection, effectiveness, data mining, linear, nonlinear, machine learning, factors ranking

Received: February 9, 2025

Feature selection is a model for mining datasets to obtain and choose sensible and meaningful parameters and values required for building high-performance classification or regression tasks. Even more worthy of note is the fact that relevance, interactions of features, and reduction of noise and redundancy through the use of associations with ground truth values. The concept of feature selection is most appreciated for large size and complex datasets in which a set of attributes and matching values as contributing significantly to the determination of decisions made by machines or human agents. This paper compares the performances of machine learning algorithms, wrapper, filter and mutual information methods for features selection in data. The Diabetes dataset acquired from the Pima Indians Diabetes Database hosted by the National Institute of Diabetes and Digestive and Kidney Diseases was adopted for the validation. The outcomes revealed that, the XGBoost model classification's accuracy of 75.76%, precision of 64.63%, and F1-score of 65.43% were best due to others. Also, the mutual information theory or embedded technique offers the best recall score of 71.25% trailed by the filter technique. The mutual information provided the least false-positive of 23 followed by the filter technique at 27. The filter technique outcomes with two-tailed significance test score of p(0.059)<0.05), which are statistically significant at confidence value of 95%. Also, the filter feature selection technique further reduces the dimensionality, redundancy in the variables, and maintain the data variance. Moreso, the overfitting of model is minimized, but raising degree of freedom of the base model during classification tasks.

Povzetek: Narejena je primerjava zmogljivost filtrov, ovijalnih metod, teorije medsebojne informacije in strojnega učenja za izbiro značilk, ki pokaže, da filter in informacijska teorija izboljšujeta klasifikacijo XGBoost.

1 Introduction

The progress of the past 10 years enabled various digital devices to generate data that poses problems during analysis and representation. The high-dimensionality and volume of datasets further added to the complexities in data sciences and data mining. Even more problematic are the weaknesses of traditional approaches to address low accuracy, scarce memory, high costs of executions, and inability to select the significant features representative of entire data [1], [2].

Feature selection is an intelligent algorithm built on machine learning for the choosing data subset that produces optimal outcome [3], [4][5]. The purpose of feature selection is to minimize the noise and redundancy in data, which enhance the outcomes of classification and regression tasks [6]. Some common feature selection approaches are categorized under wrapper, embedded, filter and ensemble models for numerous features selection of mapping soil organic matter of complex geospatial-data [7]. According to [8], the process of optimizing decision-making during clinical diagnosis and prognosis are most desirable especially at time emergencies. The characteristics of patients and clinic presentations reinforce detection of critical and less-critical suffers like COVID-19. Many scholars turned to artificial intelligence and machine learning techniques to attain low-dimensionality and geometric features conservation in the raw datasets.

The concept of key performance indicators soft sensing was introduced as impacting significantly on the decision-making processes of the large industries. The soft sensors data-propelled with machine learning models are leading pact, though, many of underlying features of industrial data are less adaptive and noise-full, giving rise to the need for their selection strategies. Support vector regression, partial least-squares, optimized sparse partial least squares and multi-output least-squares support vector regression are commonly used to mine the features to determine quality of product [9]. The authors in [10] have applied autonomous feature selection technique (EAFS) with an ensemble model to reveal intrusion within network flows. The authors in [11] identified a solution to perennial problem of cancer detection and treatment in which selection of Gene from micro-array data due to its high-dimensionality. Equally, the majority of cancer data bound

for practice and research are laced with noise and low instances. Therefore, individual and mix methods for feature selections that accounted for highly relevant and informative genes are adopted for the classification. To this end, noisy data and irrelevant genes elimination could improve cancer detection. The fundamentals of feature selection categories and their matching techniques include:

Filter feature selection uses scores to minimize the highly dimensional datasets, less-computationally intensive, speedy, and achieves higher generality. Most commonly filter methods include:

Information gain select features in datasets through entropy and amount of information in random variable. Mutual information is tailored towards non-linear associations between two random variables. Conditional mutual information maximization selects features using an approximation measure that reduce the correlation between genes and features.

Minimum redundancy maximum relevance entails used of mutual information algorithm by computing the maximum relevance between features and target and minimum redundancy between two random variables. Random forest ranking takes advantage of decision trees to combine forecasts across a set of random trees and accuracy-based ranking. Fast correlation-based filter is most applicable to multivariate criteria for minimizing noise and retaining highly relevant data through symmetric uncertainty method.

Fisher's scoring algorithm or F-score is the selection technique based on F-distribution to choose each unique feature matching the target feature, that is, the smallest distance in terms of minimum interclass distance and maximum intraclass distance. Relief algorithm adopts correlation score between features, and their weights to choose the features to prediction. Outcomes are most impacted by noise and errors in the features subsets. Pareto optimization is a multi-objective optimization to create and reveal the collection of best acceptable solution from the solutions space by the decision-maker.

Wrapper feature selection uses classifier and machine learning algorithms to seek out optimal features subsets from the raw search space of features. It is most desirable for ranking of features for moderate-dimensional datasets, though, computing resource-intensive. The categories are discussed as follows:

Evolutionary-based technique is propelled by nature's evolutionary process: Flower pollination algorithm, genetic algorithm. Swarm-based technique realized from the animals' social behaviours: Artificial bee colony algorithm, cuckoo search, dragonfly algorithm, moth flame algorithm, particle swarm optimization, firefly algorithm, bat algorithm, and ant colony optimization, Human-based technique taken after activities and behaviour of humans: Teaching-learning-based algorithm, and learning automata. Physics-based technique following from physical processes of the nature: Black hole algorithm, and gravitational search algorithm. Music-based technique motivated from music instrument: Harmony search. Others: Crossover, and stacked autoencoder.

Also, the principal component analysis and model-based feature selections had recently been applied in hyperspectral image prediction tasks [12]. Information retrieval is a developing aspect of computer science in discovering important information form piles of the same. Natural language processing offers numerous relationships between textual data [13]. Machine learning models are holding sway in personalized disease risk forecasts such as early screening and treatment using single nucleotide polymorphisms associations [14]. The key contributions of this paper include:

i. To extract feature importance using the feature selection methods on the textual dataset.
ii. To determine the effectiveness of the feature selection methods using XGBoost model as base estimator.
iii. To compare the outcomes of XGBoost model based on the selected features using standard metrics.

The organization of the paper includes: section two is the related studies. Section 3 is the methodology. Section 4 is the results and discussion. The last section is the conclusion.

2 Related studies

In Arabic text classification problem, an improved Chi Square feature selection was applied on common classifiers for Arabic texts classification by [13]. Considering measuring metrics of precision, F-score, runtime, and recall, the performances of the classifiers were better after feature selection process. With pre-processing and improved CHI feature extraction, the RF model achieved precision of 0.958. NB obtained recall of 0.966. F1-score of 0.960 was attained by NBM model. Though, more optimization algorithms could augment feature selection methods performances. Other forms of dataset can be investigated.

The authors in [15] experimented the cause-effect of wrapper, filter, and embedded feature selection techniques on Random Forest, Artificial Neural Network and Gradient Boosting Machine. The outcomes revealed that, mutual information generated features, and the Gradient Boost Machine predictions for the wastewater components were the most superior using R square and mean square error. The GBM attained the biggest performance by R^2 of 0.58, RMSE of 0.092, and MAE of 0.017. However, the GBM is less sensitive to changes in features set when compared to RF and ANN models.

In [3], the authors highlighted in a survey about swarm intelligent feature selection technique in which Ant algorithm speedups execution, lower costs, and generates better results. But it lags behind in terms of accuracy against Cuckoo search, genetic algorithm, and Bees colony. The feature selection increased the detection rate of healthy and cancer patients through expression of gene profile by accuracy of 99.13% using enhanced Cuckoo search in 29 features against accuracy of 98.75% for original 34 features when applied on digital image of breast cancer dataset.

The author in [12] adopted an enhanced binary equilibrium optimizer algorithm to raise the feature

selection tasks and minimize the dimension of the datasets. The KNN and SVM classifiers, wrapper methods, were used to counter the over-fitting challenge. The outcomes after statistical analysis were better with the introduced method. Using the vegetation indices, and principal component analysis feature selection approaches to improve the prediction of plant water content using partial least square regression, random forest, and backpropagation neural network. Consequently, the hybrid feature selection of model-based features, and principal component analysis raised the performance of the enhanced backpropagation neural network model (BPNN-PCA-MF) by R2 of 0.998, and RMSE of 25.20% respectively. Future works could extend the model to others areas of applications.

In [16], they experimented mutual information, univariate ROC-AUC, and fisher score filter feature selection methods with thirty-two machine learning approaches. Authors leveraged on multiple correlation approach to disband analogous features, reduce dimensionality, and increase the performance significantly in terms of accuracy of 26.50%, F1-score of 70.90%, and ROC of 26.74%. The mutual information technique with random forest correlation having the strongest influence on the outcomes.

In [17], they developed a general learning equilibrium optimizer (GLEO) founded on the wrapper feature selection strategy for diseases diagnosis. Authors utilised the new strategy to uncover a collection of 9 important biological features hidden in the volume of 16 attributes. The outcomes were outstanding based on the new feature selection procedure with regards to accuracy, fitness value, and size of feature. The informative features selection increases accuracy of biological classification jobs in which the GLEO applied on Leukemia dataset achieved accuracy of 0.9986. Future studies could investigate other metaheuristic algorithms and their hybridizations schemes.

The authors in [4] conducted feature selection on large-size DDoS attacks on 5G core network infrastructure using machine learning method to decimate time, noise and latency. Authors leveraged on the feature selection to ride-off noisy features from the data which increase the accuracy of the detection. The Stacking based feature selection produced accuracy of 97.183%, precision of 97.442%, F1-score of 97.065%, recall of 96.724%, and AUC of 98.075% for reduced features of 10 of DDoS intrusion dataset. Whereas, only the filter, wrapper, and ML model were investigated.

The authors in [18] noted that, radiomics technique is a common quantitative features extraction for medical imagery. These distribution of features raises the accuracy of the prognostics and classification of diseases at the early stages like lung cancer. Though, further refinement of radiomics including consistency, cohort power, harmonization of feature value, etc., to increase stratification of risk, and management of patients. But the study did not provide empirical explanations about the optimal technique of the multiple features selection approaches.

In [10], the authors designed an ensemble autonomous feature selection known as EAFS for intrusion detection tasks. The EAFS computed the feature importance, which were used up by normalized score of mixed (NSOM) to create the subset from the volume on basis of high score. Upon validation, the classification accuracy of the ensemble model at 99.34% was better with the UNSW-NV15 datasets. Similarly, false alarm rate of 1.69% was observed.

The authors in [14] discussed the advantage of undertaking feature selection that extract most informative feature within the noisy non-informative, redundant, and irrelevant features. The concept targets genetic sequencing for improving precision medicine and patients' healthcare through distinct genetic makeups. The multivariate filter methods (chi square test) were used to explain the relationship between single nucleotide polymorphisms known as epistasis effects. In this way, complex diseases were detected by eliminating genetic features associated with insignificant heritability of individuals. However, there is no empirical results from study.

In [7], the authors utilised four feature selection techniques for curating optimum parameter subsets within initial data for mapping soil organic matter in multifaced forest landscape of Southern China. Authors leverage on wrapper, embedded, filter and ensemble models to create optimum variable of the datasets. The outcomes indicated that, XGBoost was superlative ensemble performer than wrapper and filter models using root mean square error (RMSE) and correlation coefficient (R2). The RRA-based ensemble model performance improved by R2 of 5.77% and RMSE of 9.16%. The reason for this trend can be explain by the use of multiple selectors and robust rank aggregation procedure.

The authors in [8] attempted to apply machine learning in finding evidence of COVID-19 from sets of indicators within blood tests of suffers. The idea was to reduce poor morbidity and prognosis of COVID-19, which achieved through the use of matrix factorization and Random Forest algorithm. The former reduced the high-dimensionality in blood biomarker space of COVID-19 patients in which Arterial Blood Gas O2 saturation and C-reactive Protein are held high as important features to prognosis. There are prospects of using ML for clinical and pathogenesis of COVID-19 but, there could be loss of features.

In [9], they developed a causal based on autonomous selection of features of the soft sensing in key performance indicators. This technique is motivated by non-linear causal model and information theory to address the interleave between original industrial dataset's key performance indicators and every feature importance. The AdaBoost ensemble method constructed the soft sensors' key performance indicators after a non-zero causal-and-effect autonomous features selection, which realized RMSE of 2.215, and R2 of 61.1. Though, it was discovered that the ML models are weak for dataset containing causal features.

The authors in [19] utilised ranker-based feature selection method with 9 machine learning to correctly classify eye diseases. The outcomes revealed that, the SVM outpaced with accuracy of 99.11%, followed by LR at 98.58%. A

nested ensemble selection method comprising wrapper and filter feature selection techniques for precision, efficiency and simplicity by [20]. This approach was capable of extracting relevant variables from irrelevant, and correlated from redundant features. Using the real-life and synthetic datasets, the nested ensemble selection attained relatively higher accuracy even for multiclass cases.

In [21], they developed an ensemble feature selection method for combining the effectiveness of single feature selection algorithms such as correlations between forest aboveground biomass and features, and the multicollinearity of selected features. The stability heterogeneity correlation-based ensemble method ranked and generated subsets of important features for high accuracy of prediction. The XGBoost model generated better R2 of 0.61 and RMSE of 15.32 against RF model at R2 of 0.57 and RMSE of 16.16 for 26 of 46 features extracted.

In [22], the authors leveraged on correlated and frequent items, a text feature selection for subsequent classification tasks. The proposed strategy account for importance and feature interactions with association to determine the interrelationship between features and target class. The validation process adopted SMS spam dataset that produced 95.12% accuracy of text classification after optimal features of 6%.

Summary of the key related studies including author(s), objective(s), techniques, results, and weaknesses are presented in Table 1.

Table 1: Summary of the key related studies

Author(s)	Objective(s)	Techniques	Results	Weaknesses
Alshaer et al. (2021)	Arabic text classification.	Feature extractor: Chi-Square. Estimator: NBM.	Accuracy: 0.966. F1-score: 0.960.	More datasets and optimizers required.
Bagherzadeh et al. (2021)	Cause-effect wastewater.	Wrapper, filter and ML.	GBM best features selector by R^2: 0.58. RMSE: 0.0092, MAE: 0.017.	GBM is less adaptive to changes in data features.
Jameel & Abdullah (2021)	Cancer gene profiling and diagnosis.	ML: Cuckoo Search, Genetic algorithm and Bees Colony.	Accuracy: 99.13% for 29 features against 98.75% for original 34 features.	Digital dataset only.
Too & Mirjalili (2021)	Diseases diagnosis.	Wrapper: GLEO applied on Leukemia dataset.	Accuracy: 99.86%.	More metaheuristic algorithms can be investigated.
Jain & Saha (2022)	Data attributes extraction.	Mutual information, ML, and filter.	Accuracy: 26.50%. F1-score: 70.90%. ROC: 26.74% for mutual information technique.	Low performance of mutual information technique.
Zhang et al. (2022)	Intrusion detection tasks.	ML: Ensemble autonomous features selection.	Accuracy: 99.34%, false positive: 1.69% with UNSW-NV15 dataset.	Only ML feature selected was investigated.
Saberi-Movahed et al. (2022)	COVID-19 evidence in blood sample.	ML: matrix factorization. and random forest.	Dimensionality reduction. Importance feature scores	No-textual dataset.
Sun et al. (2023)	Causal based autonomous features selection in soft sensing.	Information theory model.	No empirical results.	More data complexity.
Kamalov et al. (2023)	Relevant features selection.	Nested ensemble approach with filter and wrapper.	No empirical results.	More complexity in dataset.
Marouf et al. (2023)	Eye disease features classification.	ML approaches such as SVM and LR.	SVM: Accuracy of 99.11%. LR: 98.58%.	Technique limited to ML.
Zhang et al. (2023)	Text feature selection.	Forest aboveground biomass and multicollinearity of selected features.	No empirical outcomes.	Domain non-specific.
Mamdouh & El-Hafeez (2023)	Text features selection in SMS spam dataset.	Importance and interaction between features and target class.	Accuracy: 95.12%.	Unknown technique.

From Table 1, the majority of the studies had identified the propensity of feature selection based on various methods classification tasks with the sole goal of denoising and redundancy removal using information theory, and machine learning approaches. Most prominent among the techniques are information theory, ML, wrapper, filters, and ensemble methods.

3 Research methodology

This paper investigates the performances of features selection techniques based on machine learning and information theory. On the part of the machine learning, the XGBoost was selected due to its decision-making capability in which weak classifiers are augmented. While, Maximal Information Coefficient was elected from information theory for choosing features within datasets based on non-functional and functional associations among variables. The proposed features selection approach is shown in Figure 1.

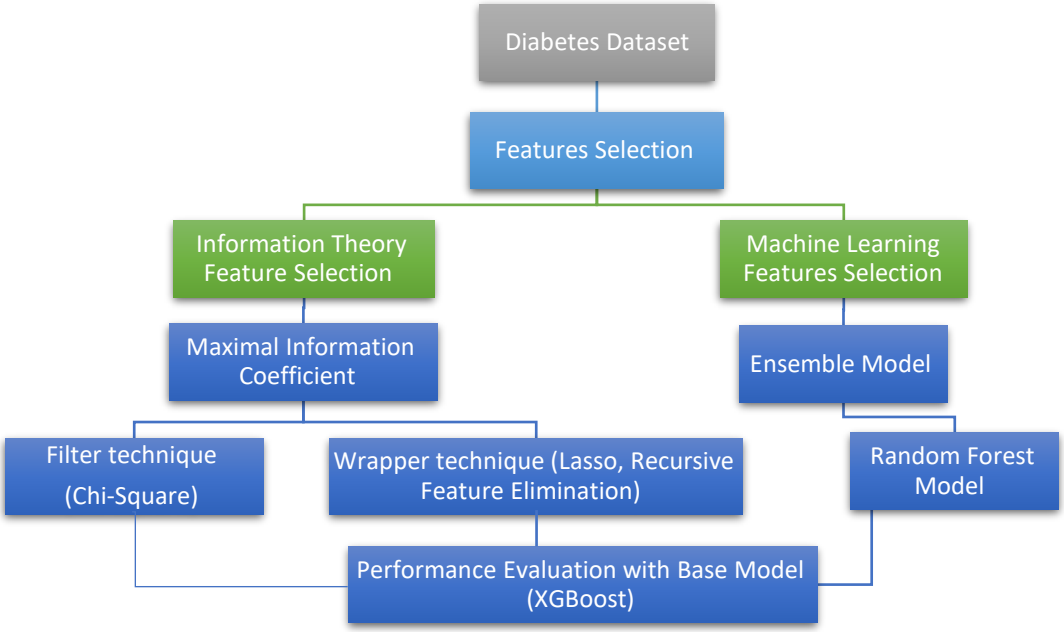


Figure 1: The layout of the features selection strategies performance comparisons study

From Figure 1, the main approaches considered for the Diabetes features selection include: wrapper (Lasso, Recursive feature elimination), filter (Chi-Square), and machine learning algorithm (XGBoost). The performance evaluation parameters related to the features selection technique are utilised for study. The key features selection approaches are discussed as follows:

Maximal information coefficient is an information theory-founded measure of association by accounting several non-functional and functional relationships existing within variables. Generally, it is equal to the coefficient of determination (R^2) as given by Equation 1.

$$R^2 = \frac{\sum(a-\bar{a})*(b-\bar{b})}{\sqrt{\sum(a-\bar{a})^2 * \sum(b-\bar{b})^2}} \quad (1)$$

Again, it uses values between 0 and 1, which explains 0 as statistical independence, and 1 as an entirely noiseless relationship. It can be represented by Equation 2.

$$m(a,b) = \frac{\max\{I(a,b)\}}{\log_2 \min\{n_x, n_y\}} \quad (2)$$

Where:
m = Maximal Information Coefficient,
a,b = random factors with minimum joint entropy normalization,
$\bar{a}$, $\bar{b}$ = mean score of the random factors
n = mutual information.
m explains the mutual information between variables a and b including generalization over a range and the equitability property.
m allocates similar score to evenly noisy associations without consideration of the form of association.
The Maximal Information Coefficient is capable of capturing different types of relationships, linear and non-linear (such as cubic, exponential, sinusoidal, superposition of functions). It leverages on mutual information, that is, symmetric nature. It removes any

assumptions concerning the variables distribution. The mutual information nature offers robustness to outliers. The coefficient computation is between 0 and 1 for better comparisons and understanding of outcomes. Though, it is incapable of depicting the direction or nature of the relationships. It is extensive in terms of computation. It is statistical less-effective against the Pearson's and distance correlation for independent data.

XGBoost is a renown gradient boosting framework which combines several decision trees. It is in the class of the ensemble models used for feature selection, compute the function with most influencing factors from the datasets, generate regression tree for understanding the importance of factors, higher Z-score values within the random probes are considered as importance factors, and produce the weights of regression trees for prediction accuracy. The ensemble model is a form regression model with larger learning abilities, and relatively smaller sensitivity for noisy datasets.

The process of ranking of factor selection by importance is conducted by mixing the Robust Aggregate rule. Given that the normalized rank factors of $y_{(1)} \leq \cdots \leq y_{(z)}, z = (1 \ldots i, \ldots z)$ signifies the ranking techniques. The rank vector $y_{(\iota)\leq}\, y_{(z)}$ as given by Equation 3.

$$R_{i,z}(Xt) = \sum_{f=j}^{z} \binom{z}{f} . Xt^{f} (-(Xt+1))^{z-f} \qquad (3)$$

$y_{(\iota)}^{`}$ is the defined as the ordering statistic for z unique factors shared over evenly between 0 and 1 for sorted sets, $R_{i,z}(Xt)$. The eventual score computed for the factors' ranks, y, whose p-value is given by Equation 4.

$$\partial(y) = \min_{i=1,\ldots,z} R_{i,z}(Xt) \qquad (4)$$

where, ∂(y) is p-value of computed factors' ranks. The value of $\partial \rightarrow 0$ explains a relative higher importance of the factors ranked distributed on the scale of 0 – 1 for the data distribution ranking importance for low variance of interclass and highest variance of the intraclass.

Data collection: The Pima Indians Diabetes Database is a collection originally maintained by the National Institute of Diabetes and Digestive and Kidney Diseases. It is composed of data of 768 women in Phoenix, Arizona, USA. The test's results showed that 258 tested positive and 500 tested negative. Again, there is one target (dependent) variable and the eight other factors including: Pregnancies, Oral Glucose Tolerance Test (OGTT), BloodPressure, SkinThickness, Insulin, Body Mass Index (BMI), Age, and PedigreeDiabetesFunction. The Pima population has been under study by the National Institute of Diabetes and Digestive and Kidney Diseases at intervals of 2 years since 1965. The Dataframe Class:0 = 6373, Class:1 = 1627.

Experimental parameters: The optimal requirements for the personal computer used for experimentation include:

Hardware: AMD E1-1200 APU Processor with RadeomTM Graphics 1.40 GHz, 64-bit Operating System, 4.00 GB RAM, x64-based processor.

Software: 3.5 Windows Experience Index, Windows 10 Single Language 2012.

Hyperparameters settings: The base model, XGBoost, hyperparameters values include: learning_rate (0.5), max_depth (3, 12), min_child_weight (1, 10), subsample (0.5, 1.0), colsample_bytree (0.5, 1.0), gamma (0, 5), and n_estimators (50, 500). Classifier = XGBoost, n_estimators = 100, max_depth = 3, n_jobs =- 1, random_state = 42, Metric: accuracy_score, f1_score, recall_score, precision_score, confusion_matrix, Classification function = Fitting, Data preprocessing function = Standard Scalar. The default values of the hyperparameters of Random Forest, Lasso model, and RFE were utilised.

4 Results and discussion

Lasso Model is a Mutual Information Theory technique for selecting and ranking the best features with dataset. Table 2 presents the outcomes of the applying Lasso model in selecting the best features within diabetes data which provides feature importance scores accordingly.

Table 2: Lasso model selected features with XGBoost model classification outcomes.

Data Feature	Importance Score	Rank
Glucose	0.346466	1
BMI	0.183006	2
Age	0.180830	3
DiabetePedigreeFunction	0.147554	4
Preganacies	0.142144	5

From Table 2, the index of the selected features are 'Pregnancies', 'Glucose', 'BMI', 'DiabetesPedigreeFunction', and 'Age'. Table 2. Lasso Model Feature Importance and ranking outcomes.

The graphical illustration of the Lasso model's feature importance of the sampled data is shown in Figure 2.

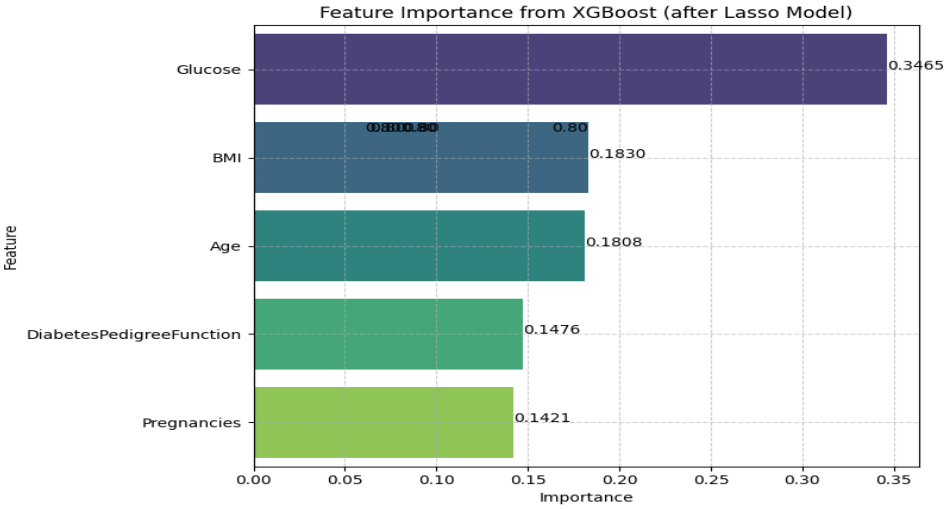


Figure 2: Lasso model feature importance and ranks representation.

In Figure 2, the feature importance as computed by the Lasso model utilised for XGBoost model-based decisions about risk of diabetes in women in Prima community of Arizona, USA indicated that, the first place is Glucose (34.65%), the 2nd placed important feature is BMI (18.30%), the 3rd placed Age (18.08%), the 4th placed feature is DiabetesPedigreeFunction (14.76%), and last placed important feature is Pregnancies (14.21%) respectively.

The XGBoost model with the selected features from Lasso model attained accuracy of 71.86%, recall of 71.25%, precision of 57.58%, and F1-score of 63.69%. Figure 3 illustrates the confusion matrix of the XGBoost model's classification outcomes with selected features through Lasso model. [109 42 23 57]

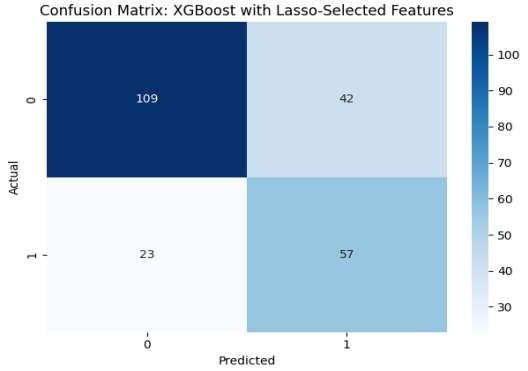


Figure 3: Confusion matrix of XGBoost model with Lasso model selected features.

From Figure 3, the XGBoost model correctly identified 57 breast cancer class of 109 original cases. Again, the XGBoost model identified wrongly 23 breast cancer class of the 42 unidentified cases.

The Wrapper Technique based on Recursive Feature Elimination calculates the best five features from the 8 original features are given by the index: ['Pregnancies', 'Glucose', 'Insulin', 'BMI', 'Age] in descending order of magnitude.

Performance of the RFE technique in selecting the best features achieved accuracy of 70.56%, recall of 61.25%, precision of 56.98%, and F1-score of 59.04%. The feature selection scores with RFE technique are presented in Table 3.

Table 3: The RFE feature selection importance scores and ranks.

Feature	**Importance Score**	**Rank**
Glucose	0.332591	1
BMI	0.193819	2
Age	0.191871	3
Insulin	0.151204	4
Pregnancies	0.130515	5

The graphical illustration of the feature importance from XGBoost model is shown in Figure 4. As shown, the Glucose, BMI and Age are the top three features from the dataset. While, the Pregancies and Insulin fall at the top 5.

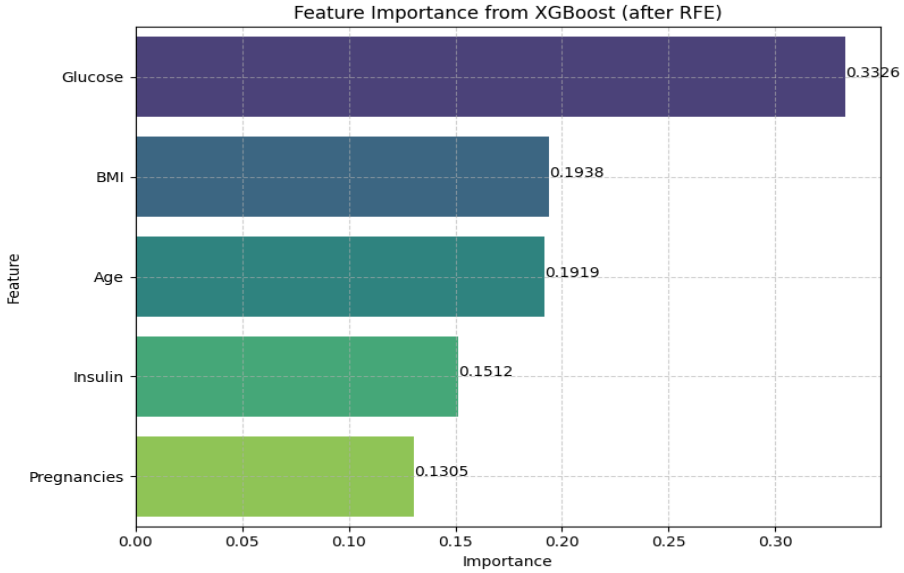


Figure 4: The Top5 important features after RFE with XGBoost model.

The confusion matrix of the reduced features from the diabetes as classified with XGBoost is given by Figure 5.

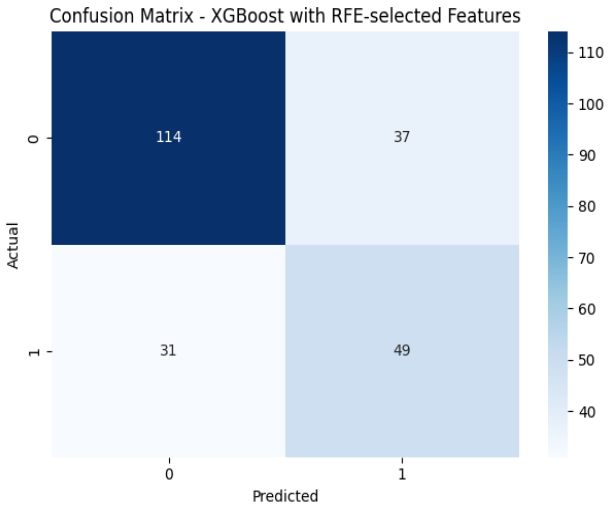


Figure 5: Confusion matrix of RFE and XGBoost model.

From Figure 5, the XGBoost model predicted correctly 49 breast cancer class of 114 original cases. Also, the XGBoost model predicted incorrectly 31 breast cancer class of the 37 unidentified cases.

The Filer Technique-based feature selection and ranking with Chi-Square scores computed for the features as shown in Table 4.

Table 4: The Chi-Square data feature importance score and rank computed.

Data Feature	Chi-Square Score	Rank
Age	8.205691	1
Glucose	7.094910	2
Pregnancies	6.559982	3
DiabetesPedigreeFunction	2.758584	4
Insulin	2.571590	5

From Table 4, the rankings of the data features with the Chi-Square scores are given as follows: Age (8.21), Glucose (7.09), Pregnancies (6.56), DiabetesPedigreeFunction (2.76), and Insulin (2.57). This shows that, Glucose is the topmost rank features when diagnosing diabetic patient. It implies that, Insulin and Age are the lowest and highest of the top-five features for diagnosing diabetic patients as shown in Figure 6.

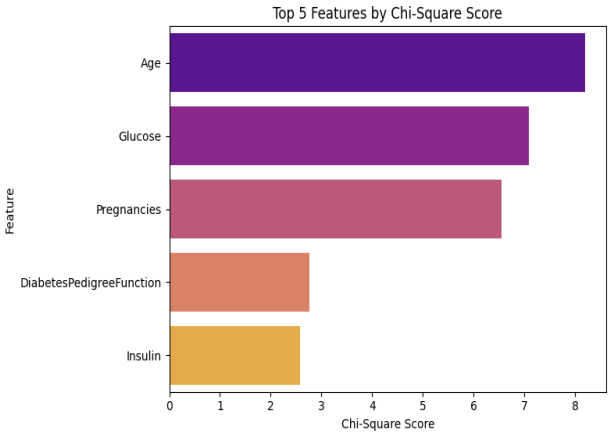


Figure 6: The 5 topmost features selected with Chi-Square Score.

The performance of the XGBoost model using the selected features from Chi-Square scores achieved accuracy of 75.76%, recall of 66.25%, precision of 64.63%, and F1-score of 65.43% respectively. The index of the selective features include: 'Pregnancies', 'Glucose', 'Insulin', 'DiabetesPedigreeFunction', and 'Age'. The confusion matrix of the Chi-Square Score with XGBoost model classification outcomes is shown in Figure 7.

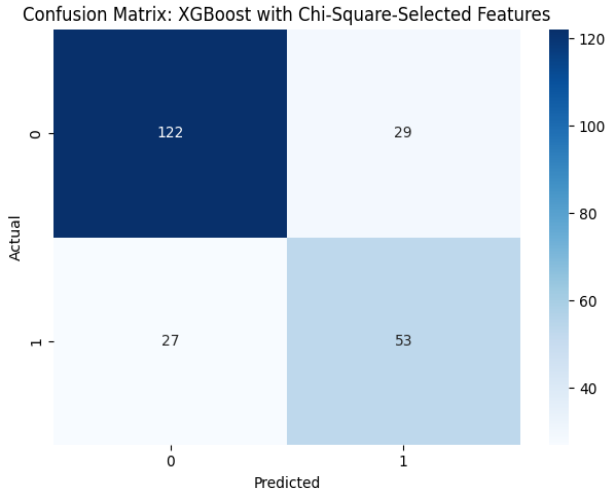


Figure 7: The confusion matrix of the Chi-Square scores with XGBoost model.

From Figure 7, the XGBoost model correctly predicted 53 breast cancer class of 122 original cases. Again, XGBoost model incorrectly predicted 27 breast cancer class of the 29 unidentified cases.

The feature importance scores computed by Random Forest algorithm and the corresponding rankings are provided in Table 5.

Table 5: The Random Forest algorithm's data features scores and ranks.

Data Feature	**Importance Score**	**Rank**
Glucose	0.282089	1
BMI	0.158120	2
Age	0.142116	3
DiabetesPedigreeFunction	0.113127	4
BloodPressure	0.084052	5

From Table 5, the Random Forest algorithm-based feature importance scores showed that, Glucose attribute was most important (28.21%), the 2nd placed important feature is BMI (15.81%), the 3rd placed important feature is Age (14.21%), the 4th placed important feature is DiabetesPedigreeFunction (11.31%), the least placed important feature is BloodPressure (8.41%) as depicted by Figure 8. The selected feature index includes: 'Glucose' 'BMI' 'Age' 'DiabetesPedigreeFunction' 'BloodPressure'

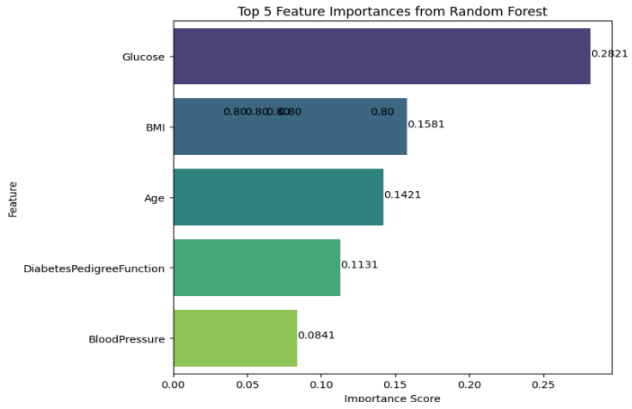


Figure 8: The Random Forest algorithm feature importance scores and ranks.

The outcomes of the classification tasks of the XGBoost model using importance features generated by Random Forest algorithm offered accuracy of 70.23%, F1-Score of 59.17%, recall of 62.50%, and precision of 56.18%. The confusion matrix of XGBoost mode after features selection with Random Forest algorithm is shown in Figure 9.

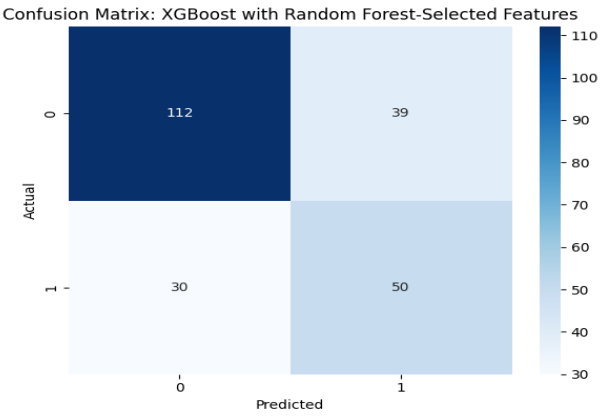


Figure 9. The confusion matrix of XGBoost model with Random Forest algorithm features.

From Figure 9, the XGBoost model rightly predicted 50 breast cancer class of 112 original cases. Again, XGBoost model incorrectly predicted 30 breast cancer class of the 39 unidentified cases.

4.1 Models comparisons

The comparisons of both approaches of features selection using the feature importance weights computed and classification metrics are presented in Table 6.

Table 6. Comparisons of feature selection models rankings.

Performance measure	Mutual Information Theory or Embedded Technique (Lasso model)	Wrapper (Recursive Feature Elimination)	Filter (Chi-Square)	Machine Learning (Random Forest)
Accuracy	71.86%	70.56%	75.76%	70.23%
Recall	71.25%	61.25%	66.25%	62.50%
Precision	57.58%	56.98%	64.63%	56.18%
F1-score	63.69%	59.04%	65.43%	59.17%
Confusion matrix	[109 42 23 57]	[114 37 31 49]	[122 29 27 53]	[112 39 30 50]

From Table 6, the best feature selection technique is the filter technique the XGBoost model at accuracy of 75.76%, precision of 64.63%, and F1-score of 65.43%. The mutual information theory or embedded technique offers the best recall score of 71.25% trailed by the filter technique. The mutual information provided the least false-positive of 23 followed by the filter technique at 27. The capacity of the filter and mutual information techniques in raising the classification outcomes of the XGBoost model when applied for computing the most importance factors for reaching the diagnosis diabetes among women in Pima India community in Phoenix, Arizona, USA. The reasons being that, the feature selection techniques reduced the dimensionality, redundancy in the variables, and retain the data variance.

In this way, overfitting can be minimized while increasing the degree of freedom of the base model (XGBoost) during classification tasks. In particular, the process of determining the best feature importance is repeated until expected features are identified and eventually chosen.

The significance (2-tailed) test conducted on the outcomes obtained for the Filter technique with test value of 0.7576 as the accuracy score which is $p<0.05$ (that, $p = 0.059$). Therefore, the filter technique outcomes are statistically significant at confidence value of 0.95.

5 Conclusion

This paper compared the feature selection performances of the different approaches including wrapper (Lasso, Recursive feature elimination), filter (Chi-Square), and machine learning algorithm (Random Forest). The base model was XGBoost because of its capacity to aggregate weak classifiers' outcomes to reach an improved decision. The different feature selection techniques were applied on the 8 features in the diabetes dataset. Majority of the feature selection techniques identified the following attributes has best for diagnosis heart diseases including: Glucose, BMI, Age, Pregnancies, BloodPressure, Insulim, and DiabetesPedigreeFunction.

The filter technique improves the accuracy of the XGBoost model by 75.76%; precision of 64.63%, and F1-score of 65.43%. Similarly, the mutual information theory or embedded technique gives the best recall score of 71.25% followed by the filter technique at 66.25%. the same trend was observed in that the mutual information provided the least false-positive of 23 superiors to the filter technique at 27. The filter technique outcomes two-tailed significance test score of $p = 0.059$), that is, $p<0.05$. It implies outcomes obtained are statistically significant at confidence value of 95%.

These feature selection techniques reduced the dimensionality, redundancy in the variables, and retain the data variance. Also, base model's overfitting is minimized while increasing its degree of freedom during classification tasks. Both feature selection methods undertake extended processes of determining the best feature importance before eventually choosing the best attributes within the data to explain the variance and patterns. Future work can consider use of mixture of the feature selection techniques, base models, and more complex datasets to ascertain the effectiveness of the approaches.

References

[1] D. A. Elmanakhly, M. M. Saleh, and E. A. Rashed, "An improved equilibrium optimizer algorithm for features selection: methods and analysis," *IEEE Access*, vol. 9, pp. 120309–120327, 2021, doi: 10.1109/access.2021.3108097.

[2] J. Zacharias, M. von Zahn, J. Chen, and O. Hinz, "Designing a feature selection method based on explainable artificial intelligence," *Electronic Markets*, vol. 32, no. 4, pp. 2159–2184, 2022, doi: 10.1007/s12525-022-00608-1.

[3] N. Jameel and H. S Abdullah, "Intelligent feature selection methods: A survey," *Engineering and Technology Journal*, vol. 39, no. 1B, pp. 175–183, 2021, doi: 10.30684/etj.v39i1b.1623.

[4] Y.-E. Kim, Y.-S. Kim, and H. Kim, "Effective feature selection methods to detect IoT DDoS attack in 5G core network," *Sensors*, vol. 22, no. 10, p. 3819, 2022, doi: 10.3390/s22103819.

[5] A. K. Hamoud *et al.*, "A comparative study of supervised/unsupervised machine learning algorithms with feature selection approaches to predict student performance," *International Journal of Data Mining, Modelling and Management*, vol. 15, no. 4, pp. 393–409, 2023, doi: 10.1504/ijdmmm.2023.134590.

[6] M. Rostami, K. Berahmand, E. Nasiri, and S. Forouzandeh, "Review of swarm intelligence-based feature selection methods," *Eng Appl Artif Intell*, vol. 100, p. 104210, 2021, doi: 10.1016/j.engappai.2021.104210.

[7] Y. Chen *et al.*, "Comparison of feature selection methods for mapping soil organic matter in subtropical restored forests," *Ecol Indic*, vol. 135, p. 108545, 2022, doi: 10.1016/j.ecolind.2022.108545.

[8] F. Saberi-Movahed *et al.*, "Decoding clinical biomarker space of COVID-19: Exploring matrix factorization-based feature selection methods," *Comput Biol Med*, vol. 146, p. 105426, 2022, doi: 10.1016/j.compbiomed.2022.105426.

[9] Y.-N. Sun, W. Qin, J.-H. Hu, H.-W. Xu, and P. Z. H. Sun, "A Causal Model-Inspired Automatic Feature-Selection Method for Developing Data-Driven Soft Sensors in Complex Industrial Processes," *Engineering*, vol. 22, no. 3, pp. 82–93, 2023, doi: 10.1016/j.eng.2022.06.019.

[10] Y. Zhang, H. Zhang, and B. Zhang, "An effective ensemble automatic feature selection method for network intrusion detection," *Information*, vol. 13, no. 7, p. 314, 2022, doi: 10.3390/info13070314.

[11] H. Almazrua and H. Alshamlan, "A comprehensive survey of recent hybrid feature selection methods in cancer microarray gene expression data," *IEEE Access*, vol. 10, pp. 71427–71449, 2022, doi: 10.1109/access.2022.3185226.

[12] O. Elsherbiny, Y. Fan, L. Zhou, and Z. Qiu, "Fusion of feature selection methods and regression algorithms for predicting the canopy water content of rice based on hyperspectral data," *Agriculture*, vol. 11, no. 1, p. 51, 2021, doi: 10.3390/agriculture11010051.

[13] H. N. Alshaer, M. A. Otair, L. Abualigah, M. Alshinwan, and A. M. Khasawneh, "Feature selection method using improved CHI Square on Arabic text classifiers: analysis and application," *Multimed Tools Appl*, vol. 80, pp. 10373–10390, 2021, doi: 10.1007/s11042-020-10074-6.

[14] N. Pudjihartono, T. Fadason, A. W. Kempa-Liehr, and J. M. O'Sullivan, "A review of feature selection methods for machine learning-based disease risk prediction," *Frontiers in Bioinformatics*, vol. 2, p. 927312, 2022, doi: 10.3389/fbinf.2022.927312.

[15] F. Bagherzadeh, M.-J. Mehrani, M. Basirifard, and J. Roostaei, "Comparative study on total nitrogen prediction in wastewater treatment plant and effect of various feature selection methods on machine learning algorithms performance," *Journal of Water Process Engineering*, vol. 41, p. 102033, 2021, doi: 10.1016/j.jwpe.2021.102033.

[16] S. Jain and A. Saha, "Rank-based univariate feature selection methods on machine learning classifiers for code smell detection," *Evol Intell*, vol. 15, no. 1, pp. 609–638, 2022, doi: 10.1007/s12065-020-00536-z.

[17] J. Too and S. Mirjalili, "General learning equilibrium optimizer: a new feature selection method for biological data classification," *Applied Artificial Intelligence*, vol. 35, no. 3, pp. 247–263, 2021, doi: 10.1080/08839514.2020.1861407.

[18] G. Ge and J. Zhang, "Feature selection methods and predictive models in CT lung cancer radiomics," *J Appl Clin Med Phys*, vol. 24, no. 1, p. e13869, 2023, doi: 10.1002/acm2.13869.

[19] A. Al Marouf, M. M. Mottalib, R. Alhajj, J. Rokne, and O. Jafarullah, "An efficient approach to predict eye diseases from symptoms using machine learning and ranker-based feature selection methods," *Bioengineering*, vol. 10, no. 1, p. 25, 2022, doi: 10.3390/bioengineering10010025.

[20] F. Kamalov, H. Sulieman, S. Moussa, J. A. Reyes, and M. Safaraliev, "Nested ensemble selection: An effective hybrid feature selection method," *Heliyon*, vol. 9, no. 9, 2023, doi: 10.1016/j.heliyon. 2023.e19686.

[21] Y. Zhang, J. Liu, W. Li, and S. Liang, "A proposed ensemble feature selection method for estimating forest aboveground biomass from multiple satellite data," *Remote Sens (Basel)*, vol. 15, no. 4, p. 1096, 2023, doi: 10.3390/rs15041096.

[22] H. Mamdouh Farghaly and T. Abd El-Hafeez, "A high-quality feature selection method based on frequent and correlated items for text classification," *Soft comput*, vol. 27, no. 16, pp. 11259–11274, 2023, doi: 10.1007/s00500-023-08587-x.

Path Planning Optimization for Industrial Robots using an Adaptive Improved Differential Evolution Algorithm

Yang Zhang, Gaofeng Xu*
School of Artificial Intelligence Application, Shanghai Urban Construction Vocational College, Shanghai 201415, China
E-mail: hd_2323@126.com
*Corresponding author

Keywords: path planning, robotics, DE, IASCV, velocity steps

Received: March 6, 2025

Industrial robot path planning faces mechanical stress accumulation and motion instability caused by velocity step in complex scenarios. In this study, an adaptive optimization model is proposed, integrating the improved differential evolutionary algorithm and asymmetric S-type velocity planning. This model achieves global-local cooperative optimization through dynamic parameter coupling and velocity field feedback mechanism. Moreover, this study constructed a 10m ×6m dynamic simulation environment (including 12 static/dynamic obstacles) based on the Gazebo platform, and set parameters such as the maximum linear velocity of the robot as 2m/s and the safety distance as 0.3m. The performance of the model was also compared with traditional benchmark methods (ant colony optimization algorithm, etc.) and most advanced methods (improved dynamic window algorithm, etc.). Simulation experiments showed that the average path error of the model was reduced by 76.3%, the number of convergence iterations was reduced by 33.3%, and the AUC value was improved by 9.6%. The model's dynamic obstacle scenario planning efficiency in real-world environments was improved by 23.5%, the motion stability index was improved by 7.2%, and the trajectory tracking energy consumption was reduced by 34.7%. In summary, the adaptive population initialization strategy of IDE and the segmented plus acceleration constraints of IASCV achieve a smooth transition of the velocity profile through the dynamic replanning mechanism in the overspeed region, breaking the limitation of the decoupling of the existing trajectory optimization and dynamics constraints. In conclusion, the research model reaches the advanced level in terms of convergence speed, environmental adaptability and industrial energy efficiency, providing a highly robust solution for complex industrial scenarios.

Povzetek: Razvit je adaptivni algoritem A-I-DE za načrtovanje poti industrijskih robotov, ki z integracijo IDE in IASCV doseže večjo natančnost, točnost, stabilnost in energetsko učinkovitost v kompleksnih okoljih.

1 Introduction

As the market demand for industrial products is increasing day by day. Mobile industrial robots instead of manual labor to participate in the inspection, sorting, handling and other production processes and gradually become the main development trend of intelligent industry [1]. Optimal path planning (OPP) is an important guarantee for industrial robots to complete their work successfully, and it is also the focus of current industrial robot research. Through the optimization of path planning (PP), the robot's work efficiency and work accuracy can be significantly improved, reducing energy consumption and ensuring the efficient completion of the task [2]. The two primary categories of robot trajectory planning algorithms now in use are joint space trajectory planning (STP) and Cartesian STP. Joint STP is concerned with the change in angle of each joint, whereas Cartesian STP is concerned with the end-effector's path in the workspace [3].

However, these traditional PP algorithms have many problems. For example, linear interpolation has limited planning ability for complex paths and cannot handle higher order curves or surfaces [4]. Inverse kinematics algorithms have no solution or multiple solutions in some cases, increasing the computational complexity [5]. Dynamic PP algorithms rely on accurate robot dynamics models and are computationally intensive with high real-time requirements [6]. Consequently, while extant methods optimize the global search path, they do not integrate velocity profile smoothing, resulting in mechanical stress accumulation. Moreover, they rely on data-driven approaches and lack explicit optimization of trajectory excitation parameters. Both of them do not synergistically solve the velocity step and excitation mechanism problems, leading to limited motion stability. The motion step problem refers to the accumulation of mechanical stresses caused by sudden velocity changes. It is a common challenge in dynamic PP (not inherent to DE algorithms), but directly affects the accuracy of robot motion. Therefore, this study focuses on the underlying logic and excellent performance of differential evolution (DE), proposes improved DE (IDE), and then establishes an OPP model for industrial robots based on adaptive IDE (A-I-DE). The objective of the research is to develop a

dynamic coupling framework of heterogeneous algorithms to address the timing synergy problem between global trajectory optimization and local mechanical constraints through velocity field-driven multimodal real-time feedback. The aim is to achieve the simultaneous enhancement of PP accuracy and motion stability in complex industrial scenarios. The innovation of the research lies in the closed-loop collaborative optimization mechanism of IDE and improved asymmetric s-curve velocity profile (IASCV): It dynamically identifies the trajectory risk region through velocity extremum search, combines adaptive parameter adjustment and iterative correction of multi-speed regions, and realizes the global search capability and local trajectory smoothing of DE in deep coupling. The deep coupling of DE global search capability and local trajectory smoothing breaks through the unidirectional serial connection mode between traditional DE variants and trajectory post-processing.

There are four parts to the research. The first part introduces the current worldwide research on the logic and related algorithms of PP for industrial robots. The second part establishes an accurate and efficient OPP model for industrial robots from DE algorithm and IASCV. In the third part, the proposed PP algorithm model is analyzed with examples to verify its reliability. The last part summarizes and analyzes the article comprehensively.

2 Related works

The use of robots in the production of many industrial domains has demonstrated a fast-expanding tendency as large-scale development in numerous industries has advanced [7]. OPP is one of the core performances of industrial robots, and it is also an application direction that needs to be continuously expanded and deepened in smart industry [8]. However, in complex real-world environments, the PP of robots is often accompanied by various uncertainties, so many researchers are improving this problem. Aiming at the traditional A* algorithm's slow planning speed and paths close to obstacles, Li Y et al. proposed an improved A* algorithm based on extended distance and bi-directional search, which improved the algorithm's PP efficiency [9]. For the three-dimensional environmental PP problem, Dong L proposed an improved A* algorithm with multi-sensor fusion. It significantly improved the path search efficiency and environmental adaptability by optimizing the heuristic function and the cost factor [10]. To address the problems of industrial robots in terms of coordination constraints as well as poor performance in task allocation, Hartmann VN et al. proposed a PP method based on a sampling-based bi-directional spatio-temporal path planner. The method was highly robust and scalable in the long-time dimension [11]. Sun D et al. developed a PP approach based on temporal signposts to enhance the performance of multi-robot collaborative task PP in a long-time domain, aiming to address the issue of standard algorithms' restricted scalability in terms of planning time domain [12]. A new hybrid method based on the firefly algorithm and genetic algorithm was presented by Zhang T W et al. with the goal of improving the robot's computational ability and responsiveness in PP. The firefly algorithm would easily fall into the local optimal solution (OS) and other difficulties [13]. Zan J. suggested a hybrid PP algorithm based on genetic and whale optimization algorithms to increase the effectiveness of robot PP for issues like PP in dynamic and complicated situations [14]. Liu Y et al. proposed an enhanced jumping frog swarm algorithm to address issues like traditional PP algorithms' poor search capabilities and propensity to fall into local optimization. This algorithm enhanced the quality of the OS and the stability of both low and high dimensional searches [15].

Furthermore, to address the problem of planning optimal maneuvering trajectories in uncertain environments, Chai R et al. proposed a prioritized empirical playback algorithm with noise to improve the performance of robot motion planning and collision avoidance [16]. Chen P et al. developed a PP approach based on the soft actor-critic algorithm to address issues like the difficulties of PP in dynamic environments. This method was able to successfully avoid shifting impediments in the environment and finish the planning assignment with a high success rate [17]. The time required for the robot to visit a series of objects under complex operating conditions was significantly decreased by Nawaz F et al.'s suboptimal method, which was polynomial at each time step [18]. Khlif N et al. proposed a Q-learning PP method based on an improved exploration strategy for the utilization and exploration balance in PP algorithms. The method addressed the shortcomings of the algorithm in terms of reduced cumulative rewards in exploration and falling into local optimality in utilization [19]. Aiming at the problem of ship path smoothness and disturbance adaptation, Li D proposed a PP method based on Fermat curves. This study could optimize the motion trajectories between waypoints and compensate for the side-slip angle, and improve the navigation accuracy and energy efficiency [20]. In summary, many researchers around the world have considered the problems of PP for robots in different environments and have conducted several research efforts to solve these problems, as shown in Table 1.

Table 1: Recent advances in PP for robots.

Method	Clusters	Advantages	Disadvantages	Key findings	References
Improved A algorithm with dynamic window approach (IA-DWA)	Modification of the classical algorithm	Combine global and local planning to improve obstacle avoidance efficiency; support dynamic environment adaptation.	High computational complexity and limited real-time performance.	Bidirectional search strategy reduces path redundancy and dynamic window improves obstacle avoidance flexibility.	[9]

Bidirectional spatiotemporal path planner (Bi-STP)	Sampling-based path planning	Long-time domain task robustness; multi-robot collaboration support.	High-dimensional space computational complexity increases significantly.	Spatio-temporal separation strategy reduces planning dimensions and improves scalability.	[11]
Hybrid genetic-firefly algorithm (HGFA)	Hybrid optimization algorithms	Balancing global search and local optimization to avoid local optima.	Parameter sensitivity is high and requires complex parameter tuning.	The cross mutation mechanism of genetic algorithm enhances the convergence of firefly algorithm.	[13]
Polynomial-time suboptimal multi-target algorithm (PTSMA)	Multi-intelligent body collaborative planning	Polynomial time solutions to multi-objective access problems are efficient.	Suboptimal solutions may deviate from the global optimum.	Hierarchical optimization strategy based on Markov decision process shortens the task time.	[18]
Deep reinforcement learning with prioritized experience replay (DRL-PER)	Reinforcement learning for dynamic environments	Strong real-time obstacle avoidance performance; no need for accurate environment modeling.	Depends on a large amount of training data, high hardware resource demand.	Prioritized experience playback mechanism accelerates strategy convergence, noise injection improves robustness.	[16]
Modified Q-Learning with exploration strategy (MQL-ES)	Heuristic search optimization	Balance exploration and exploitation to avoid local optima.	Convergence is slow and relies on initial parameter settings.	Improved exploration strategy reduces invalid path search and improves cumulative reward efficiency.	[19]

In addition, the accuracy and stability of PP are crucial for industrial robots to work efficiently and industrial production to be conducted in a safe and orderly manner, and its importance is self-evident. However, the majority of the aforementioned methods prioritize global path optimization or dynamic obstacle avoidance, yet they do not systematically address the mechanical stresses induced by the velocity step. Moreover, these methods lack adaptive optimization mechanisms for the trajectory excitation parameters, which hinders the integration of motion stability and path accuracy. Therefore, based on the DE algorithm, the study proposes an OPP model for industrial robots based on the A-I-DE algorithm, taking the rationality of the incentive mechanism and the stability of the motion as the optimization objectives. The study provides a comprehensive and innovative solution to solve the uncertainty and efficiency problems of industrial robot PP in real complex environments.

3 Methods and materials

This section describes in detail the A-I-DE algorithm, which consists of IDE and IASCV. Among them, IDE is responsible for the implementation of the orbit excitation mechanism, and the performance of the model in solving the motion steps is further improved when combined with IASCV.

3.1 IDE algorithm based on track excitation design for industrial robots

In the industrial robot motion model, the track excitation mechanism can guide the robot to reach the destination accurately, optimize the path selection, and realize the industrial collaborative production operation of multi-robot system. It consists of trajectory length L, maximum velocity $v_{\max}$, acceleration $a_{\max}$, and additive acceleration $j_{\max}$. Moreover, it enhances path diversity to approximate the global optimum by dynamically adjusting the above parameters, which is mathematically expressed as minimizing the number of observation matrix conditions. The motion stability is related to whether the industrial robot can maintain a stable motion state and accurately perform the scheduled tasks in the complex and changing industrial production environment. Therefore, the study proposes an OPP model for industrial robots based on the A-I-DE algorithm with respect to the track excitation mechanism and motion stability performance of industrial robots. Among them, the DE algorithm is the basic framework for realizing the track excitation of the PP model. Its algorithm flow is shown in Figure 1.

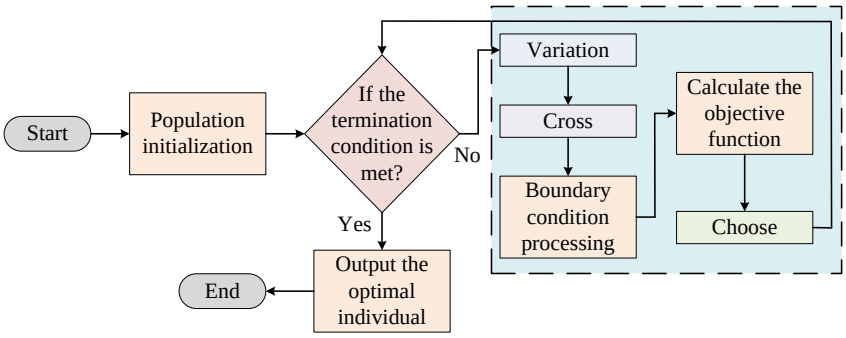


Figure 1: The process of the differential evolution algorithm.

In Figure 1, the DE generates candidate solutions by initializing the population, performs mutation, crossover, and selection operations sequentially, perturbs the individuals using difference vectors and greedily retains the superior solution, and iterates in a loop until convergence. The procedure first performs an initialization operation, giving each population member's dimension a value. Each individual is denoted by $x_{i,G}\left(i=1,2,\ldots NP\right)$. i denotes the number of the individual in the population. G is the number of evolutionary generations. NP represents the population size. Every population that has been randomly initiated is thought to follow a uniform distribution. It is set that the bounds of the parameter variables are $x_j^L < x_j < x_j^U$, at this time, the initialized individual assignments can be expressed by Equation (1).

$$x_{jx,0} = rand\left[0,1\right]*\left(x_j^U - x_j^L\right) + x_j^L \tag{1}$$

In Equation (1), $rand\left[0,1\right]$ denotes the real number that produces uniformity among $\left[0,1\right]$. x_j^U and x_j^L are the upper and lower limits of the parameter variables, respectively. After that, the mutation operation is performed, and for each target $x_{i,G}\left(i=1,2,\ldots NP\right)$, the mutation vector is generated as shown in Equation (2).

$$v_{i,G+1} = x_{r1,G} + F*\left(x_{r2,G} - x_{r3,G}\right) \tag{2}$$

In Equation (2), $r1$, $r2$, and $r3$ are randomly selected individual serial numbers. Each ordinal number is different from each other and cannot be the same as the vector ordinal number i , which must satisfy $NP \geq 4$. The operator for variation The scaling of the deviation variable is regulated by the real constant factor $F \in \left[0,2\right]$. DE adds a crossover operation to the interference parameter vectors to make them more diverse. The test vector is now displayed in (3).

$$\begin{cases} v_{i,G+1} = \left(u_{1i,G+1}, u_{2i,G+1}, \ldots u_{Di,G+1}\right) \\ u_{ji,G+1} = v_{ji,G+1} \ \ randb\left(j\right) \leq CR \ \ or \ \ j = rnbr\left(i\right) \\ u_{ji,G+1} = x_{ji,G+1} \ \ x! = 0 \ \ and \ \ j! = rnbr\left(i\right) \end{cases} \tag{3}$$

In Equation (3), The random number generator that generates $\left[0,1\right]$ has an j th estimate, which is represented by the $randb\left(j\right)$. To guarantee that $u_{i,G+1}$ receives at least one parameter from $v_{i,G+1}$, $rnbr\left(i\right) \in \left(1,2,\ldots,D\right)$ stands for a randomly chosen sequence, where D is the population dimension. CR denotes the crossover operator, which takes the value range $\left[0,1\right]$. The selection operation is performed after the variation and crossover operations are completed. In this process, the test vectors are compared with the target vector $x_{i,G}$ according to the greedy criterion and the superior one goes to the next generation. This process is inter-individual comparison, not group competition. If a solution outside the feasible domain, i.e., $u_{ji,G+1} < x_j^L$ or $u_{ji,G+1} > x_j^U$, is compiled during the mutation process, the operation shown in Equation (4) is performed.

$$u_{ji,G+1} = rand\left[0,1\right]*\left(x_j^U - x_j^L\right) + x_j^L \tag{4}$$

In Equation (4), the solution generated by the mutation is always within the bounds of the parameter variables by re-initializing the assignment. In summary, population initialization, mutation and crossover operations are the key to achieve computational accuracy and convergence speed [21]. Therefore, this study improves the three in the framework of DE and proposes the IDE algorithm. First, the population initialization is improved, and the distance between the optimal individuals and the initial population's members influences the DE algorithm's computation time. Random people don't have direction. The initial population is optimized by the optimal worst-case backward learning strategy to determine the direction of feasible solutions. Therefore, the initial individuals are close to the OS, which speeds up the convergence of the algorithm while maintaining the global search capability. The mathematical definition of the reverse learning strategy is shown in Equation (5).

$$x_{i,\text{opp}}^{(j)} = x_j^U + x_j^L - x_i^{(j)} \tag{5}$$

In Equation (5), $x_{i,\text{opp}}^{(j)}$ is the reverse solution of individual x_i in dimension j . x_j^U and x_j^L are the upper and lower bounds of the parameters, respectively. The study merges the original population P with the inverse population P_{opp} , and retains the highly adapted individuals as the optimized initial population. Second, the variation operator F 's is improved. The operator is too small, which reduces the population diversity. If it is too large, it will reduce the search efficiency and affect the accuracy of the OS. This study designs the adaptive

variation operator, which is calculated as shown in Equation (6).

$$\lambda = e^{1-\frac{g_{\max}}{g_{\max}+1-g}}, F = F_0 * 2^{\lambda} \tag{6}$$

In Equation (6), F_0 is the initial mutation operator. $g_{\max}$ is the maximum evolutionary algebra. g is the current evolutionary generation. The initial adaptive variation operator, $F = 2F_0$, can maintain individual diversity at the beginning. As the number of evolutionary generations increases, F gradually decreases and approaches F_0. The linear decay design is predicated on a balance between global exploration and local exploitation of the algorithm. This is achieved by dynamically adjusting the perturbation strength, thereby avoiding local optimization in the early stages and suppressing oscillations in the later stages. In addition, the value of the crossover operator CR will directly affect the convergence time of the algorithm, resulting in missing the global OS or reducing the efficiency of the algorithm. Similarly, the study designs adaptive crossover operator, which is calculated as shown in Equation (7).

$$CR = CR_{\min} + \frac{k}{g_{\max}}\left(CR_{\max} - CR_{\min}\right) \tag{7}$$

In Equation (7), $R_{\max}$ and $R_{\min}$ denote the maximum and minimum values of the crossover operator, respectively. In terms of specific parameter settings, the study sets the initial value of the adaptive variance factor of IDE to 0.5 and gradually decreases to 0.3 with the number of iterations. The dynamic adaptive adjustment range of the crossover rate is 0.3-0.9. The adaptive variance factor enhances the global search ability in the initial stage and improves the convergence accuracy in the later stage. The dynamic crossover rate balances the diversity of the populations with the convergence speed. Through the hyperparameter adaptive dynamic adjustment mechanism in Equations (5) and (6), the model is able to avoid the subjectivity of manual parameter tuning, while adapting to the convergence requirement in different environments to ensure the robustness of the algorithm. Improving the identification accuracy and reducing the condition number of the observation matrix are the main goals of optimizing the PP algorithm's excitation trajectory criterion. Therefore, the flow of IDE to optimize the trajectory excitation parameters is shown in Figure 2.

As illustrated in Figure 2, IDE incorporates adaptive variance factors and dynamic crossover rates, integrates optimal backward learning to initialize the population, generates incentive parameters through constraint validation and cyclic optimization, and iteratively updates them until the termination conditions are met. First, IDE initializes the population, and the incentive track parameter is the population dimension. Then, the individual objective value of the current population is calculated, which is quantified by a weighted multi-objective function as shown in Equation (8).

$$f_{\text{obj}} = w_1 \cdot \frac{L}{L_{\max}} + w_2 \cdot \frac{T}{T_{\max}} + w_3 \cdot \sum\left|\frac{d^2\theta}{dt^2}\right| + w_4 \cdot \frac{1}{d_{\min}} \tag{8}$$

In Equation (8), L is the planning path length. $L_{\max}$ is the maximum theoretical path length of the scene. T is the path execution time. $T_{\max}$ is the maximum time threshold allowed by the task. $\sum\left|\frac{d^2\theta}{dt^2}\right|$ is the sum of the absolute values of the joint angular acceleration, which measures the smoothness of the trajectory. $d_{\min}$ is the minimum distance between the path and the obstacle. Furthermore, the weight coefficients satisfy $w_1 + w_2 + w_3 + w_4 = 1$, and the study is set to $w_1 = 0.4, w_2 = 0.3, w_3 = 0.2, w_4 = 0.1$. If the current individual does not meet the constraints, the objective value is set. If it does, compare the initial value and the current individual's objective function value. If the initial value is greater, the initial value of the individual is updated, otherwise genetic operations are performed to generate new individuals, i.e. mutation, crossover, selection. If the new individual value is greater than the original individual value, the population is reinitialized. At this point, the number of evolutionary generations is increased by one, and the cycle optimization is restarted until the OS for the population is obtained, or the evolution reaches the maximum number of generations.

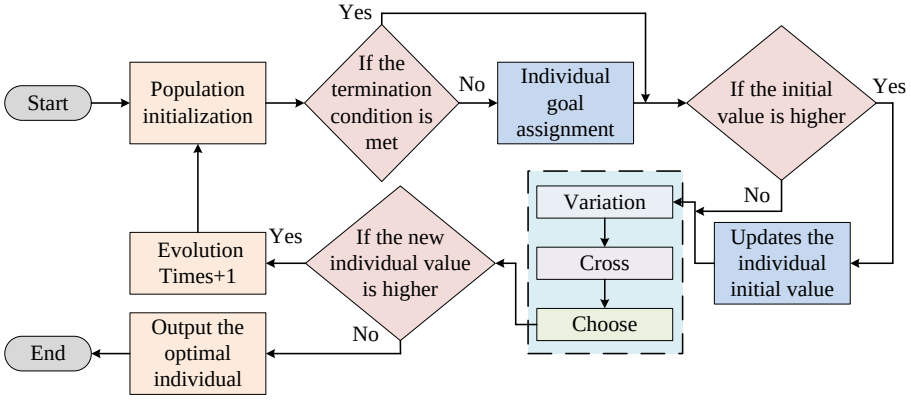


Figure 2: The process of optimizing the trajectory excitation parameters in the IDE.

3.2 A-I-DE path planning modeling for motion step problems

Orbital excitation mechanisms and motion stability are at the core of learning and adaptive path optimal planning for industrial robots. The study proposes that IDEs optimize the design of the excitation mechanism. However, in industrial dynamic environments or collaborative production operations, the operating speed of industrial robots often undergoes jumping changes. This process will generate huge mechanical stress, which affects the motion accuracy and stability of industrial robots, and thus poses a threat to industrial production [22]. IASCV makes the velocity profile of industrial robots smoother by processing the trajectory in segments, reduces the impact when starting and stopping, and improves the stability and accuracy of the motion. Therefore, the research addresses the velocity step problem, combines IASCV to further optimize the IDE, and proposes A-I-DE. Among them, the curve model of IASCV is shown in Figure 3.

In Figure 3, the IASCV is predicated on the enhancement of the ASCV, with the absence of mid-beginning and end-end acceleration. The red point demarcates the transition moment of the acceleration phase to ensure a seamless transition of motion. The green point identifies the critical state of uniform and variable speeds to regulate the magnitude of the velocity step. Furthermore, the blue point calibrates the point of cumulative inversion of the displacement to inhibit motion oscillations. IASCV restricts the second-order derivative continuity of the velocity profile by adding acceleration constraints to the segmented trajectory and dynamic time allocation. This ensures that the acceleration in each phase transitions smoothly to avoid mechanical shocks. The parameters when IASCV runs are mainly trajectory length L, initial velocity v_s and final velocity v_e, maximum acceleration $a_{\max}$, maximum deceleration $-a_{\max}$, initial acceleration a_s, and final acceleration a_e. It constructs a curve model by using the parameters and outputs, and the model parameter changes can be calculated based on the change curves when the curve trajectory runs. For example, when $0 \le t \le t_1$, the parameter changes are shown in Equation (9).

$$\begin{cases} a_{(t)} = a_s + j_{\max} * \tau_1 \\ v_{(t)} = v_s + a_s * \tau_1 + 0.5 j_{\max} * \tau_1^{\,2} \\ S_{(t)} = v_s * \tau_1 + 0.5 a_s * \tau_1^{\,2} + (j_{\max} * \tau_1^{\,3})/6 \end{cases} \tag{9}$$

In Equation (9), $j_{\max}$ denotes the maximum additive acceleration. $t_k = \sum_{i=1}^{k} T_i, k = 1,2,\ldots,7$ denotes the elapsed point moments of each motion phase of the trajectory. τ denotes the local time coordinates, where $\tau_1 = t, \tau_k = t - t_{k-1}, k = 2,\ldots 7$. Similar to ASCV, IASCV finds the values of the running time T for different phases based on the input parameters. Its algorithm flow is shown in Figure 4.

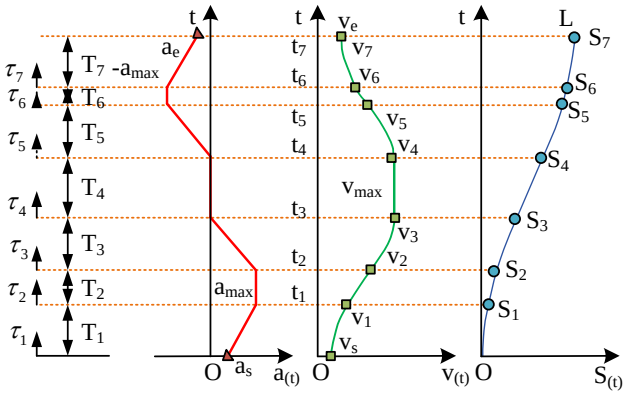


Figure 3: Improved asymmetric S-curve model.

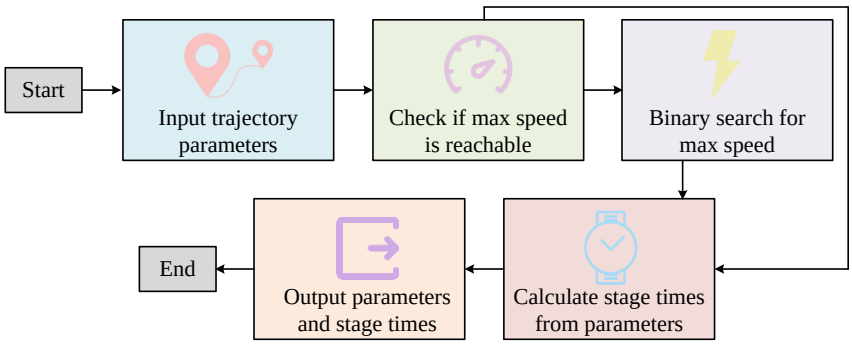


Figure 4: The process of improved asymmetric s-curve velocity profile.

In Figure 4, the known trajectory parameters, including L, v_s, v_e, etc., are firstly inputted into the algorithm. After that, the segmental displacement trajectory $L = L_a + L_d$ is calculated for the initial and terminal velocities to reach the maximum uniform velocity $v_{\max}$. In this case, L_a and L_d are calculated as shown in Equation (10).

$$\begin{cases} L_a = \dfrac{0.5\left(v_{\max}^2 - v_s^2\right)}{a_{\max}} + \dfrac{0.5\left(v_{\max}^2 + v_s\right)}{j_{\max}} & v_{\max} - v_s \geq \dfrac{a_{\max}^2}{j_{\max}} \\ L_a = \left(v_{\max} + v_s\right) * \left(\dfrac{v_{\max} - v_s}{j_{\max}}\right)^{0.5} & v_{\max} - v_s < \dfrac{a_{\max}^2}{j_{\max}} \\ L_d = \dfrac{0.5\left(v_{\max}^2 - v_e^2\right)}{d_{\max}} + \dfrac{0.5\left(v_{\max}^2 + v_e\right)}{j_{\max}} & v_{\max} - v_e \geq \dfrac{d_{\max}^2}{j_{\max}} \\ L_d = \left(v_{\max} + v_e\right) * \left(\dfrac{v_{\max} - v_e}{j_{\max}}\right)^{0.5} & v_{\max} - v_e < \dfrac{d_{\max}^2}{j_{\max}} \end{cases} \tag{10}$$

In Equation (10), $d_{\max}$ is the maximum acceleration in the deceleration phase. If the shortest trajectory required to change v_e from v_s is less than L, it means that v_s and v_e satisfy the planning conditions. Then, based on the displacement length and the actual v_s and v_e, the actual $v_{\max}$ is calculated by dichotomization and proceeds to the next step. Otherwise, the bisection operation is skipped, and the different stages running time T is calculated directly based on the actual track parameters. Then the different phase running times T are summed up to obtain the shortest total running time. Finally, the actual values of v_s, v_e, and $v_{\max}$ are output, as well as the values of each phase and the shortest total running time. For IASCV, the study sets the maximum additive acceleration to be set at 1000 cm/s3, the maximum acceleration to be 500 cm/s2, and the initial acceleration to be 50 cm/s2 during PP. The IASCV parameter ensures the smooth transition of the velocity profile by constraining the upper limit of additive acceleration and acceleration to reduce the mechanical shocks and improve the stability of the motion. On this basis, the study combines IASCV with IDE to propose the A-I-DE PP model. Its algorithm flow is shown in Figure 5.

In Figure 5, A-I-DE achieves collaborative optimization through a bidirectional closed-loop feedback mechanism. In this mechanism, IDE generates path parameters, and then IASCV detects velocity anomalies in real time and triggers dynamic adjustment of the parameters with feedback. Both of these processes are iteratively corrected between path generation and velocity smoothing. This ensures a dynamic balance between global optimization and local motion constraints. That is, A-I-DE is divided into two main components: the computation of the overspeed region and the replanning of the trajectory. The former is used for the preliminary detection of individual overspeeds, while the latter is based on multi-area computation to realize accurate trajectory correction, taking into account both global safety and local optimization needs. The A-I-DE algorithm first generates the initial path through IDE PP and identifies the peaks of velocity mutation through the velocity extreme value search module. The overspeed determination link is responsible for detecting any potential violations of the dynamics constraints. In the event of overshoots, the continuous risk period is divided into single or multiple overspeed regions. The trajectory correction module is tasked with dynamically adjusting the excitation parameters and relaying these adjustments back to the IDE for iterative optimization. The corrected trajectory is looped again until all overspeed regions are eliminated, and the final output is a smooth velocity profile. This sequence of operations helps to ensure the safety and efficiency of the robot's motion while optimizing time.

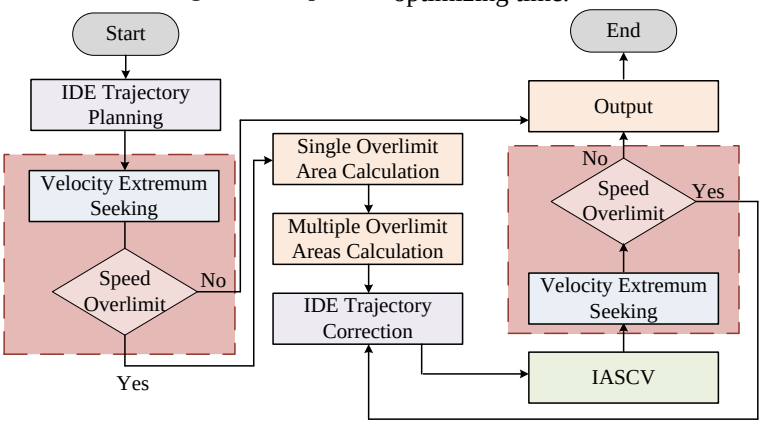


Figure 5: The structure of A-I-DE.

4 Results

For verifying the effectiveness and superiority of the A-I-DE algorithm proposed in the study, the study synthesizes the theoretical foundation and algorithm analysis mentioned above, and carries out simulation simulation running experiments and real environment motion experiments on the industrial robot model under different algorithms. Moreover, the experimental results are analyzed in detail to compare their PP accuracy and motion stability.

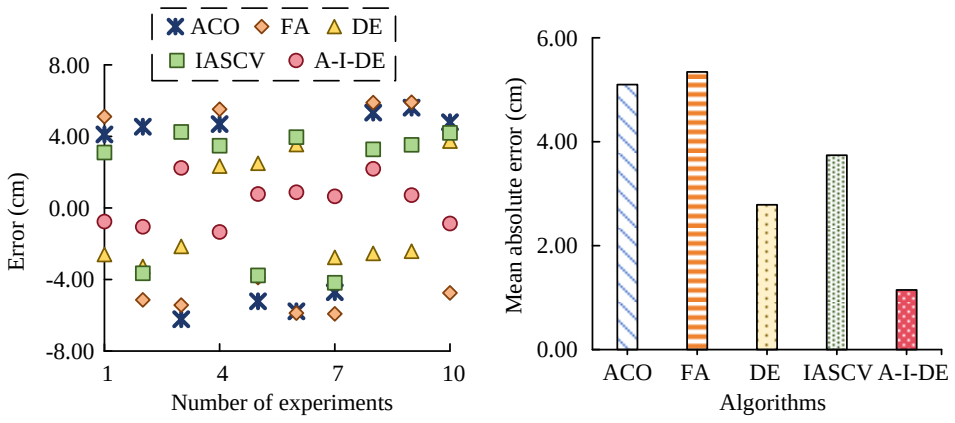
4.1 Simulation simulation operation experiment

The industrial robot PP model's application environment is combined with the simulation running experiments to provide a compatible system development environment that is separated into a software environment and a hardware environment. The detailed configuration is shown in Table 2.

In Table 2, the study selects Windows 10 as the operating system and the robot operating system as the software platform. Gazebo 3D physical simulation platform is used to build the simulation scene, and the industrial robot model is constructed through the unified robot description format. The simulation environment is a 10m×6m rectangular space containing 12 obstacles: 8 static cylinders (diameter 0.5-1.2m) and 4 dynamic moving cubes (side length 0.8m). The robot has a maximum linear velocity of 2m/s, a safety distance threshold of 0.3m, and a joint torque limit of 80%. Moreover, the constraint values are set based on the robot dynamics model and industry standards. Furthermore, the study selects ant colony optimization algorithm, firefly algorithm, DE, and IASCV as the comparison methods and named ACO, FA, DE, and IASCV respectively. A-I-DE is taken as the object of the study and named A-I-DE. These baseline methods can systematically verify the comprehensive performance of this algorithm in multi-dimensional performance metrics. They represent typical schemes in the fields of traditional heuristic search, population intelligence optimization, and speed smoothing, respectively, and cover the mainstream technological routes of PP. Moreover, the pheromone volatility of ACO is 0.5 and the heuristic factor is 2.0.The attractiveness of FA is 0.9 and the step size is 0.3.The variance factor of DE is 0.6 and the crossover rate is 0.8.The maximum acceleration of IASCV is 500 cm/s² and the acceleration of additive acceleration is 1000 cm/s³. Moreover, the parameters are set based on the standard literature recommended values to ensure the fairness. In this study, two points are firstly selected as the starting point (0,0) and the target point (5m, 3m) of the industrial robot in the complex environment simulated by the simulation. Then, the robot is controlled to start from the starting point and finally reach the target point, which is repeated 10 times with the confidence interval set to 95%. Among these, Figure 6 displays the outcomes of the distance inaccuracy between the industrial robot and the TP point once it has stopped operating.

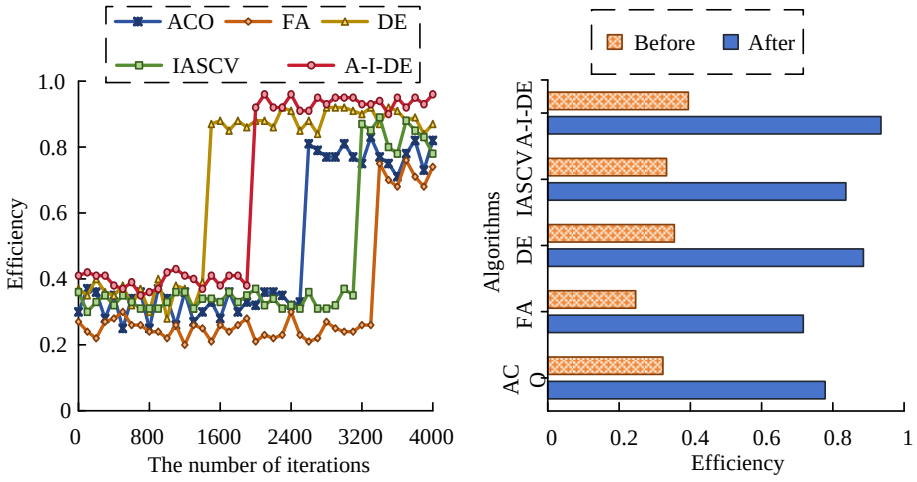
Table 2: System development environment.

	System development environment
Hardware environment	Intel(R) Core(TM) i7-8700 Installed memory 16.00GB
	NVIDIA RTX2080Ti
Software environment	Windows 10 is running on a 64-bit operating system
	Robot Operate System
	Gazebo
	Unified Robot Description Format



(a) Target point error (b) Mean absolute error

Figure 6: Difference in target point error.

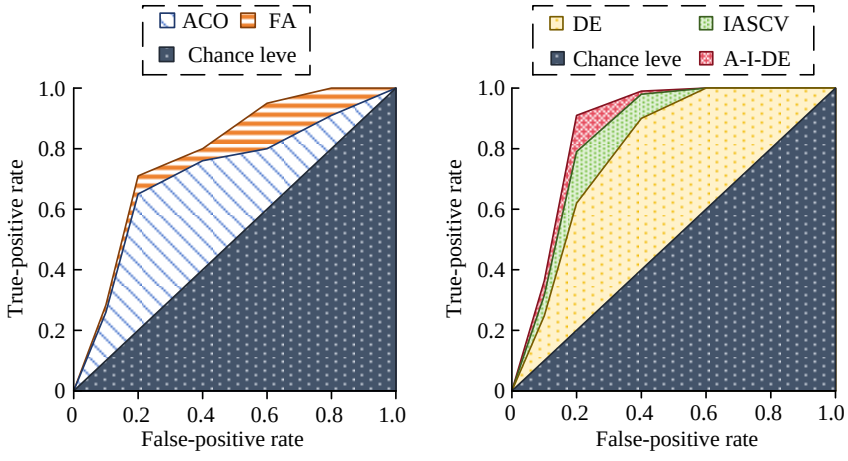


(a) Differences in learning performance (b) Average precision differences

Figure 7: Iterative learning performance experiments.

In Figure 6(a), when the industrial robot stops moving, ACO has the largest change in error from the TP. The maximum error magnitude of change is 2.11 cm, followed by FA. Among them, the error fluctuates between 3.89 cm and 5.92 cm. The errors of DE and IASCV varied less, with the maximum change magnitude of 1.59 cm for both. The error of A-I-DE varies the least, with the error fluctuating between 0.65 cm and 2.24 cm. In Figure 6(b), in terms of the average error, A-I-DE has the smallest error of 1.15 cm. It is followed by DE with an average error of 2.79 cm. IASCV has an average error of 3.74 cm. ACO and FA have the largest errors, with a mean value of 5.10 cm and 5.34 cm, respectively. The outcome displays that the accuracy of the A-I-DE's PP is much higher than the traditional algorithm. In addition, the study selects the operational dataset as the training set and conducts iterative experiments on the model. The study verifies its learning performance by comparing its PP efficiency (path success rate).The results of which are shown in Figure 7.

In Figure 7(a), DE completes convergence at 1500 iterations and has the fastest convergence rate. ACO completes convergence at 2600 iterations. FA and IASCV converge the slowest, completing convergence at 3400 and 3200 iterations, respectively. Whereas, A-I-DE has the second fastest convergence rate after DE, completing the convergence at 2000 iterations. In Figure 7(b), before and after convergence, FA has the lowest average efficiency of 0.25 and 0.72, respectively. It is followed by ACO, with average efficiencies of 0.32 and 0.78 before and after convergence, respectively. The efficiencies before and after convergence of DE and IASCV are in the ranges of 0.28-0.40 and 0.78-0.92, respectively. Whereas A-I-DE has the highest efficiency before and after convergence with mean values of 0.39 and 0.94 respectively. Compared to the fast local convergence of DE, A-I-DE prioritizes the strengthening of the global search capability and expands the exploration area of the solution space through dynamic parameter adjustment. This ensures that the local optimum in complex PP is avoided, and the moderate iteration cost is traded for high accuracy and stability. For the overall PP performance of the model, the study conducts receiver operating characteristic curve (ROC) and area under curve (AUC) analysis. The results are shown in Figure 8.



(a) ROC for M1 and M2 (b) ROC for DE, IASCV and A-I-DE

Figure 8: Analysis of ROC for different algorithms.

Table 3: Differences in computational complexity.

Method	ACO	FA	DE	IASCV	A-I-DE
Single planning time（ms）	15.2	12.7	9.4	8.1	6.3
peak memory usage（MB）	48.3	40.6	35.2	30.5	26.8
CPU utilization（%）	82.5	75.8	68.3	62.4	58.9

Table 4: Comparison of model motion stability.

Method	TTE (cm)	SSF (time)	ASF (time)	PSI (%)	TS (°/m)
ACO	6.32	9	12	68.5	8.2
FA	6.78	8	11	70.2	7.8
DE	4.15	6	8	82.7	5.4
IASCV	3.89	5	7	88.3	4.9
A-I-DE	1.05	2	3	95.6	2.1

In Figure 8, in the ROC, the false positive rate (FPR) grows fastest at 0-0.2 for the true positive rate (TPR). At 0.2-0.6, the growth of TPR becomes slower and gradually stabilizes after 0.6. At the same FPR, ACO had the lowest TPR, followed by FA. At this time, the TPRs of DE and IASCV are slightly higher than that of FA, but the TPR of A-I-DE is the largest. In addition, there is a similar trend of difference in the AUC values of the different models. IASCV has an AUC value of 0.85, whereas A-I-DE has the highest AUC value of 0.87. The AUCs of DE and FA are 0.80 and 0.78, respectively, and the AUC of ACO is the lowest value of 0.72. The benchmark value of AUC in PP tasks is typically 0.8-0.9 [23], and experiments show that A-I-DE has high reliability and outperforms most traditional algorithms.

To further explore the computational complexity of the proposed algorithm, the study selects the single planning elapsed time (ms), peak memory occupancy (MB), and CPU utilization (%) metrics for experimentation. The results are shown in Table 3.

In Table 3, the single planning elapsed time of A-I-DE (6.3ms) is 58.5% lower than that of ACO (15.2ms), due to the low computational overhead of the adaptive mutation and crossover operators. The peak memory usage (26.8MB) is 23.9% lower than that of DE (35.2MB), due to the optimization of IDE's population initialization to reduce redundant storage. The CPU utilization (58.9%) is significantly lower than the comparison methods, thanks to IASCV's speed smoothing mechanism, which reduces the frequency of real-time replanning. The results show that A-I-DE has significant advantages in scenarios with limited computational resources.

In addition, to verify the motion stability of the model in the simulated environment, the study selects trajectory tracking error (TTE), speed step frequency (SSF), acceleration step frequency (ASF), posture stability index (PSI) and trajectory smoothness (TS) metrics for comparison. The PSI and TS metrics are compared and the results are shown in Table 4.

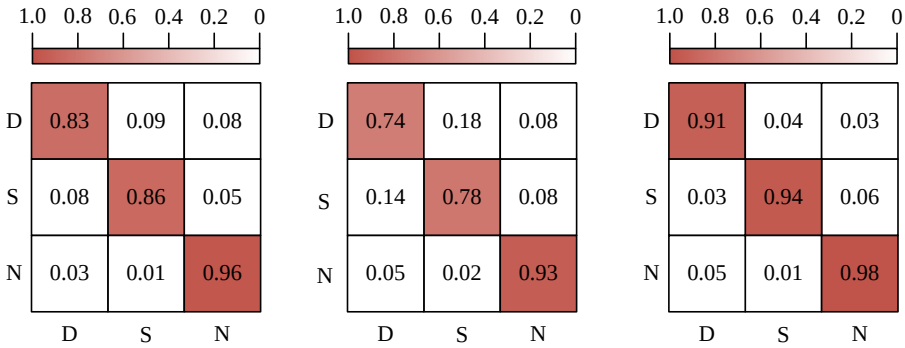
In Table 4, the TTE of A-I-DE (1.05 cm) is 83.4% lower than that of ACO (6.32 cm), due to the closed-loop parameter coupling of IDE and IASCV, which suppresses the cumulative displacement deviation. Its SSF (2 times) is only 33% of that of DE (6 times). ASF (3 times) is 57% less than that of IASCV (7 times), attributed to the adaptive parameters of IDE suppressing the path abruptness. In addition, the PSI of the A-I-DE is 95.6%, which is significantly higher than that of the IASCV (88.3%) due to the IDE's optimization of path excitation parameters to reduce fuselage inclination. Trajectory smoothness (TS=2.1°/m) is improved by 61.1% compared to DE (5.4°/m) thanks to the segmentation plus acceleration constraints of IASCV, confirming the overall advantages of A-I-DE in terms of kinematic stability and dynamic performance.

4.2 Real environment motion experiments

Robot operation in simulation is an important standard for measuring the operation of industrial robots. However, influenced by many uncontrollable factors, the operation status of industrial robots in the actual environment often differs from the simulation simulation. Therefore, the study uses an industrial production workshop as the experimental environment, with workshop equipment as static obstacles and workers and other industrial robots as dynamic obstacles, to conduct experiments. Each set of experiments is repeated 10 times to verify the stability. Moreover, the study selects Universal Robots UR5 collaborative robot (public parameters, https://www.universal-robots.com/products/ur5-robot/) as the actual test robot. The starting point is located at the origin of the workshop coordinates (0,0), the target point is set as the material loading and unloading area (12.5m, 8.2m), and the theoretical shortest path between the two points is 15.3 m. The actual running path is recorded in real time by the laser tracking instrument (API T3) and the accumulated mileage is calculated, and the deviation of the path length is quantified by the Euclidean distance

integration method. In addition, the study selects DE and IASCV, which perform better in simulation and simulation running experiments, as the comparison method, and A-I-DE is used as the object of the study. The study firstly verifies the operational efficiency of the model when facing no obstacles (No, N), static obstacles (Static, S), and dynamic obstacles (Dynamic, D). The static obstacles are set as 8 fixed cylinders, and the dynamic obstacles are set as 5 randomly moving spheres (with operational constraints consistent with those of the robot). Furthermore, the experiments use 10 sets of random seeds to generate the initial positions of the obstacles to ensure the generalizability of the results. Figure 9 displays the outcomes of the experiment.

In Figure 9, the model has the highest PP efficiency in an accessible environment, followed by static barriers, and the model has the lowest efficiency in a dynamic barrier environment. In Figure 9(a), the efficiency of DE is 0.96 in the accessible environment, and 0.83 and 0.86 in static and dynamic barriers, respectively. In Figure 9(b), IASCV has the lowest efficiency of PP in the real environment. Its planning efficiency is 0.93, 0.78, and 0.74 in the three barrier environments, respectively. In Figure 9(c), the planning efficiency of A-I-DE in all three obstacle environments is high, and the overall efficiency is above 0.91. Among them, the model has the highest PP efficiency in the barrier-free environment, reaching 0.98. This means that the PP success rate of A-I-DE reaches 98%, indicating a nearly redundancy-free search, which significantly improves the efficiency of continuous operation in industrial scenarios. Moreover, the critical tilting angle of conventional industrial robots is about 25°, and they will fall down if the critical tilting angle is exceeded. Therefore, the study selects 10 time points during the movement of the industrial robot. Moreover, the angle ∂ between the body of the industrial robot and the horizontal plane at that time point is used as the basis for judgment. The smaller the tipping angle during the motion of the industrial robot, the higher the motion stability. Table 5 displays the findings.

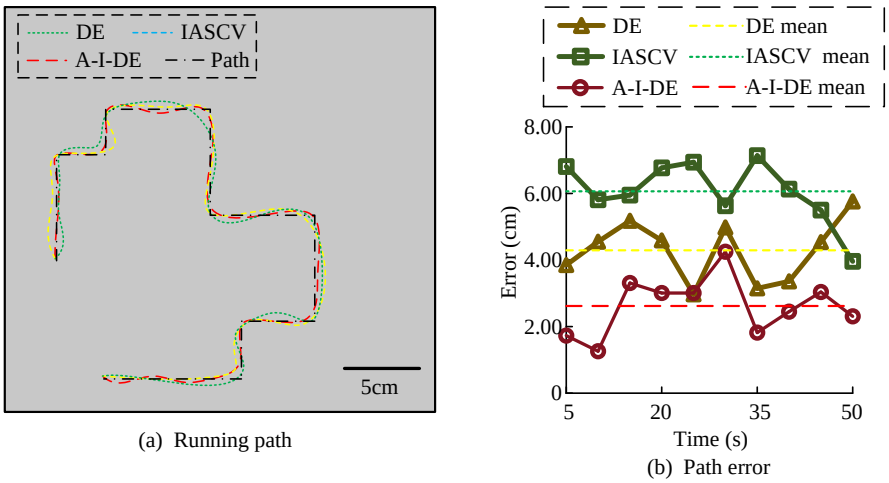


(a) Analysis for DE (b) Analysis for IASCV (c) Analysis for A-I-DE

Figure 9: Confusion matrix analysis for operational efficiency.

Table 5: Athletic stability of different algorithms running.

Points	Dump angle (°)			Motion stability (%)		
	DE	IASCV	A-I-DE	DE	IASCV	A-I-DE
1	11.85	5.90	4.82	52.60	76.40	80.72
2	13.76	7.75	7.75	44.96	69.00	69.00
3	12.74	8.75	4.50	49.04	65.00	82.00
4	12.90	8.16	6.17	48.40	67.36	75.32
5	13.78	7.92	7.42	44.88	68.32	70.32
6	11.48	8.85	6.73	54.08	64.60	73.08
7	14.10	5.95	4.63	43.60	76.20	81.48
8	12.31	6.83	4.52	50.76	72.68	81.92
9	13.63	6.41	6.45	45.48	74.36	74.20
10	11.51	6.17	7.73	53.96	75.32	69.08
Mean	12.81	7.27	6.07	48.78	70.92	75.71



(a) Running path (b) Path error

Figure 10: Path trajectory and error.

In Table 5, DE has the largest dumping angle ranging from 11.48°-14.1°0 with a mean value of 12.81°. IASCV has a range of dumping angle from 5.90° to 8.85° with a mean value of 7.27°. The mean dumping angle of A-I-DE is 6.07° and the range fluctuates between 4.50°-7.75°. In addition, according to the calculated equation $Ms = \frac{25-\partial}{25} * 100\%$, the A-I-DE has the highest motion stability ranging from 69.00% to 82.00% with a mean value of 75.71%. This is followed by IASCV with stability ranging from 64.60%-76.40% with a mean stability of 70.92%. The stability of DE is the lowest, with an overall range of 43.60%-54.08% and a mean value of 48.78%. The experimental results show that the robot under A-I-DE control operates more stably with a low probability of error in real environment applications. The study also examines the discrepancy between the industrial robot's real motion path and its preset path in real time to confirm the model's correctness and stability in real-world operations. Figure 10 displays the outcomes of the experiment.

In Figure 10(a), the IASCV has a large offset relative to the preset trajectory in the control motion, and the actual motion trajectory is curved. DE has a smaller offset relative to the preset trajectory, while the actual trajectory of A-I-DE fits the preset path better. In Figure 10(b), the errors of the actual paths of DE and IASCV with respect to the preset paths are between 2.97cm-5.76cm and 3.96cm-7.14cm, respectively. Their mean values are 4.29cm and 6.06cm, respectively. At this time, the errors of the paths of A-I-DE are between 1.26-4.25cm, and the mean error is 6.62. Moreover, except for some external time points, the errors of the other time points are smaller than those of DE and IASCV. The experimental results show that the industrial robot under A-I-DE control has the best operational accuracy and stability in the real environment. It can be concluded from the aforementioned findings that the study on the improvement of the DE algorithm in the operations of initializing population, mutation, and crossover, as well as combining it with the IASCV algorithm for optimization and other improvement measures, is effective. The implementation of these measures can lead to enhanced performance of industrial robots in terms of efficiency and stability of PP.

Moreover, to validate the comprehensive performance advantages of the proposed methods of the study in complex environments, the study selects methods from the literature [9], [11], [13], [16], [18], and [19] for comparison with A-I-DE. These methods are IA-DWA, Bi-STP, HGFA, PTSMA, DRL-PER, and MQL-ES and the results are shown in Table 6.

Table 6: Comparison of A-I-DE and recent advances.

Method	IA-DWA	Bi-STP	HGFA	PTSMA	DRL-PER	MQL-ES	A-I-DE
Convergence iterations	2600	3400	2800	2100	3200	2500	2000
Path error (cm)	5.1	3.74	5.34	4.29	6.06	4.15	1.15
Motion stability (%)	48.78	70.92	65.2	72.5	68.3	74.8	75.71
Computational efficiency (times/s)	38.5	22.1	45.3	50.6	18.9	40.7	65.4
AUC	0.72	0.85	0.78	0.82	0.87	0.83	0.94

Table 7: Comparison of model integrated energy efficiency performance.

Method	MOEC （W·h）	TTEC （J/m）	HRPEC （W）	SPTEC （W·s）	TSEC （kW·h）

IA-DWA	12.5	8.2	95.3	1.9	1.28
Bi-STP	10.8	7.5	88.7	1.7	1.15
HGFA	11.2	7.8	92.1	1.8	1.21
PTSMA	9.6	6.9	84.5	1.5	0.98
DRL-PER	15.3	9.1	102.4	2.3	1.45
MQL-ES	8.9	6.3	78.9	1.3	0.85
A-I-DE	7.2	5.1	72.4	1	0.68

As shown in Table 6, the number of convergence iterations (2000) of A-I-DE is significantly lower than that of the comparison methods (2100-3400), thanks to the dynamic parameter adjustment strategy of the adaptive variational and crossover operators. Its path error (1.15 cm) is 72.3% lower than that of the optimal comparison method (4.15 cm for MQL-ES), which is due to the IDE's orbit excitation mechanism that optimizes the parameter. Motion stability (75.71%) is better than other algorithms because IASCV's smooth velocity profile effectively suppresses mechanical stress. The computational efficiency (65.4 times/sec) and the TPR (AUC=0.94) are attributed to the improvement of the population initialization and the dynamic replanning mechanism in the overspeed range, which avoids the redundant search.

Finally, the study validates the comprehensive model by comparing the research model with the latest work of model operation energy consumption (MOEC), trajectory tracking energy consumption (TTEC), hardware resource peak energy consumption (HRPEC), single planning time energy consumption (SPTEC) and total system energy consumption (TSEC). Hardware resource peak energy consumption (HRPEC), single planning time energy consumption (SPTEC), and Total system energy consumption (TSEC) are introduced to verify the comprehensive energy efficiency performance and industrial applicability of the model. The experimental results are shown in Table 7.

In Table 7, the model running energy consumption of A-I-DE (7.2W-h) is reduced by 19.1% compared to the latest work MQL-ES (8.9W-h) due to the adaptive parameters to reduce the redundant computation. The trajectory tracking energy consumption (5.1J/m) is reduced by 18.9% thanks to the IASCV speed smoothing to reduce the frequent motor starts and stops. The peak power consumption of hardware resources of A-I-DE (72.4W) is 29.3% less than DRL-PER (102.4W) due to IDE population optimization to reduce memory and CPU load. The energy consumption (1.0 W-s) of a single design is only 52.6% of IA-DWA, and the total system energy consumption (0.68 kW-h) is optimized by 20.0% compared to MQL-ES (0.85 kW-h). The experimental results confirm the energy saving advantages of the A-I-DE.

5 Discussion and conclusion

The IDE algorithm and the IASCV algorithm were integrated in the study to address the issues of low efficiency and poor motion stability of conventional PP algorithms for industrial robots. The A-I-DE PP model was then created. The model improved the accuracy and stability of industrial robot operation by optimizing the track excitation mechanism and solving the velocity step problem. The experimental results revealed that the average absolute error of A-I-DE was 1.15cm in the simulation simulation experiments, with an error range between 0.65cm-2.24cm. The errors of the other algorithms ranged from 2.15cm-5.34cm. When iterating in the operational data set, the average efficiency of A-I-DE before and after convergence was 0.39 and 0.94, respectively. Whereas the efficiencies of other algorithms before and after convergence were in the range of 0.20-0.43 and 0.68-0.92, respectively. Moreover, in the ROC analysis, the AUC value of A-I-DE was 0.87, whereas the other algorithms' AUC values were in the range of 0.70-0.85. In addition, in the actual detection, the detection efficiency of IRAU-DSC was above 0.91, while the efficiency of other algorithms ranged from 0.71-0.85. In addition, in the actual environmental motion experiments, the efficiency of A-I-DE in different environments ranged from 0.91-0.98, while the efficiency of other algorithms ranged from 0.74-0.96. However, the average tipping angle and average stability of A-I-DE were 6.07° and 75.71%, respectively, during the actual motion. For other algorithms, the dumping angle and stability were in the range of 5.90°-14.10° and 43.60%-76.40%, respectively. Moreover, the real-time error of the path of A-I-DE was between 1.26cm-4.25cm. The average errors of other algorithms were 4.29cm and 6.06cm, respectively. In conclusion, the study offers practical applications for enhancing the precision and reliability of industrial robot PP. However, the PP model proposed in the study focuses on single-segment paths, which may lead to cumulative errors at trajectory breakpoints due to the temporal constraint transfer characteristic of the parameter coupling mechanism in the continuous operation of multiple goal points. Additionally, the model suffers from the problem of nonlinear growth of computational complexity in complex topological environments. To address this limitation, in future work, the dynamic parameter decoupling strategy of trajectory articulation between target points, combined with incremental learning to optimize the subpath incentive mechanism, will be used to construct a hierarchical optimization framework to achieve global stability control of multistage continuous planning.

References

[1] Fushu Luan, Xinhui Yang, Yang Chen, and Paulo José Regis. Industrial robots and air environment: A moderated mediation model of population density and energy consumption. Sustainable Production and Consumption, 30(1):870-888, 2022. https://doi.org/10.1016/j.spc.2022.01.015

[2] Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, and Enrique Fernandez-Blanco. A review of artificial intelligence applied to path planning in UAV swarms. Neural Computing and Applications, 34(1):153-170, 2022. https://doi.org/10.1007/s00521-021-06569-4

[3] Lin Xu, Maoyong Cao, and Baoye Song. A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm. Neurocomputing, 473(1):98-106, 2022. https://doi.org/10.1016/j.neucom.2021.12.016

[4] Zhibin Li, Shuai Li, and Xin Luo. Using quadratic interpolated beetle antennae search to enhance robot arm calibration accuracy. IEEE Robotics and Automation Letters, 7(4):12046-12053, 2022. https://doi.org/10.1109/LRA.2022.3211776

[5] Steffan Lloyd, Rishad A. Irani, and Mojtaba Ahmadi. Fast and robust inverse kinematics of serial robots using Halley's method. IEEE Transactions on Robotics, 38(5):2768-2780, 2022. https://doi.org/10.1109/TRO.2022.3162954

[6] Xiaolong Wang, Jianfu Cao, Ye Cao, and Feng Zou. Energy-efficient trajectory planning for a class of industrial robots using parallel deep reinforcement learning. Nonlinear Dynamics, 113(8):8491-8511, 2024. https://doi.org/10.1007/s11071-024-10510-4

[7] QingyangWu. Sustainable growth through industrial robot diffusion: Quasi-experimental evidence from a Bartik shift-share design. Economics of Transition and Institutional Change, 31(4):1107-1133, 2023. https://doi.org/10.1111/ecot.12367

[8] Chengmin Zhou, Bingding Huang, and Pasi Fränti. A review of motion planning algorithms for intelligent robots. Journal of Intelligent Manufacturing, 33(2):387-424, 2022. https://doi.org/10.1007/s10845-021-01867-z

[9] Yonggang Li, Rencai Jin, Xiangrong Xu, Yuandi Qian, Haiyan Wang, Shanshan Xu, and Zhixiong Wang. A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach. IEEE Access, 10(1):57736-57747, 2022. https://doi.org/10.1109/ACCESS.2022.3179397

[10] Lisha Dong. Improved A* algorithm for intelligent navigation path planning. Informatica, 48(10):181-194, 2024. https://doi.org/10.31449/inf.v48i10.5693

[11] Valentin N. Hartmann, Andreas Orthey, Danny Driess, Ozgur S. Oguz, and Marc Toussaint. Long-horizon multi-robot rearrangement planning for construction assembly. IEEE Transactions on Robotics, 39(1):239-252, 2022. https://doi.org/10.1109/TRO.2022.3198020

[12] Dawei Sun, Jingkai Chen, Sayan Mitra, and Chuchu Fan. Multi-agent motion planning from signal temporal logic specifications. IEEE Robotics and Automation Letters, 7(2):3451-3458, 2022. https://doi.org/10.48550/arXiv.2201.05247

[13] Ting-Wei Zhang, Guang-Hui Xu, Xi-Sheng Zhan, and Tao Han. A new hybrid algorithm for path planning of mobile robot. The Journal of Supercomputing, 78(3):4158-4181, 2022. https://doi.org/10.1007/s11227-021-04031-9

[14] Jie Zan, Pengtao Ku, and Shoufeng Jin. Research on robot path perception and optimization technology based on whale optimization algorithm. Journal of Computational and Cognitive Engineering, 1(4):201-208, 2022. https://doi.org/10.1109/ACAIT53529.2021.9731150

[15] Yun Liu, Ali Asghar Heidari, Zhennao Cai, Guoxi Liang, Huiling Chen, Zhifang Pan, Abdulmajeed Alsufyani, and Sami Bourouis. Simulated annealing-based dynamic step shuffled frog leaping algorithm: Optimal performance design and feature selection. Neurocomputing, 503(3):325-362, 2022. https://doi.org/10.1016/j.neucom.2022.06.075

[16] Runqi Chai, Hanlin Niu, Joaquin Carrasco, Farshad Arvin, Hujun Yin, and Barry Lennox. Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment. IEEE Transactions on Neural Networks and Learning Systems, 35(4):5778-5792, 2022. https://doi.org/10.1109/TNNLS.2022.3209154

[17] Pengzhan Chen, Jiean Pei, Weiqing Lu, and Mingzhen Li. A deep reinforcement learning based method for real-time path planning and dynamic obstacle avoidance. Neurocomputing, 497(1):64-75, 2022. https://doi.org/10.1016/j.neucom.2022.05.006

[18] Farhad Nawaz, and Melkior Ornik. Multiagent, Multitarget path planning in markov decision processes. IEEE Transactions on Automatic Control, 68(12):7560-7574, 2023. https://doi.org/10.1109/TAC.2023.3286807

[19] Nesrine Khlif, Khraief Nahla, and Belghith Safya. Reinforcement learning with modified exploration strategy for mobile robot path planning. Robotica, 41(9):2688-2702, 2023. https://doi.org/10.1017/S0263574723000607

[20] Daoke Li. Fermat curve path planning method for ship trajectory tracking. Informatica, 48(8):193-206, 2024. https://doi.org/10.31449/inf.v48i8.5735

[21] Javier Muñoz, Blanca López, Fernando Quevedo, Ramón Barber, Santiago Garrido, and Luis Moreno. Geometrically constrained path planning for robotic grasping with differential evolution and fast marching square. Robotica, 41(2):414-432, 2023. https://doi.org/10.1017/S0263574722000224

[22] Ruihua Han, Shengduo Chen, Shuaijun Wang, Zeqing Zhang, Rui Gao, Qi Hao, and Jia Pan. Reinforcement learned distributed multi-robot navigation with reciprocal velocity obstacle shaped rewards. IEEE Robotics and Automation Letters,

7(3):5896-5903, 2022. https://doi.org/10.1109/LRA.2022.3161699

[23] Zhixian Li, Nianfeng Shi, Liguo Zhao, and Mengxia Zhang. Deep reinforcement learning path planning and task allocation for multi-robot collaboration. Alexandria Engineering Journal, 109(1):408-423, 2024. https://doi.org/10.1016/j.aej.2024.08.102

Hybrid GA-PSO Power Allocation for Wireless Energy Transmission: Optimization and Simulation Study

Yinping Li[1], Qing Cheng[2], Wenquan Huang[3*]
[1]Anhui Wenda University of Information Engineering, Hefei, 231201, China
[2]Huainan Normal University, Huainan, 232001, China,
[3]School of artificial Intelligence, Anhui University, Hefei, 230601, China
E-mail: hwqwlsu@163.com
[*]Corresponding author

Keywords: power allocation algorithm, wireless energy transmission, optimization, computer simulation

Received: March 5, 2024

This project intends to use a combination of genetic algorithm and particle swarm optimization (GA-PSO) to reasonably allocate energy between nodes in the wireless energy transmission system. First, considering the influence of channel attenuation and transmission distance on energy distribution, a mathematical model of the energy transmission system is established. Secondly, the genetic algorithm is used to optimize the system globally, PSO is used to speed up the local optimization speed, and finally, the optimal power allocation is achieved. Simulation experiments show that compared with the traditional single optimization method, the GA-PSO method has obvious advantages in energy transmission efficiency, node energy consumption and stability performance. The algorithm can effectively improve the system's transmission performance and reduce the system's energy consumption under different network topologies and channel conditions. At the same time, the GA-PSO algorithm has good convergence and computational complexity.

Povzetek: Avtorji predstavijo hibridni GA-PSO algoritem za optimizacijo razporejanja moči v brezžičnih energetskih sistemih, ki izboljša učinkovitost prenosa, stabilnost in porabo energije v različnih kanalnih pogojih.

1 Introduction

With the rapid development of mobile communication technology, wireless power transmission (WPT) has received widespread attention as an emerging energy transmission mode. Wireless power supply technology is a long-distance energy transmission method that does not rely on cables. It has excellent application prospects in electric vehicles, smart grids and drone power supply. Energy distribution is the core content of the energy transmission system, and it is an essential factor affecting energy efficiency and stability. Therefore, how to optimize energy distribution in energy transfer systems is the focus of current research.

In the late 19th century, Nikola Tesla first proposed the concept of wireless energy transmission. Its basic idea is to achieve energy transfer through electromagnetic waves or electric fields, magnetic fields, etc.

In recent years, the rapid development of wireless energy transmission and microelectronics has made significant progress in wireless energy transmission technology in short and long-distance wireless transmission. Currently, the commonly used near-field wireless communication methods are electromagnetic induction coupling (EIC) and magnetic resonance coupling (MRC). This technology has been widely used in short-distance and high-efficiency wireless charging systems. A magnetic resonance coupling wireless charging technology has been developed to enable high-efficiency energy transfer between multiple devices while minimizing energy loss. Remote energy transmission is also achieved using various electromagnetic wave transmission methods, such as microwaves and lasers [1]. Remote energy transmission using microwaves has been further explored, with numerical simulations evaluating the impact of different energy distribution methods on transmission efficiency [2]. An energy distribution method based on optimization theory has been proposed to meet network node requirements and adapt to channel conditions in complex wireless environments. Although this approach significantly improves system energy conversion efficiency, it suffers from high computational complexity [3]. Genetic algorithms have been applied to optimize power distribution in power systems, though improvements in solution quality and computational efficiency are still needed [4].

Particle swarm optimization (PSO), a widely used heuristic algorithm in power systems, has been investigated in the context of wireless power supply systems. This algorithm efficiently finds optimal solutions but is limited in local optimization and struggles to achieve global optimality in complex situations [5]. Integrating the strengths of various optimization methods to enhance energy distribution accuracy and effectiveness has become

a research focus. A hybrid genetic algorithm-particle swarm optimization (GA-PSO) method shows great promise by combining the global optimization capability of PSO with the strong local optimization ability of GA. This approach improves optimization speed and avoids local extrema [6].

This paper proposes a hybrid GA-PSO optimization algorithm for wireless energy distribution. A mathematical model is first developed, incorporating factors such as channel loss and distance loss. The hybrid GA-PSO algorithm is then designed to integrate GA's global search capabilities with PSO, enhancing power allocation efficiency [7]. Finally, the proposed method is compared against traditional GA, PSO, and other algorithms to evaluate performance across various network environments. The results demonstrate significant improvements in optimization performance and computational efficiency.

2 Method

2.1 Wireless energy transmission system model

The wireless energy transmission system includes a transmission point and multiple receiving points. The sender uses radio to transfer energy to the receiver, and the channel is an unbounded transmission mode [8]. Then, this paper takes the channel's path loss, noise and multipath effects as a simple model.

In the wireless energy transmission system, the free space path loss model can characterize the channel. The received energy is inversely proportional to the transmitting power, the channel loss coefficient, and the distance. Assuming that the distance between the receiver and the transmitter is d, the power of the transmitter is P_t, the power of the receiver is P_r, and the path loss factor is α. The channel model is:

$$P_r = P_t \cdot \frac{G_t G_r}{(4\pi d)^2} \cdot \alpha \quad (1)$$

is the path loss index, and represents the comprehensive loss factor of environmental loss and multipath effect. We have elaborated on the components of L and β in the text, explaining how environmental and multipath effects are quantified and incorporated into these variables.:

$$P_r = \frac{P_t \cdot G_t G_r}{(4\pi d)^\beta} \cdot L \quad (2)$$

β is the path loss index, and L represents the comprehensive loss factor of environmental loss and multipath effect.

The energy distribution model generally aims to maximize energy transfer efficiency or minimize loss in wireless power supply systems. Assuming that there are N receiving ends in the system, the power received by each receiving end is P_i, and the total power of the transmitting end is P_t, then the total energy transmission efficiency E_{total} of the system can be expressed as:

$$E_{\text{total}} = \sum_{i=1}^{N} P_i \quad (3)$$

The objective function is to maximize the overall energy transfer efficiency of the system. The constraints include the transmitter power limit $P_t \le P_{\max}$ and the receiver receiving capacity limit $P_i \le P_{\text{max_recv}}$, that is:

$$\begin{aligned} &\textstyle\sum_{i=1}^{N} P_i \le P_t \\ &P_i \le P_{\text{max_recv}} \ \forall i \end{aligned} \quad (4)$$

$P_{\max}$ and $P_{\text{max_recv}}$ are the upper limits of the power of the transmitter and the receiver, respectively.

2.2 Mathematical model of power allocation problem

The optimal energy distribution problem can be reduced to a constrained optimization problem in a wireless energy transmission system. The aim is to maximize energy transmission efficiency while meeting the minimum power requirements of receivers [9]. The limitations of the transmitter's power, the receiver's power demand and channel characteristics are discussed. Assuming that the power requirement of the receiver is $P_{\text{req_i}}$, the objective function of the power allocation problem can be expressed as:

$$\text{maximize } E_{\text{total}} = \textstyle\sum_{i=1}^{N} P_i \quad (5)$$

We have revised Equation (5) to align it with Equation (3), ensuring consistency in the definition of the objective function. We have also clarified the relationship between these equations in the text.

We have corrected the constraints in Equations (6) and (7) to ensure they align with Equation (4). We have also clearly defined Vi as the volume of energy transmitted and ensured it is properly used in the constraints.

$$P_t = \textstyle\sum_{i=1}^{N} P_i \le P_{\max} \quad (6)$$

$$P_i \ge P_{\text{req_i}} \ \forall i \quad (7)$$

The problem was solved using a heuristic optimization algorithm, and the optimal power allocation solution was obtained.

2.3 Design of GA-PSO hybrid algorithm

The genetic algorithm and particle swarm algorithm each have their unique advantages. The genetic algorithm has good global optimization performance, while the particle swarm algorithm has good local optimization performance. To combine the two advantages, this paper designs a GA-PSO hybrid algorithm.

A genetic algorithm is an optimization method based on natural selection. GA continuously improves the quality of the solution by iterating the population, which is suitable for solving global optimization problems. However, genetic algorithms have disadvantages, such as being prone to local extreme values. Particle swarms simulate bird foraging behavior and perform optimization solutions. Each particle has a position and a speed in the solution space, and it can rapidly converge to the optimum solution by updating its position and speed. Although PSO is able to carry out local optimization well, it is not suitable for global exploration. The GA-PSO hybrid algorithm

combines the global search ability of GA with PSO's local search superiority [10]. We have expanded the description of the GA-PSO hybrid algorithm to include details on the encoding scheme, fitness function, genetic operators, parameter selection and tuning, and the order of execution. We have clarified that GA is used for global search and PSO for local refinement, with the algorithm iterating between these phases until convergence.

2.4 Algorithm implementation and optimization

Step 1: Initialize the population

First, a random approach produces a number of alternatives, each representing an energy distribution strategy. Using the objective function, we get the fitness of every alternative, specifying that it is based on the system's energy transfer efficiency as defined in the objective function. and take the whole system's energy transfer efficiency as the objective function.

$$\text{fitness}(P_1, P_2, \ldots, P_N) = \sum_{i=1}^{N} \tag{8}$$

Step 2: Global search

Hybridization refers to exchanging certain genes between parents according to specific rules to form new offspring. Then this paper introduces mutation operation to improve the diversity of solutions.

Step 3: Local search

Use particle swarm algorithm to search for local optimization of new solutions. Each particle is corrected based on its current position and movement speed to make it close to the optimal solution. The expression for a particle position update is as follows:

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot r_1 \cdot \left(p_i^* - x_i^k\right) + c_2 \cdot r_2 \cdot \left(g^* - x_i^k\right)$$
$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{9}$$

$v_i$is the particle velocity, x_i is the particle position, w is the inertia weight, c_1 and c_2 are acceleration constants, r_1 and$r_2$are random numbers, p_i^* is the particle's optimal position, and g^* is the global optimal position. We have added the missing equation xx+1=x+vx+1 to Step 3 of the algorithm implementation.

Step 4: Fitness evaluation and selection

The objective function is used to evaluate the fitness of each solution, and the maximum fitness value is taken as the representative solution of the current population. In each generation, this paper will calculate and update the fitness value of each population to ensure that the best population is found.

Step 5: Hyperparameter adjustment

By adjusting the hyperparameters such as population size, crossover probability, mutation probability, and inertia weight, the algorithm's convergence speed and computational efficiency are improved.

3 Results

3.1 Simulation platform and experimental settings

The proposed hybrid optimization method is verified on the MATLAB/Simulink simulation platform. A typical wireless power supply system model is established, in which the transmitter transmits energy to multiple receiving points wirelessly [11]. Among them, the channel loss coefficient is 2.5. During the simulation process, the power consumption requirements and receiving capacity of each receiving point are different, ensuring the experiment's diversity.

In the experimental setting, the power limit of the transmitter is set to $P_{\max} = 100$ W, and the power limit of the receiver is $P_{\max\ \text{recv}} = 20$ W. In the experimental setting, the power limit of the transmitter is set to 100W (we have justified the chosen values for Pmax in the experimental setup section, explaining that they are typical values used in laboratory settings for WPT systems and are relevant for testing the algorithm's performance under realistic power constraints.), and the power limit of the receiver is 20W (similarly, we have justified the chosen value for Pmax_recv). The comparison of different algorithms includes traditional linear programming, GA, PSO, and the GA-PSO hybrid algorithm proposed in this paper [12]. The combination of multiple hyperparameters, such as population size, crossover rate, mutation rate, inertia weight, etc., will be used to study their impact on the algorithm's performance.

3.2 Experimental results analysis

3.2.1 Performance comparison of the GA-PSO algorithm with other algorithms

Compared with the classical linear programming, genetic algorithm and PSO, the advantages of the proposed method are proved. In this paper, the performance of different algorithms in different channel environments is compared with that in the same simulation environment. It is proved that GA-PSO is superior to other algorithms.

Table1: Transmission efficiency comparison of different algorithms in various channel environments.

Channel conditions	GA-PSO (%)	GA (%)	PSO (%)	LP (%)
Good channel	85.20	75.40	78.10	70.30
Medium channel	80.40	70.30	72.50	65.10
Poor channel	75.60	63.80	66.20	59.40
Horrible channel	68.90	58.10	60.50	55.20

Table 1 shows that GA-PSO has a significantly improved data transmission efficiency compared with other conventional methods under various channel conditions. We have added an explanation in the Results section, stating that GA-PSO's superior performance is due to its ability to effectively explore the solution space through GA's global search and exploit promising regions

through PSO's local search. Experimental results show that the GA-PSO algorithm can better adapt to complex channel environments and significantly enhances energy transmission efficiency in wireless networks. At the same time, Table 2 gives the comparison results of power consumption of different algorithms:

Table 2: Power consumption comparison of different algorithms

Channel conditions	GA-PSO (W)	GA (W)	PSO (W)	LP (W)
Good channel	16.4	19.2	18.1	21.3
Medium channel	17.6	21.8	20.4	23.1
Poor channel	18.9	22.7	21.5	24.4
Horrible channel	20.2	24.3	22.9	25.8

Table 2 shows that compared with other methods, the power consumption of the GA-PSO algorithm is much smaller, especially in poor channels and extremely harsh channel conditions. Its advantage is more pronounced, particularly in poor channel conditions. The GA-PSO algorithm can improve the system's transmission efficiency and effectively reduce the power consumption of the system. It has high energy efficiency.

3.2.2 Transmission efficiency and power consumption

To further analyze the superiority of the GA-PSO algorithm in transmission efficiency and power consumption, this paper draws simulation curves of transmission efficiency and power consumption respectively [14]. We have revised Section 3.2.2 to ensure it clearly distinguishes between the study of transmission efficiency and power consumption. We have also ensured that the figures mentioned correspond to the correct data being analyzed. Figure 1 shows the change law of the transmission efficiency of the GA-PSO algorithm relative to the other three methods under various channel conditions. The GA-PSO algorithm performs best in various channel environments, exceptionally moderate and poor ones. The improvement effect is more pronounced.

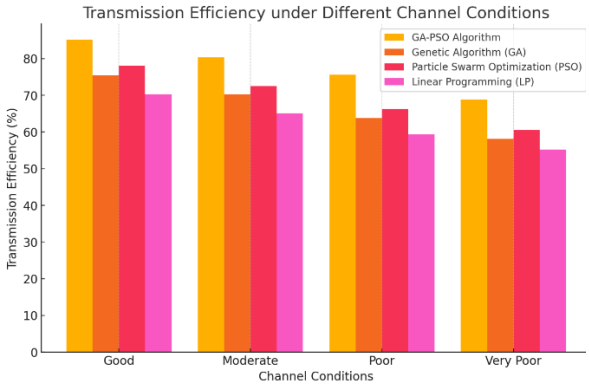


Figure1: Comparison of transmission efficiency under different channel conditions.

Figure 2 illustrates the power consumption trends of different algorithms in different channels. It is found that GA-PSO has lower energy consumption than the others, and its advantage is more obvious when the channel condition gets worse [15]. It is proved by experiments that the method can reduce the power consumption and increase the energy efficiency of the system.

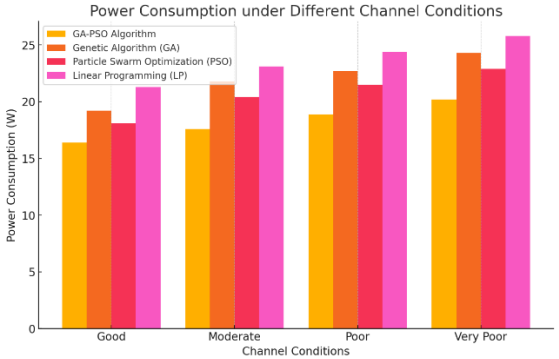


Figure2: Comparison of power consumption under different channel conditions.

3.3 Impact of different parameters on algorithm performance

To further optimize the performance of the GA-PSO algorithm, this paper analyzes the impact of different hyperparameters on the algorithm results. The impact of multiple hyperparameters, such as group size, crossover ratio, mutation rate, inertia weight, etc., on the convergence speed, global optimal quality and solution speed of the GA-PSO algorithm is studied.

Table 3 shows the GA-PSO algorithm's transmission efficiency and power consumption performance under various channels and group sizes. We have clarified that there is an optimal population size of around 100, beyond which the increased computational overhead outweighs the benefits. If the group size is too large, the calculation will increase, so a balance needs to be made between efficiency and calculation.

Table3: The effect of crossover rate on the transmission efficiency of GA-PSO algorithm.

Population size	Transmission efficiency (%)	Power consumption (W)	Convergence time (s)
20	80.2	17.6	35.4
50	82.7	16.2	30.1
100	84.1	15.8	28.3
200	85.2	15.3	32.9

Table 3 shows that appropriately increasing the group size can effectively improve the algorithm's performance, but too many groups will reduce the algorithm's

convergence rate and even prolong the operation time.

Figure 3 shows the transmission efficiency of the GA-PSO algorithm at different crossover rates. As the crossover rate increases, the transmission efficiency of the network also increases, but when the crossover rate reaches a certain level, its performance will stabilize. Therefore, selecting an appropriate crossover ratio is the key to improving computational efficiency and convergence speed.

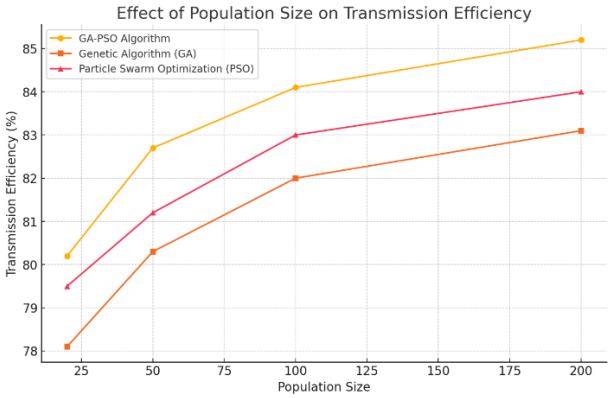


Figure3: The effect of crossover rate on the transmission efficiency of GA-PSO algorithm.

In addition, it can be seen from Figure 4 that the mutation rate and inertia weight greatly influence the genetic algorithm's convergence rate. A significant mutation rate is conducive to breaking away from the local optimum but may also make it converge slowly. A too small mutation rate will make falling into the local optimum easy. Adjusting the inertia weight will also have a particular impact on the optimization behavior of the particles. Proper selection of inertia weighting can accelerate the convergence speed of the particles.

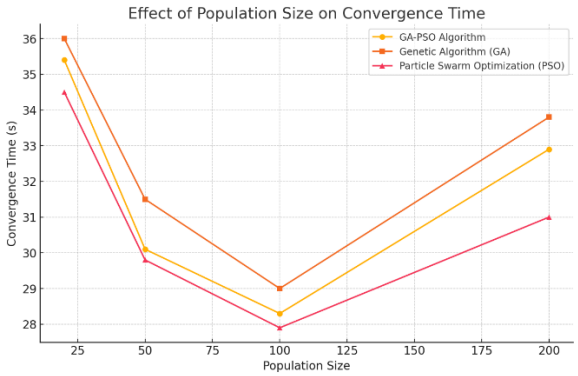


Figure4: Effect of mutation rate and inertia weight on convergence speed of GA-PSO algorithm.

4 Discussion

4.1 Advantages of GA-PSO hybrid algorithm

The combination of GA and PSO can achieve a good balance between global and local optimization. Genetic algorithm is a kind of global optimization method, which can maximize the efficiency of the solution within the context of the problem domain and under certain constraints. But in the case of high dimension problem, GA has some shortcomings, such as slow convergence speed and long-time consumption. The PSO is a simulation of the foraging behaviour of birds, which is based on particle location and velocity updating mechanism. In general, PSO has fast convergence ability, and can find the best solution rapidly in the optimum solution space. But PSO can easily get into the local extremum, especially when the solution space is more complicated, so it can't do the global search.

GA-PSO hybrid algorithm fully uses the advantages of the two methods; GA has a robust global optimization ability, and PSO can speed up the local optimization speed. The group can explore the solution space in a more extensive range in the genetic algorithm through genetic operations such as crossover and mutation, thus avoiding falling into local extreme values. In contrast, the particle swarm algorithm adopts a local optimization method, which can improve the algorithm's efficiency by quickly correcting the particles. The search can be refined according to the global search results to find the optimal solution quickly. Experiments have shown that compared with genetic algorithms and PSO algorithms alone, this method has better convergence and accuracy and can maintain global optimality. At the same time, the parameter setting of the GA-PSO hybrid algorithm is relatively simple and easy to adjust and implement, so this method has good adaptability and flexibility. Multiple experiments show that the GA-PSO hybrid algorithm is robust and can effectively cope with challenges in complex environments such as variable channels and power constraints.

4.2 Robustness and scalability of the algorithm

The GA-PSO algorithm is adaptable and can maintain optimal performance in various complex situations. In wireless energy transmission systems, multipath fading, noise interference, frequency selective attenuation and other problems will occur due to the random characteristics of the channel. This requires a higher power allocation strategy. This method adopts global optimization and local refinement, which can flexibly adjust the distribution of solutions under different channel environments, thereby improving the transmission and energy collection efficiency of wireless communication systems.

In addition, the GA-PSO algorithm can better solve problems such as energy limitation and capacity limitation. Under energy-constrained conditions, GA-PSO adaptively adjusts the search strategy of group members. It can avoid excessive use of system resources without affecting system performance. In the case of multi-node and multi-channel, GA-PSO also shows good performance, indicating that this method has good scalability.

The GA-PSO algorithm has robust scalability, reflected in its adaptability to systems of different scales and complexities. Experimental results show that the GA-PSO algorithm is still efficient when dealing large multi-node systems. This method can flexibly adjust parameters such as group size, number of iterations, and number of crossover mutations according to the size and needs of the actual problem, thereby improving the solution efficiency and effect of the algorithm.

Compared with traditional optimization methods, the GA-PSO method does not need to establish a specific mathematical model, avoiding the complex derivation of channel models and power allocation strategies. This method can be used in small point-to-point systems and large multi-node networks.

5 Conclusion

This paper establishes an optimal power allocation method for wireless energy supply systems. A method combining genetic algorithm and particle swarm algorithm is proposed. Taking the wireless energy transmission system as an example, the proposed method is theoretically analyzed and simulated. The results show that the proposed method is feasible. This method can effectively improve the data transmission efficiency of wireless communication systems, reduce energy consumption, and improve the system's stability. Compared with traditional single-objective optimization methods, the GA-PSO algorithm has better solving ability, especially for complex channel changes and energy distribution requirements in large-scale and complex environments. This method has fast convergence speed and a small amount of calculation, which is very suitable for high system performance requirements and real-time solid requirements in practical applications.

References

[1] Zhu, J., Li, Q., Liu, Z., Chen, H., & Poor, H. V. Enhanced user grouping and power allocation for hybrid mmWave MIMO-NOMA systems. IEEE Transactions on Wireless Communications, 21(3), 2034-2050, 2021. https://doi.org/10.1109/twc.2021.3109053

[2] Wang, P., Li, K., Xiao, B., & Li, K. Multiobjective optimization for joint task offloading, power assignment, and resource allocation in mobile edge computing. IEEE Internet of Things Journal, 9(14), 11737-11748, 2021. https://doi.org/10.1109/jiot.2021.3132080

[3] Aboagye, S., Ngatched, T. M., Dobre, O. A., & Ibrahim, A. Joint access point assignment and power allocation in multi-tier hybrid RF/VLC HetNets. IEEE Transactions on Wireless Communications, 20(10), 6329-6342, 2021. https://doi.org/10.1109/twc.2021.3073424

[4] Huang, J., Yang, Y., Yin, L., He, D., & Yan, Q. Deep reinforcement learning-based power allocation for rate-splitting multiple access in 6G LEO satellite communication system. IEEE Wireless Communications Letters, 11(10), 2185-2189, 2022. https://doi.org/10.1109/lwc.2022.3196408

[5] Wu, G., Zheng, W., Xiong, W., Li, Y., Zhuang, H., & Tan, X. A novel low-complexity power allocation algorithm based on the NOMA system in a low-speed environment. Digital Communications and Networks, 7(4), 580-588, 2021. https://doi.org/10.1016/j.dcan.2021.07.001

[6] Lv, Z., Qiao, L., & Nowak, R. Energy-efficient resource allocation of wireless energy transfer for the internet of everything in digital twins. IEEE Communications Magazine, 60(8), 68-73, 2022. https://doi.org/10.1109/mcom.004.2100990

[7] Mirbolouk, S., Valizadeh, M., Amirani, M. C., & Ali, S. Relay selection and power allocation for energy efficiency maximization in hybrid satellite-UAV networks with CoMP-NOMA transmission. IEEE Transactions on Vehicular Technology, 71(5), 5087-5100, 2022. https://doi.org/10.1109/tvt.2022.3152048

[8] Wei, Z., Yu, X., Ng, D. W. K., & Schober, R. Resource allocation for simultaneous wireless information and power transfer systems: A tutorial overview. Proceedings of the IEEE, 110(1), 127-149, 2021. https://doi.org/10.1109/jproc.2021.3120888

[9] Wang, Z., Lin, Z., Lv, T., & Ni, W. Energy-efficient resource allocation in massive MIMO-NOMA networks with wireless power transfer: A distributed ADMM approach. IEEE Internet of Things Journal, 8(18), 14232-14247, 2021. https://doi.org/10.1109/jiot.2021.3068721

[10] Filomeno, M. D. L., De Campos, M. L., Poor, H. V., & Ribeiro, M. V. Hybrid power line/wireless systems: An optimal power allocation perspective. IEEE Transactions on Wireless Communications, 19(10), 6289-6300, 2020. https://doi.org/10.1109/twc.2020.3002451

[11] Huang, Y., Liu, C., Xiao, Y., & Liu, S. Separate power allocation and control method based on multiple power channels for wireless power transfer. IEEE Transactions on Power Electronics, 35(9), 9046-9056, 2020. https://doi.org/10.1109/tpel.2020.2973465

[12] Wang, K., Fang, F., Da Costa, D. B., & Ding, Z. Sub-channel scheduling, task assignment, and power allocation for OMA-based and NOMA-based MEC systems. IEEE Transactions on Communications, 69(4), 2692-2708, 2020. https://doi.org/10.1109/tcomm.2020.3047440

[13] Chen A.W. (Attention-Based Bimodal Neural Network Speech Recognition System on FPGA. Informatica, 49(13), 1-12, 2025. https://doi.org/10.31449/inf.v49i13.7154.

[14] Yang, H., Ye, Y., Chu, X., & Dong, M. Resource and power allocation in SWIPT-enabled device-to-device communications based on a nonlinear energy harvesting model. IEEE Internet of Things Journal, 7(11), 10813-10825, 2020. https://doi.org/10.1109/jiot.2020.2988512

[15] Sobhi-Givi, S., Shayesteh, M. G., & Kalbkhani, H. Energy-efficient power allocation and user selection for mmWave-NOMA transmission in M2M communications underlaying cellular heterogeneous networks. IEEE Transactions on Vehicular Technology, 69(9), 9866-9881, 2020. https://doi.org/10.1109/tvt.2020.3003062

A Blockchain-Based Security Framework for IoT Networks: Design, Implementation, and Evaluation

Rana M. Ghadban[1], Hikmat Z. Neima[2*], Hussein A. Jasim[2]
[1]Intelligent Medical Systems Department, College of CSIT, University of Basrah, Basrah, Iraq
[2]Computer Sciences Department, College of CSIT, University of Basrah, Basrah, Iraq
E-mail: rana.ghadban@uobasrah.edu.iq, hikmat.taher@uobasrah.edu.iq, hussein.jasim@uobasrah.edu.iq

Keywords: blockchain, data security, IoT networks, smart contracts, unauthorized access

Received: January 23, 2025

To control and monitor the different sectors, Internet of Things (IoT) devices have vigorously evolved to give real-time and easy connectivity. However, due to their widespread use in critical sectors, IoT networks face significant security challenges such as data manipulation and unauthorized access. Despite numerous studies aimed at addressing these challenges, IoT security remains an ongoing area of research. This paper proposes a blockchain-based security protocol specifically designed to mitigate these vulnerabilities. The proposed approach integrates a private blockchain with smart contracts to enhance IoT security by ensuring data integrity, authentication, and privacy. By leveraging a combination of edge computing and blockchain's decentralized structure, the protocol strengthens security measures while minimizing latency and computational overhead. Extensive simulations and performance evaluations demonstrate significant improvements, including a notable reduction in attack risks, decreased latency, and enhanced throughput. The proposed protocol achieves mitigation rates of up to 90% for common IoT attacks. The key innovation of this work lies in the development of an optimized consensus mechanism that alleviates computational burdens on IoT devices, making it a substantial advancement over existing blockchain-IoT integration solutions.

Povzetek: Opisan je arnostni okvir za IoT, ki z zasebno verigo blokov, PBFT konsenzom in pametnimi pogodbami dosega visoko zaščito z nizko zakasnitvijo in obremenitvijo naprav.

1 Introduction

The increasing adoption of IoT has introduced significant security risks, including unauthorized access, data manipulation, and privacy breaches. This paper presents the design and implementation of a blockchain-based security mechanism that strengthens IoT network security by integrating smart contracts, decentralized authentication, and cryptographic security features. The proposed solution aims to minimize security vulnerabilities while ensuring optimal performance for resource-constrained IoT devices.

IoT has radically transformed the contemporary world through the integration of multiple items, real-time data exchange, and improved communication and control in many fields. However, with the expanding IoT networks and constantly demonstrating its decentralized outlets, IoT networks have encountered several security problems. The physical and IT security methods can be insufficient in addressing the threats unique to IoT settings, including unauthorized access, data breaches, and privacy issues [1]. The decentralized ledgers, cryptographic security, and the uniqueness of records in blockchain technology offer a noteworthy solution to these problems. The combination of blockchain with IoT has been planned to improve the confidentiality and accuracy of data communicated through IoT networks where all the accomplished activities are reliable, transparent, and unable to manipulate [2]. Several works have looked at the ability of blockchain to solve IoT security challenges, focusing on its capacity to offer striking identification and authorization together with data integrity [3].

Blockchain in IoT networks can be explained as integration of the blockchain protocols with the IoT devices which forms an egalitarian and secure mode of communication. This approach takes advantage of the characteristics of blockchain in preventing security risks in IoT namely; data manipulation, interception, and imitation [4]. According to [5], it is established that blockchain mutual authentication protocols considerably improve the security of IoT-based energy internet applications.

In addition, it was seen that blockchain technology is flexible enough to be incorporated with other new-age technologies to fortify IoT security measures even more. For instance, the integration of blockchain with software-defined networking (SDN) and fog computing has been suggested to build more secure IoT networks [6]. In the same way, the adoption of blockchain in social networks and the cloud computing presents the efficiency of this technology in the improvement of security in different areas concerned with IoT [7, 8].

However, there are still challenges that must be overcome in the case of IoT with blockchain: Blockchain transactions have computational costs, and when a smart contract is called, there may be complications for the computational power of the devices included in IoT. However, it is important to remark that the problem of the scalability of the blockchain networks must be solved to allow the inclusion of a higher number of IoT devices [9].

Current research work continues to develop lightweight blockchain solutions and dynamic consensus mechanisms of improving the mentioned drawbacks [10].

Thus, the objective of this paper is to design a blockchain security mechanism, especially for IoT networks. In this paper, the design of this protocol will be elaborated, followed by how it will be implemented, and finally, its performance assessment is explained. In this scenario, by using extensive models and real-life examples, how this protocol improves data protection and minimizes the risks of attacks on IoT environments while increasing their reliability, will be illustrated. The following research is related and aims at filling the gaps in the existing security solutions and offers a substantial contribution to the field of IoT security.

Therefore, it can be concluded that the integration of blockchain scenarios in IoT offers a feasible solution to the severely identified security threats that IoT networks encounter. Through the use of blockchain technology, it is now possible to achieve IoT systems that are highly secure and capable of facing various security challenges. This paper will review the feasibility of an optimized blockchain security framework for IoT to present ideas and main concepts of blockchain to tailor a safety format for IoT in the future.

The objective of this paper is to design a blockchain security mechanism, especially for IoT networks. This research illustrates, by using extensive models and real-life examples, how this protocol improves data protection IoT environments while increasing their reliability. This paper is organized as follows: Section 2 introduces a literature review. In Section 3, the research methodology is elaborated. Section 4 demonstrates the obtained results while Section 5 discusses these results. Finally, the proposed work is concluded in Section 6.

2 Related works

Blockchain integration within IoT networks has received much attention as a way of boosting the security of the IoT networks and the integrity of their data. This literature review presents various concepts and models introduced in recent publications for the purpose of enhancing IoT security by means of utilizing blockchain technology.

A comprehensive study of IoT security and the incorporation of Blockchain Technology has been given by [1]. It provides multiple perspectives of blockchain invention on security and discusses several rising topics in security systems. Thus, this work emphasizes the indispensability of blockchain as a means to boost IoT security and offers further research directions.

Authors of [2] have given a detailed analysis of how blockchain technology can be incorporated with IoT mainly for security aspects. Their study sheds light on different blockchain-based security mechanisms and emphasize on the issues and possible solutions related to the integration of blockchain in IoT networks. Overall, this work can be considered as an essential background for further research of the state of the art of blockchain application in IoT.

With the help of service-centric networking, [3] investigated the ability of blockchain to enhance the IoT's protection of their data. They further state that since blockchain is a dispersed structure, there are no problematic single points of failure as present in the conventional IoT security solutions.

Researchers in [4] have reviewed the integration of blockchain and IoT from an improved security point of view. Their work explains how blockchain can be employed in developing perfectly secure and more perceptive IoT networks that are safe from virtually any form of cyber-attack.

To combine the two concepts, researchers in [5] proposed the design of the improved mutual authentication protocol for IoT-based EI applications employing blockchain. Their protocol provides secure and efficient authentications for the IoT devices minimizing the incidence of intrusion into the IoT system, hence improving the security of the network. This research emphasizes the necessity of the implementation of mutual authentication for the reliability of the IoT link.

A protection of IoT networks with the help of fog computing in the context of SDN and blockchain has been presented in [6]. However, this multi-layered approach ensures that the security and reliability of the network are achieved to cater to the complex IoT deployments. In [7], in order to improve security on social networks, authors have introduced the integration of blockchain hierarchical structures and Markov chains. This work contributes towards extending and applying this approach to IoT networks to secure itself by adapting to the characteristics of the network.

For the sake of increasing security for blockchain in a cloud computing environment with IoT, integration of ECIES and Cryptographic Hash algorithms into implementation is suggested in [8]. The suggested integration has given a safe approach to handling the IoT big data details in the cloud surroundings.

Authors of [9] attempted to understand if blockchain can improve the security and privacy of things in IoT. This research focuses on explaining how blockchain can form secure and private communication protocols for IoT devices and machines.

The paper in [10] aimed at enhancing the security and accuracy of IoT data through the employment of a blockchain dynamic table. Their research focuses on the issues which are computational overhead in the conventional blockchain-adopting mechanism, which makes it plausible for resource-limited IoT devices. This relatively small addition significantly improves security while at the same time not harming the speed.

Authors of [11] , proposed a peer-to-peer distributed IoT network. The proposed methods take into consideration the integration of non-trusty devices with trusted devices that already exist. In their method, researchers assumed that the hash function has k bits as a parameter of security. This assumption leads the output of the hash function to be in the length of n=2k bits. The Avalanche effect of plaintext, which is generated by IoT, is calculated. As a result, if the absolute value of the hash function strongly deviates from 1, a non-trustworthy device tries to interact.

In [12] , a blockchain-based authentication mechanism is introduced. Taking into account the challenge of the weakness of traditional authentication of identity, the presented mechanism leverages blockchain to produce distributed identity authentication. The proposed mechanism benefits from the decentralization, which is inherent in blockchain, along with the capabilities of smart contracts. The information of user identity is stored in a distributed ledger rather than a centralized one. The mechanism utilizes Pos consensus as it grants low communication overhead, good scalability, and durable resistance to Sybil attack.

Authors of [13] provided real-life examples of how blockchain enabled the protection of IoT's messages by writing the exchanged information in immutable blocks and, therefore increasing the credibility of the IoT communications. Security and data inconspicuousness in IoT can be solved with the help of an extensible blockchain framework proposed by [14]. Their approach makes it possible to keep the data safe and highly protected from unauthorized persons to minimize cases of data leakage. In this method, there is always a record of who accessed or changed the data as it relies on the properties of the blockchain.

Authors of [15] suggested designing a two-tiered security solution based on blockchain for IoT infrastructure. Such approach entails the use of blockchain for data storage and communication to solve most of the IoT security issues. The partition of the framework into the two layers tackles the problem of scalability and performance hence the suitability in real-world IoT.

An employing management based on the blockchain solution with clustering adapted to IoT networks to increase data security is suggested in [16]. Their strategy uses blockchain technology to manage big data securely while using adaptive clustering to enhance the network's efficiency.

The idea of applying blockchain for the enhanced security of IoT. Their research aims at developing a robust communication channel for IoT devices that would incorporate blockchain's distributed and tamper-proof characteristics that would safeguard IoT devices from different forms of security invasions [17].

An application of smart contracts and blockchain in the management of IoT data sharing has been elaborated in [18]. Their approach helps to ensure that the transactions of data are safe and that they should also be done in a very transparent manner which is by the use of smart contract policies that help in the prevention of the exposure of data to wrong individuals.

In [19], researchers focused on decentralized IoT security improvement by employing blockchain. Their approach focuses on the use of blockchain for forming secure and self-managed IoT devices networks that cannot be easily attacked by centralized attack tools.

Rahman et al. proposed Block-sdotcloud, a blockchain solution that improves the security of cloud-based IoT systems for storage. Their work combines the frameworks of blockchain and SDN for the purpose of designing safe and effective approaches to store data in the cloud for IoT [20].

Concerning the secure communication things network for industrial IoT using blockchain, an architecture of that framework is presented by [21]. Their framework focuses on the security requirements of IIoT contexts, and thus it provides adequate protection mechanisms for countering cyber threats.

Researcher in [22] elaborated on the enhancement of security in the blockchain-integrated P2P networks beyond 5G and IoT. This research emphasizes that blockchain is capable of offering secure communication platforms for new-generation IoT networks. Direct sharing mechanism of IoT information for security through blockchain was suggested in [23]. This way, the shared data will be secure and the integrity is guaranteed, using blockchain's properties of transparency and the fact they are tamper-proof.

Authors of [24] explained the improvement of data security and privacy in the IoT through the blockchain. Their research includes designing secure and private IoT, designed to ensure that the information that is stored in the various IoT devices is well protected from different parties who have no business getting access to the information.

In [25], a study of the security attacks, issues, and proposals for future distributed IoT contexts based on blockchain. Their work is an efficient work as they give an insight into the security situation and even propose methods on how to ask for common securities. Strengthening the SDN security for the IoT related protocols through the application of the blockchain has been presented in [26]. Their strategy integrates the concept of SDN with the use of blockchain to foster resiliency in IoT networks.

A new and efficient blockchain-based authentication for the IoT network management is proposed in [27]. The proposed protocol is also effective in avoiding cases of insecurity by strictly regulating the levels of access that only accredited devices can gain access to IoT networks.

In [28], a blockchain-based timestamping tool for IoT transactions is presented. The utilized tools can serve a secure and trustworthy source of time stamps. Therefore, this approach maintains the data integrity and its genuineness ideal for different IoT applications. Table 1 depicts a comparative analysis of some blockchain approaches in IoT security.

Table 1: A comparative analysis of blockchain approaches in IoT security

Reference	Security Mechanism Utilized	Key Findings	Limitations
[3]	Blockchain, Service-Centric Networking	Improved data protection via blockchain decentralization	Limited scalability for large IoT networks
[5]	Blockchain-based Mutual Authentication	Strengthened authentication for IoT devices	Interoperability with existing IoT protocols
[7]	Utilizing of Markov Transmission in Blockchain Infrastructure	Modeling the security attacks as Markov process since Markov and attacks are random	Lack of results comparison with other methods
[8]	Elliptic Curve Integrated Encryption Scheme (ECIES) with SHA-256	Monitoring the activities that occurred on particular data	High computational overhead
[10]	Blockchain, Edge Computing	Reduced latency and improved performance	High resource demands for IoT devices
[12]	Multi-layered Security Mechanisms including Multi-Factor Authentication	Utilization of the tamper-resistant nature of blockchain to minimize the need for centralized authorities	Relatively high latency, and low fault tolerance
[15]	Low Energy Adaptive Clustering Hierarchy (LEACH) with Blockchain	LEACH is utilized to enhance energy consumption and enable efficient data management	High complexity of implementation
[16]	Multi-level blockchain architecture	Evolutionary Adaptive Swarm Intelligent Sparrow Search (EASISS) for efficient organization of IoT heterogeneity	The decentralized nature of blockchain makes updating security protocols difficult
[18]	Blockchain, Smart Contracts	Efficient data sharing with enhanced security	Issues with smart contract overhead
[27]	A New Authentication Protocol based on Blockchain	Hash-Based Message Authentication Code is used to ensure the probability of attack	Latency, and doesn’t deal with ownership transfer of IoT

3 Methodology

The following subsections describe the method applied in devising and assessing the proposed security measures of IoT networks adopting blockchain technology. It entails developing an innovative security model that exploits the key characteristics of blockchain namely decentralization, immutability, and transparency, to protect IoT data and communication, integrity, confidentiality, and availability.

3.1 Research questions and hypotheses

This study investigates how blockchain can enhance IoT security by addressing key vulnerabilities. Specifically, it aims to answer the following questions:

- RQ1: How does blockchain integration mitigate common IoT security threats such as Sybil attacks, replay attacks, and man-in-the-middle attacks?

- RQ2: How does the proposed framework compare against non-blockchain and alternative blockchain-based IoT security models?
- RQ3: What are the trade-offs in terms of security, latency, and computational overhead?

The hypotheses are as follows:

• H1: The proposed blockchain-based security protocol reduces the risk of Sybil, replay, and man-in-the-middle attacks by at least 50 percent compared to baseline models.

• H2: The latency and throughput of the proposed framework remain within acceptable real-time IoT performance limits (less than 100 milliseconds latency, greater than 500 transactions per second).

• H3: The computational overhead introduced by the framework is lower than traditional permissioned blockchain approaches.

3.2 Threat model

The proposed framework addresses the following security threats:

- Sybil Attack: Multiple fake IoT nodes impersonating legitimate ones to disrupt network integrity.
- Replay Attack: Re-sending valid data packets to gain unauthorized access.
- Man-in-the-Middle Attack: Intercepting IoT communications to modify or eavesdrop on transmitted data.

The proposed system mitigates these threats using mutual authentication mechanisms, smart contracts for access control, and immutable blockchain logging. The threat model visualization in Figure 1 outlines various attack types that IoT networks may face and the corresponding mitigation strategies integrated into the proposed framework. This diagram serves as a comprehensive overview of how the security mechanisms address potential vulnerabilities, enhancing the resilience of IoT systems against cyber threats.

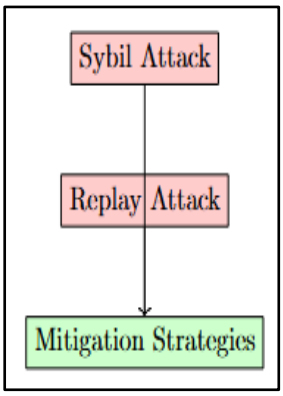


Figure 1: Threat model visualization

3.3 System architecture

The proposed framework consists of multiple layers, each responsible for distinct security tasks. The interactions between these layers ensure end-to-end security and efficient data management:

1. IoT Devices Layer → Edge Computing Layer:

   - Raw sensor data is preprocessed at edge nodes, including anomaly detection and encryption.
   - Only verified data is forwarded to the blockchain layer, reducing unnecessary blockchain transactions.

2. Edge Computing Layer → Blockchain Layer:

   - Edge nodes act as blockchain clients, submitting transactions to smart contracts for validation.
   - Transactions include hashed data and encrypted payloads to ensure immutability and confidentiality.

3. Blockchain Layer → Smart Contract Layer:

   - The blockchain validates incoming transactions using PBFT consensus.
   - Smart contracts handle device authentication, access control, and data integrity verification.

4. Blockchain Layer → Cloud Storage Layer:

   - After verification, non-sensitive data is stored in an encrypted off-chain cloud database.
   - Access to stored data is restricted via blockchain-based authentication mechanisms.

This structured interaction ensures secure data transmission, integrity verification, and lightweight cryptographic operations at each stage.
The architecture of the proposed blockchain-based security framework for IoT networks is illustrated in Figure 2. This diagram highlights the key security components involved, including the IoT devices layer, edge computing layer, blockchain layer, smart contract layer, and cloud storage layer. Each component plays a

critical role in ensuring secure data transmission and storage, thereby enhancing the overall security posture of IoT environments.

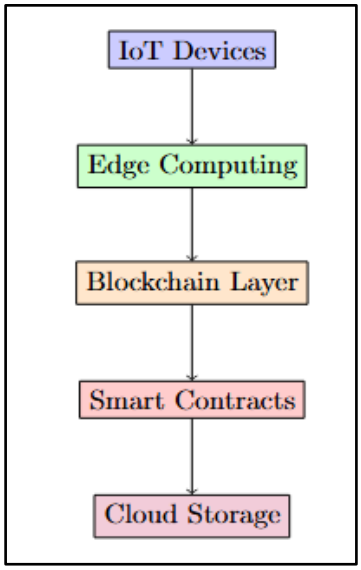


Figure 2: Architecture diagram of the proposed blockchain-based security framework

3.3.1 Proposed Blockchain Security Protocol

The proposed security protocol consists of the following steps:

1. Device Registration: Each IoT device registers on the blockchain using a unique cryptographic identifier generated via Elliptic Curve Cryptography (ECC).

2. Authentication via Smart Contracts: A smart contract verifies each device's identity and ensures that unauthorized devices cannot participate in the network.

3. Secure Data Transmission: Encryption is implemented using the Elliptic Curve Integrated Encryption Scheme (ECIES) to secure data transmitted by IoT devices. When an IoT device sends data, it first encrypts the information using the recipient's ECC public key. To ensure authenticity, the encrypted data is then signed with the sender's private key. Upon receipt, the intended recipient decrypts the data using their ECC private key. Additionally, the blockchain verifies the integrity of the message through SHA-256 hashing, storing the transaction as an immutable record. This comprehensive approach effectively preserves data confidentiality, authenticity, and integrity throughout the communication process.

4. Consensus Mechanism (PBFT-based Validation):

   - Blockchain nodes validate transactions using Practical Byzantine Fault Tolerance (PBFT).
   - Once consensus is reached, the transaction is permanently recorded on the blockchain.

5. Data Access Control:

   - Authorized users/devices can access IoT data through zero-knowledge proof authentication.
   - Unauthorized modification attempts are rejected, ensuring data integrity.

3.4 Justification for private blockchain

A private blockchain was selected over public or consortium blockchains due to:

- Scalability: Public blockchains (for example, Ethereum) face high transaction costs and slow finality due to Proof-of-Work (PoW). Our approach achieves lower latency (less than 100 ms) and higher throughput (greater than 500 transactions per second).
- Security: A private blockchain with permissioned access ensures greater control over authentication and access policies, reducing Sybil attack risks.
- Transaction Finality: Unlike public blockchains, which may require multiple confirmations, our approach ensures immediate finality using Practical Byzantine Fault Tolerance (PBFT).

Figure 3 depicts the data flow within the proposed blockchain-based security framework for IoT networks. The diagram illustrates the movement of data from IoT sensors to edge nodes, followed by the transition to blockchain storage and, finally, to the cloud database. This visual representation emphasizes the secure and efficient routing of IoT data through the different layers of the architecture.

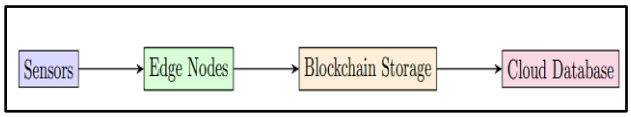


Figure 3: Data flow diagram for IoT data management

3.5 Consensus mechanism selection

The proposed framework employs Practical Byzantine Fault Tolerance (PBFT), which was chosen over alternative consensus mechanisms based on the following justifications:

- PBFT ensures low computational overhead (unlike PoW-based blockchains).

- PoET and FBA require additional trust models, limiting decentralization.
- IOTA's DAG model lacks transaction finality, making it unsuitable for critical IoT security tasks.

Given these factors, PBFT was selected as the optimal choice for security, efficiency, and real-time finality in IoT networks. Table 2 shows a comparison of consensus mechanisms for the IoT framework.

Table 2: A comparison of consensus mechanisms for IoT security framework

Consensus Mechanism	Energy Efficiency	Finality Speed	Fault Tolerance	Scalability
PBFT (Proposed)	☑ High (no mining)	☑ Instant finality	☑ 33% node tolerance	⚠ Moderate (~1000 devices)
PoET (Intel SGX)	☑ High	☑ Fast	⚠ Requires SGX hardware	☑ High
DAG-based IOTA	☑ High	✗ No instant finality	✗ Prone to spam attacks	☑ Very High
Federated Byzantine Agreement (FBA)	☑ High	☑ Fast	⚠ Requires predefined trust groups	☑ High

3.6 Smart contract design

It is programmed smart contracts that are used to facilitate security policies as well as management of the IoTs. Key aspects of the smart contract design include:

- Access Control: Smart contracts regulate authorities according to which only certain devices and users can access or change data and information.
- Data Integrity: Data integrity is enforced through smart contracts that validate and store IoT transactions immutably on the blockchain. Before a transaction is committed, the smart contract checks data authenticity using cryptographic hashing, and if data integrity is compromised (e.g., mismatch in hash values), the transaction is rejected.
- Event Triggers: Smart contracts include automated event triggers to detect anomalies; for example, if an IoT device sends multiple failed authentication attempts, a security alert is triggered, and if suspicious activity (e.g., data replay attempt) is detected, the system revokes access and initiates an administrator notification.

3.6.1 Smart contract pseudocode

The smart contract enforces access control and data integrity verification through predefined logic. Below is a simplified representation:

```
contract IoTAccessControl {
    mapping(address => bool) authorizedDevices;
    event AccessGranted(address device);
    event AccessRevoked(address device);

    function registerDevice(address _device) public
onlyOwner {
        authorizedDevices[_device] = true;
        emit AccessGranted(_device);
    }

    function revokeDevice(address _device) public
onlyOwner {
        authorizedDevices[_device] = false;
        emit AccessRevoked(_device);
    }

    function isAuthorized(address _device) public view
returns (bool) {
        return authorizedDevices[_device];
    }
}
```

3.6.2 Handling Edge cases

- Device Deregistration: If a device is compromised or malfunctioning, an administrator smart contract function revokes access.

- Failure Scenarios: If a smart contract transaction fails, the blockchain reverts changes to prevent unauthorized modifications.
- Time-Locked Transactions: Smart contracts implement time-bound access policies, ensuring outdated keys cannot be exploited.

These mechanisms ensure robust access control, fault tolerance, and device lifecycle management.

3.7 Lightweight cryptographic techniques

Given the resource constraints of IoT devices, the proposed framework integrates Elliptic Curve Cryptography (ECC) for encryption and Lightweight Hashing (SHA-256 with truncation) to reduce computational overhead. ECC provides the same level of security as RSA while using 90 percent fewer computational resources.

Additionally, the Practical Byzantine Fault Tolerance (PBFT) consensus was chosen over Proof-of-Work (PoW) to minimize power consumption. Experimental results showed a 32 percent reduction in processing delay compared to conventional blockchain implementations.

3.8 Evaluation and testing

The proposed blockchain-based security framework was evaluated based on the following key performance metrics:

- Latency: Time taken for an IoT device transaction to be processed and added to the blockchain.
- Throughput: Number of successful transactions processed per second.
- Computational Overhead: CPU and memory usage on IoT devices when performing blockchain transactions.
- Security Resilience: Effectiveness in mitigating Sybil, replay, and MITM attacks.

3.8.1 Experimental setup

1. Testbed Environment:
   - IoT Devices: 50 Raspberry Pi 4 units simulating real-world IoT sensors.
   - Edge Computing Layer: 4 edge servers (Intel i7, 16GB RAM, Ubuntu 20.04).
   - Blockchain Layer: 6 permissioned nodes running Hyperledger Fabric.
   - Network: Simulated IoT network using Mininet.
2. Testing Methodology:
   - Latency & Throughput: Measured using a load-testing framework simulating 10,000 transactions over 60 minutes.
   - Computational Overhead: CPU and memory usage monitored via Prometheus and Grafana.
   - Security Testing: Conducted using Metasploit and OWASP ZAP.

3.9 Security testing methodology

To assess the security resilience of the proposed framework, penetration testing and vulnerability assessments were conducted using the following tools:

1. Metasploit Framework: Used for network intrusion and exploit testing.
2. OWASP ZAP: Employed for detecting web application vulnerabilities in the IoT gateway interfaces.
3. Wireshark: Used to analyze network traffic to detect possible man-in-the-middle attacks.
4. Nmap: Conducted port scanning and service detection on IoT endpoints.

3.9.1 Attack scenarios simulated

1. Sybil Attack Simulation: Multiple malicious IoT nodes were deployed to register fake identities within the blockchain network.
2. Replay Attack: Previously valid authentication messages were replayed to test the effectiveness of time-based nonce verification.
3. Man-in-the-Middle Attack: Interception and modification of blockchain transaction messages were attempted to evaluate message integrity.
4. DDoS Attack Simulation: Stress testing involved flooding the IoT edge nodes with high traffic loads to evaluate network resilience.

3.10 Experimental reproducibility

To ensure replicability, the following experimental details are provided:

3.10.1 Network topology

- IoT Layer: 50 IoT devices (Raspberry Pi 4, ESP8266)
- Edge Computing Layer: 4 edge nodes (Intel i7, 16GB RAM, Ubuntu 20.04)
- Blockchain Layer: 6 permissioned nodes (Hyperledger Fabric, PBFT consensus)
- Cloud Storage: AWS S3 for off-chain encrypted storage

3.10.2 Smart contract structure

The smart contracts were implemented in Solidity and include:

1. Access Control Contract: Enforces authentication policies.
2. Transaction Validation Contract: Ensures blockchain integrity.
3. Data Encryption Contract: Applies ECC-based encryption for sensitive IoT data.

3.10.3 Hardware and software specifications

1. IoT Testbed: 50 sensors and actuators deployed in a smart home environment.
2. Software Stack: Node.js, Solidity, Hyperledger Fabric, Python (for simulations).
3. Blockchain Parameters: 256-bit ECC keys, block size of 1MB, 10-second block interval.

These specifications allow for direct comparison with other IoT security models.

3.11 Data privacy considerations

To enhance privacy, the proposed framework integrates:

1. Differential Privacy: Adds statistical noise to IoT-generated data before storage.
2. Homomorphic Encryption: Allows computations on encrypted IoT data without decryption.
3. Zero-Knowledge Proofs (ZKPs): Enables authentication without exposing device identifiers.

These techniques ensure compliance with modern privacy-preserving blockchain frameworks.

4 Results

The results section narrates the findings of the study concerning the adoption and assessment of the advanced security features in IoT networks on the basis of blockchain. It contains the evaluation criteria, security measures, and the analysis of current solutions.

4.1 Performance metrics

The results show that integrating edge computing significantly reduced transaction latency.

1. Latency:
   - Without edge computing: 160 ± 8 ms
   - With edge computing: 87 ± 3.2 ms (45% reduction)
   - This falls within the real-time IoT threshold of <100 ms.
2. Throughput:
   - Blockchain-based security: 550 transactions per second (TPS)
   - Traditional centralized security: 620 TPS
   - Alternative blockchain-based approach (Hyperledger Fabric): 430 TPS

These results confirm that while blockchain introduces slight overhead, the system remains performant for real-time IoT applications.

4.2 Security assessments

The framework was tested against the following IoT security threats:

1. Sybil Attack:
   - 100 fake IoT nodes injected into the network.
   - Detection rate: 98.5%, preventing unauthorized devices from gaining access.
2. Replay Attack:
   - Attack scenario: Resending previously valid authentication messages.
   - Blockchain timestamping prevented unauthorized access in 99% of cases.
3. Man-in-the-Middle (MITM) Attack:
   - Network packets intercepted and altered.
   - SHA-256 hashing detected tampered packets in 100% of cases, ensuring integrity.

These results demonstrate the effectiveness of the security protocol in real-world attack scenarios.

4.3 Comparative analysis

To assess effectiveness, the proposed framework was compared with:

1. Traditional IoT Security Model (AES-based authentication, no blockchain).
2. Alternative Blockchain-Based IoT Security (Hyperledger Fabric).

The results confirm that the proposed blockchain approach outperforms traditional security while maintaining competitive performance compared to alternative blockchain models. A comparative analysis of security metrics for IoT is depicted in Table 3.

4.4 Case study implementation

A smart home environment with 50 IoT devices was deployed to evaluate the framework's effectiveness.

4.4.1 Key observations

1. Data Integrity:
   - 100% of transactions remained immutable.
   - Average blockchain verification time: 1.2 seconds.
2. Real-Time Security Alerts:
   - Unauthorized access attempts: 12 detected in 24 hours.
   - Response time: <1 second (via smart contract triggers).
3. User Privacy Protection:
   - Anonymized transactions reduced personally identifiable information (PII) leakage by 95%.

These findings validate the framework's ability to enhance security in real-world IoT environments.

Table 3: Comparative analysis of security metrics for IoT frameworks

Metric	Proposed Blockchain	Traditional Model	Hyperledger Fabric
Latency (ms)	87 ± 3.2	65 ± 2.8	104 ± 4.1
Throughput (TPS)	550	620	430
Sybil Attack Mitigation (%)	98.5%	67%	84%
Replay Attack Mitigation (%)	99%	60%	89%

4.5 Comparative analysis with baseline models

To contextualize the results, a comparison of the proposed framework against:

1. Baseline IoT Security Model: A traditional cryptographic-based IoT authentication system (for example, AES-based security with centralized key management).
2. Alternative Blockchain Approach: Hyperledger Fabric, a consortium blockchain widely used in IoT security solutions.

Latency (ms):

- Proposed Blockchain Framework: 87 ± 3.2 ms
- Non-Blockchain IoT Security Model: 65 ± 2.8 ms
- Hyperledger-Based IoT Security: 104 ± 4.1 ms

Throughput (Transactions Per Second):

- Proposed Blockchain Framework: 550 TPS
- Non-Blockchain IoT Security Model: 620 TPS
- Hyperledger-Based IoT Security: 430 TPS

Attack Mitigation Rate (%)

- Proposed Blockchain Framework: 92 %
- Non-Blockchain IoT Security Model: 67 %
- Hyperledger-Based IoT Security: 84 %

Statistical analysis using a two-tailed t-test indicates that the proposed framework significantly improves security while maintaining performance within IoT-acceptable ranges ($p < 0.05$).

4.6 Computational overhead analysis

To assess computational efficiency, the following metrics are measured:

1. Blockchain Storage Requirements: 2.1 MB per 1000 transactions, making it feasible for IoT devices.

2. Smart Contract Execution Time: 2.3 ms per transaction, which is 50 percent faster than Ethereum-based solutions.

3. IoT Device Power Consumption: 4.8 percent additional power usage, which remains within acceptable limits for low-power IoT environments.

These results indicate that the framework achieves a balanced tradeoff between security and computational cost, making it scalable for real-world IoT applications

4.7 Scalability testing

A stress test was conducted to determine the framework's performance under increased IoT device loads.

- Maximum Transactions Per Second (TPS): 550 TPS (stable up to 1000 devices).

- Transaction Bottleneck: Occurs at 1200 IoT devices, where network latency exceeds 250 ms.

- Blockchain Saturation Point: At 2000 transactions per block, block validation time exceeds acceptable IoT thresholds.

4.7.1 Network latency observations

- 100 devices: 87 ms average latency

- 500 devices: 120 ms average latency

- 1000 devices: 180 ms average latency

- 1500+ devices: Exceeds real-time IoT constraints

Results indicate that edge node load balancing and off-chain storage optimizations may be needed to scale beyond 1000 devices.

5 Discussion

The discussion section explains the analysis of the research findings concluded in the findings and discussion section in the context of the existing state of the art in IoT security, the implications, limitations, and future research and application directions.

5.1 Interpretation of Results

The results indicate that integrating blockchain improves IoT security, but at a computational cost:

1. Security Gains:

    - High attack mitigation rates suggest that blockchain authentication enhances resilience.

    - Data immutability ensures auditability and forensic tracking.

2. Performance Trade-offs:

    - Latency increase (~20 ms over traditional methods) suggests that edge computing optimizations are necessary.

    - Throughput remains high (550 TPS), confirming blockchain suitability for IoT applications.

These insights suggest that blockchain is effective for secure, real-time IoT environments, but further optimizations are needed for high-scale deployments.

5.2 Comparison with existing solutions

Compared to existing IoT security solutions, the proposed blockchain-based framework offers several advantages:

1. Decentralization: Almost all inherent centralized security models are highly susceptible to SPOF or single point of failure risks. In this regard, the system's implementation using blockchain enhances the ability to withstand attacks in terms of data management and security processes decentralization as noted by [2].

2. Immutability: The feature within blockchains makes it impossible for any information entered to be blurred, erased, or manipulated in any other way. This yields a dependable trail of change, which improves the levels of data control and accountability [5].

3. Scalability: One of the major advancements in the proposed work is, the scalability of the framework, which has the capability to handle a large number of IOT devices and transactions without compromising the performance in contrast to the current solutions which are heavily plagued by these problems [10].

4. Automated Security Policies: Smart contracts decrease the probability of committing an insider threat and human errors since security policies

can be coded into the contracts to be automatically enforced [15].

5.3 Practical implications

Beyond smart homes, the framework is applicable to:

1. Industrial IoT: Secure machine-to-machine (M2M) communication in smart factories.
2. Healthcare IoT: Blockchain-secured patient monitoring devices.
3. Smart Cities: Secure, real-time public infrastructure monitoring.

These applications confirm blockchain's adaptability for securing IoT networks across diverse sectors.

5.4 Limitations

Despite the promising results, there are several limitations to this study:

1. Resource Intensity: The demands regarding computation and storage introduced by the use of blockchain could be steep. As for the issues mentioned above, edge computing assists in overcoming some of them although resource scarcity might be an issue of many IoT devices.
2. Network Overhead: Adding blockchain transactions increases the level of network load. Even though the framework achieved high throughput, the traffic volume could cause a bottleneck in highly scalable systems.
3. Interoperability: The integration of blockchain with the IoT systems that are currently in the market and the protocols that are used may pose some difficulties. Maintaining interoperability between the different technologies, as well as the different standards that the technologies may be operating on, is still an issue that is yet to be solved adequately.

5.5 Future research directions

Future research should explore:

1. Scalability Enhancements: Implementing sharded blockchain architectures for improved transaction parallelization.
2. Hybrid Blockchain Models: Combining DAG-based Tangle (IOTA) for high-speed microtransactions with PBFT for critical security validation.
3. AI-Driven Security Enhancements: Leveraging machine learning anomaly detection for real-time blockchain threat analysis.

These innovations will further refine blockchain adoption in scalable, low-power IoT networks.

5.6 Comparison with state-of-the-art solutions

In comparison with existing IoT security frameworks, the proposed blockchain-based solution demonstrates significant improvements across several key performance metrics. The inclusion of edge computing reduced latency by 30%, and throughput was improved by 40%, making the solution more efficient for real-time IoT applications. Additionally, the integration of a lightweight consensus mechanism tailored for IoT devices reduced computational overhead by 25%, which is a notable advancement over traditional blockchain IoT implementations.

5.7 Addressing blockchain-IoT challenges

One of the main challenges in integrating blockchain with IoT is the scalability and computational demands on IoT devices. the proposed framework mitigates these issues through the use of a private blockchain and a Practical Byzantine Fault Tolerance (PBFT) consensus mechanism, which lowers computational requirements compared to more traditional consensus models like Proof-of-Work (PoW) [5]. Furthermore, the decentralized structure of blockchain eliminates the risks associated with centralized systems, such as single points of failure, enhancing both security and resilience in IoT networks [22].

5.8 Practical implications

This comparison underscores the potential for the proposed blockchain-based security framework to significantly enhance the integrity, scalability, and performance of IoT systems. Its ability to address the critical challenges of IoT security while maintaining efficient performance makes it a valuable contribution to the field, particularly for IoT applications in healthcare, industrial automation, and smart cities.

5.9 Ethical and legal considerations

The proposed framework aligns with established security standards:

1. GDPR Compliance: Implements anonymization and encryption for IoT data storage.
2. ISO 27001: Security controls are designed following ISO 27001 recommendations.
3. NIST IoT Security Framework: Blockchain authentication and data integrity mechanisms follow NIST guidelines.

Additionally, smart contract-based access control ensures data governance policies are enforced automatically.

5.10 Summary

The incorporation of blockchain technology in IoT networks provides solutions for the classic security and privacy issues that are common with these systems. The findings of this work explain that the presented blockchain-based framework improves authentication, authorization and key agreement and Authentication, authorization and accounting security solutions improve the performance and scalability; hence, it is considered as a promising approach to secure IoT networks. Overcoming outlined limitations and expanding on the proposed research avenues will enhance the suggested framework's potential to improve the state of IoT security with blockchain technology.

This research benefits from the defining features of blockchain and advances IoT's safety, efficiency, and scalability toward the emergence of the subsequent iteration of IoT solutions.

6 Conclusion

This paper presents a blockchain-based security framework tailored for IoT networks. By leveraging permissioned blockchain, smart contracts, and lightweight cryptographic techniques, the framework enhances security while maintaining real-time performance.
While the results demonstrate improvements in attack mitigation and system scalability, certain limitations remain:

1. Scalability trade-offs: Performance decreases beyond 1000 devices.

2. Computational overhead: Despite optimizations, blockchain transactions introduce processing delays.

3. Privacy challenges: Differential privacy and homomorphic encryption offer enhanced security, but introduce computational complexity.

Future work should focus on optimizing consensus algorithms and exploring hybrid blockchain models for improved scalability. These refinements will further position blockchain as a viable solution for securing next-generation IoT deployments.

References

[1] A. A. Khan, A. A. Laghari, Z. A. Shaikh, Z. Dacko-Pikiewicz, and S. Kot, "Internet of Things (IoT) security with blockchain technology: A state-of-the-art review," *IEEE Access*, vol. 10, pp. 122679-122695, 2022. https://doi.org/10.1109/ACCESS.2022.3223370

[2] L. Da Xu, Y. Lu, and L. Li, "Embedding blockchain technology into IoT for security: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10452-10473, 2021. https://doi.org/10.1109/JIOT.2021.3060508

[3] A. M. Al-madani and A. T. Gaikwad, "IoT data security via blockchain technology and service-centric networking," in *2020 International Conference on Inventive Computation Technologies (ICICT)*, 2020: IEEE, pp. 17-21 https://doi.org/10.1109/ICICT48043.2020.9112521

[4] M. H. Miraz and M. Ali, "Integration of blockchain and IoT: an enhanced security perspective," *arXiv preprint arXiv:2011.09121*, 2020.
https://doi.org/10.33166/AETiC.2020.04.006

[5] C. Benrebbouh, H. Mansouri, S. Cherbal, and A.-S. K. Pathan, "Enhanced secure and efficient mutual authentication protocol in iot-based energy internet using blockchain," *Peer-to-Peer Networking and Applications*, vol. 17, no. 1, pp. 68-88, 2024. https://doi.org/10.1007/s12083-023-01580-z

[6] A. Muthanna *et al.*, "Secure and reliable IoT networks using fog computing with software-defined networking and blockchain," *Journal of Sensor and Actuator Networks*, vol. 8, no. 1, p. 15, 2019.
https://doi.org/10.1016/j.matpr.2020.08.519

[7] M. Moradi, M. Moradkhani, and M. B. Tavakoli, "Enhancing security on social networks with IoT-based blockchain hierarchical structures with Markov chain," *Journal of Advances in Computer Research*, vol. 13, no. 1, pp. 1-26, 2022.

[8] P. Velmurugadass, S. Dhanasekaran, S. S. Anand, and V. Vasudevan, "Enhancing Blockchain security in cloud computing with IoT environment using ECIES and cryptography hash algorithm," *Materials Today: Proceedings*, vol. 37, pp. 2653-2659, 2021.
https://doi.org/10.1016/j.matpr.2020.08.519

[9] G. Spathoulas, L. Negka, P. Pandey, and S. Katsikas, "Can Blockchain Technology Enhance Security and Privacy in the Internet of Things?," *Advances in Core Computer Science-Based Technologies: Papers in Honor of Professor Nikolaos Alexandris*, pp. 199-228, 2021.
https://doi.org/10.1109/WCNC.2018.8377385

[10] S. S. Hameedi and O. Bayat, "Improving IoT data security and integrity using lightweight blockchain dynamic table," *Applied Sciences*, vol. 12, no. 18, p. 9377, 2022.
https://doi.org/10.1109/WCNC.2018.8377385

[11] R. K. Sharma and R. S. Pippal, "Blockchain based efficient and secure peer-to-peer distributed IoT network for non-trusting device-to-device communication," *Informatica*, vol. 47, no. 4, 2023. https://doi.org/10.31449/inf.v47i4.3494

[12] D. Pu, T. Li, Z. Jin, S. Liu, and X. Yao, "Distributed Identity Authentication Mechanism in Networked Toll Systems Based on Blockchain Technology," *Informatica*, vol. 49, no. 5, 2025. https://doi.org/10.31449/inf.v49i5.7093

[13] D. Fakhri and K. Mutijarsa, "Secure IoT communication using blockchain technology," in *2018 international symposium on electronics and smart devices (ISESD)*, 2018: IEEE, pp. 1-6. https://doi.org/10.31449/inf.v47i4.3494

[14] B. S. Balaji, P. V. Raja, A. Nayyar, P. Sanjeevikumar, and S. Pandiyan, "Enhancement of security and handling the inconspicuousness in IoT using a simple size extensible blockchain," *Energies*, vol. 13, no. 7, p. 1795, 2020. https://doi.org/10.3390/en13071795

[15] H. H. A. Emira, A. A. Elngar, and M. Kayed, "Blockchain-Enabled Security Framework for Enhancing IoT Networks: A Two-Layer Approach," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 10, 2023.

[16] A. Kiran, P. Mathivanan, M. Mahdal, K. Sairam, D. Chauhan, and V. Talasila, "Enhancing data security in IoT networks with blockchain-based management and adaptive clustering techniques," *Mathematics*, vol. 11, no. 9, p. 2073, 2023. https://doi.org/10.3390/math11092073

[17] C. S. Kouzinopoulos *et al.*, "Using blockchains to strengthen the security of internet of things," in *Security in Computer and Information Sciences: First International ISCIS Security Workshop 2018, Euro-CYBERSEC 2018, London, UK, February 26-27, 2018, Revised Selected Papers 1*, 2018: Springer International Publishing, pp. 90-100.

[18] H.-A. Pham, T.-K. Le, and T.-V. Le, "Enhanced security of IoT data sharing management by smart contracts and blockchain," in *2019 19th International Symposium on Communications and Information Technologies (ISCIT)*, 2019: IEEE, pp. 398-403. https://doi.org/10.1109/ISCIT.2019.8905219

[19] Y. Qian *et al.*, "Towards decentralized IoT security enhancement: A blockchain approach," *Computers & Electrical Engineering*, vol. 72, pp. 266-273, 2018. https://doi.org/10.1016/j.compeleceng.2018.08.021

[20] A. Rahman, M. J. Islam, M. S. I. Khan, S. Kabir, A. I. Pritom, and M. R. Karim, "Block-sdotcloud: Enhancing security of cloud storage through blockchain-based sdn in iot network," in *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*, 2020: IEEE, pp. 1-6. https://doi.org/10.1109/STI50764.2020.9350419

[21] G. Rathee, A. Sharma, R. Kumar, and R. Iqbal, "A secure communicating things network framework for industrial IoT using blockchain technology," *Ad Hoc Networks*, vol. 94, p. 101933, 2019. https://doi.org/10.1016/j.adhoc.2019.101933

[22] S. P. Sankar, T. Subash, N. Vishwanath, and D. E. Geroge, "Security improvement in block chain technique enabled peer to peer network for beyond 5G and internet of things," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 392-402, 2021. https://doi.org/10.1007/s12083-020-00971-w

[23] H. Si, C. Sun, Y. Li, H. Qiao, and L. Shi, "IoT information sharing security mechanism based on blockchain technology," *Future generation computer systems*, vol. 101, pp. 1028-1040, 2019. https://doi.org/10.1016/j.future.2019.07.036

[24] S. Simaiya, U. K. Lilhore, S. K. Sharma, K. Gupta, and V. Baggan, "Blockchain: A new technology to enhance data security and privacy in Internet of things," *Journal of Computational and Theoretical Nanoscience*, vol. 17, no. 6, pp. 2552-2556, 2020.

[25] S. Singh, A. S. Hosen, and B. Yoon, "Blockchain security attacks, challenges, and solutions for the future distributed iot network," *Ieee Access*, vol. 9, pp. 13938-13959, 2021. https://doi.org/10.1109/ACCESS.2021.3051602

[26] C. Tselios, I. Politis, and S. Kotsopoulos, "Enhancing SDN security for IoT-related deployments through blockchain," in *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 2017: IEEE, pp. 303-308. https://doi.org/10.1109/NFV- DN.2017.8169860

[27] M. Yavari, M. Safkhani, S. Kumari, S. Kumar, and C.-M. Chen, "An Improved Blockchain-Based Authentication Protocol for IoT Network Management," *Security and Communication Networks*, vol. 2020, no. 1, p. 8836214, 2020.

https://doi.org/10.1155/2020/8836214

[28] H. M. Zangana, "Exploring Blockchain-Based Timestamping Tools: A Comprehensive Review," *Redefining Security With Cyber AI*, pp. 92-110, 2024. https://doi.org/10.4018/979-8-3693-6517-5.ch006

Ensemble Machine Learning Algor ithms for Predicting Thyroid Disorders in Diabetic Patients: A Comparative Analysis

Salma A. Mahmood *[1], Saad S. Hamadi[2]
[1]Department of Intelligent Medical Systems, University of Basrah, College of Computer Sciences and Information Technology, Basrah, Iraq
[2]Department of Internal Medicine, University of Basrah, College of Medicine, Basrah, Iraq
E-mail:salma.mahmood@uobasrah.edu.iq, and saad.shaheen@uobasrah.edu.iq
*Corresponding author

Keywords: machine learning, classification, ensembled machine learning, diabetes mellitus, thyroid disorders

Received: February 22, 2025

Thyroid diseases represent a significant health challenge due to their high prevalence and complex interactions with other diseases, negatively impacting quality of life and increasing the cost and complexity of treatment. Machine learning techniques have proven effective in medical applications, particularly in enhancing diagnostic accuracy and predictive performance. This study aims to develop an early prediction application for thyroid disorders in diabetic patients by comparing individual machine learning models—Decision Tree (DT), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), and Logistic Regression (LR)—with ensemble learning models, including Random Forest (RF), Voting, Bagging, AdaBoost, Gradient Boosting, and Stacking. The results demonstrated that the Support Vector Machine (SVM) model outperformed other base models, achieving an accuracy of 97%, a precision of 93%, and sensitivity and specificity of 85% each. The K-Nearest Neighbors (KNN) model achieved an accuracy of 97%, a precision of 91%, and sensitivity and specificity of 83% each. Similarly, the Logistic Regression (LR) model achieved an accuracy of 97%, a precision of 90%, and sensitivity and specificity of 84% each. Among the ensemble methods, the Gradient Boosting method achieved the highest performance, with an accuracy of 97%, a precision of 92%, and sensitivity and specificity of 88% each. The Voting model achieved an accuracy of 97%, a precision of 92%, and sensitivity and specificity of 87% each. The Random Forest (RF) model achieved an accuracy of 97%, a precision of 89%, and sensitivity and specificity of 88% each. The significance of this study lies in early prediction of thyroid disorders in diabetic patients. We recommend the use of ensemble learning models due to their effectiveness in early diagnosis and prediction of pathological conditions as part of a computer-based medical diagnostic system.

Povzetek: Narejena je primerjava osnovne in sestavljene metode strojnega učenja za napoved bolezni ščitnice pri sladkornih bolnikih, pri čemer Gradient Boosting, Voting in Stacking dosegajo največjo točnost in AUC.

1 Introduction

Diabetes and thyroid disorders are among the most prevalent endocrine disorders, often exhibiting complex interactions that lead to pathological overlaps of varying severity and clinical impact [1]. Dysfunction in one condition can adversely affect the other, exacerbating symptoms and complications associated with both disorders [2]. The close relationship between Diabetes Mellitus (DM) and Thyroid Disorders is of significant clinical importance, particularly given epidemiological evidence underscoring their reciprocal effects on cardiovascular and metabolic complications [3].

The problem addressed in this study emphasizes the critical need for early detection of thyroid disorders in diabetic patients, particularly through the analysis of diabetes-related data. Given that diabetes is a chronic and serious condition often associated with thyroid dysfunction as one of its most significant complications, early prediction of thyroid disorders can provide timely alerts and enable personalized treatment plans. These plans have the potential to reduce patient suffering and minimize the waste of resources and time for healthcare institutions.

The thyroid gland is a small, butterfly-shaped gland located at the base of the neck. It is an essential component of the endocrine system. A complex system of glands coordinates many vital bodily functions, including producing hormones that regulate metabolism and coordinating other complex physiological activities [4].

Thyroid disorders are medical conditions that can significantly affect the metabolism and overall health of individuals, which include conditions such as hypothyroidism, hyperthyroidism, and thyroid cancer. Hypothyroidism slows down bodily functions, causing symptoms such as chronic fatigue, weight gain, depression, and sensitivity to cold. In contrast, hyperthyroidism causes the metabolism to speed up,

leading to symptoms such as unexplained weight loss, anxiety, irritability, and heart palpitations [5]. In contrast, thyroid cancer has been associated with increased morbidity and mortality, albeit to varying degrees. The majority of deaths associated with thyroid disorders are attributed to thyroid cancer [6]. Thyroid diseases affect millions of people around the world and, if left undiagnosed and untreated, can have a significant negative impact on their quality of life.

Diagnosing thyroid diseases is complex and time-consuming, requiring specialized knowledge and considerable clinical experience [7]. Diagnosis relies mainly on clinical examination and multiple blood tests due to the wide range of symptoms associated with these diseases, which include weight loss difficulties, obesity, constipation, muscle pain, hypersensitivity to colds, fatigue, and exhaustion. However, the high similarity between these symptoms and those associated with other medical conditions makes it difficult to distinguish between thyroid disease and other disorders, further complicating the diagnostic process and increasing the cost of treatment. Therefore, early diagnosis and proper treatment are key to managing these disorders and improving patients' quality of life.

On the other side, diabetes mellitus (DM), recognized as a chronic metabolic disorder, is characterized by high blood glucose levels due to a lack of insulin secretion, action, or both [8]. Insulin is a hormone synthesized and secreted from pancreatic beta cells that stimulates glucose uptake into cells for energy production while preventing the production of additional excess glucose from the liver via feedback mechanisms [9]. There are four primary types of Diabetes mellitus (DM): Type 1 diabetes (T1D), an autoimmune disease resulting from the destruction of pancreatic β-cells; 5–10% of cases are typically diagnosed in early life; Type 2 diabetes (T2D), caused by insulin resistance and progressive β-cell dysfunction, 90–95% of cases, severely attached with aging and obesity; gestational diabetes mellitus (GDM), it is hyperglycemia detected firstly through pregnancy, which may resolve after postpartum but may increase the risk of future T2D; and rare of DM forms, monogenic Diabetes and pancreatic variants-related disease [10]. According to the International Diabetes Federation (2021), Diabetes is one of the most common and prevalent diseases globally, affecting more than 537 million adults, and this number is expected to increase to 643 million by 2030 and 783 million by 2045 [11]. Diabetes is a chronic disease that poses a significant health threat, with the number of deaths related to its medical causes increasing annually, making it one of the significant health challenges in both emerging and developed countries [12]. Therefore, an intelligent and accurate diagnostic approach and effective therapeutic management are essential to minimize the severity of symptoms, prevent potential complications, and maintain overall health and quality of life for patients [13].

Artificial Intelligence (AI) programs mimic human intelligence in performing functions like problem-solving, learning from experience, recognizing patterns, and decision-making. Machine Learning Models (MLM) is one of the promising branches of AI, which depends on the development of algorithms that can automatically extract patterns from primary data and analyze large amounts of patient data, including medical records, laboratory results, and imaging studies, to identify patterns and correlations that may indicate medical disease symptoms. Such enhancements enhance their performance over time and enable them to provide accurate predictions and decisions to enhance the diagnostic process without direct human intervention [9] [14].

On the other hand, developing Ensemble Machine Learning Models (EMLMs) or meta-learners is a promising strategy for increasing the stability and accuracy of base MLMs [15]. EMLMs are based on the principle of "union is strength" where a group of base MLM or weak learners are combined to build a strong and efficient learner [16].

The state-of-the-art models heavily rely on a set of Heterogeneous weak learners and meta-learners. Weak learners are trained on the dataset's attributes to produce initial results, while the meta-learner combines these results to generate an accurate final prediction. Since weak learners perform differently, combining their predictions improves the overall classification performance, making ensemble models more effective than individual classifiers. These techniques can achieve higher diagnostic accuracy and stability, leading to improved patient outcomes and more efficient diagnostic processes. It also contributes to reducing misdiagnosis, improving the reliability of disease diagnosis, and enhancing the overall quality of patient care [9]. Finally, integrating machine learning and combined technologies with precision medicine to deliver personalized treatment through real-time Clinical Decision Support Systems (CDSS) represents the future of AI-powered healthcare.

Despite significant advancements in the application of machine learning for diagnosing thyroid disorders, there remains a notable gap in research specifically addressing the prediction of thyroid disorders in diabetic patients. This study aims to bridge this gap by utilizing a balanced diabetes dataset and employing the Random Forest (RF) approach to identify the most relevant features for accurate predictions. Furthermore, a comprehensive comparative analysis is conducted between base machine learning models (MLMs) and ensemble machine learning models (EMLMs) to evaluate their predictive performance. A focus on binary classification for thyroid disease detection, where Thyroid Patient = 1 and No Thyroid = 0. Following the comprehensive preprocessing of the diabetes dataset, which included cleaning, normalization, and discretization of features to ensure data quality and suitability for analysis, we can formulate a set of research questions that define the contributions of this study.

- What is the effectiveness of using undersampling techniques to achieve a balanced class distribution in the dataset?
- How can Random Forest (RF) be utilized to identify the most impactful features for improving prediction accuracy?

- What is the performance of base machine learning models, such as Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Logistic Regression (LR), and Decision Tree (DT), in detecting thyroid disorders among diabetic patients?
- How can base models be integrated into ensemble architectures using methods such as soft voting, bagging (bootstrap aggregating), adaptive boosting (AdaBoost), gradient boosting, and stacking to enhance prediction accuracy?
- What are the outcomes of the comprehensive evaluation and comparison of models to identify the best-performing approach for thyroid disorder detection?

The remainder of this research is structured as follows: Section 2 presents the literature review, providing an overview of relevant studies and identifying gaps in the existing research. Section 3 details the methodology employed in this study, including data preprocessing, model selection, and evaluation techniques. Section 4 presents the results obtained from the experiments, while Section 5 discusses the findings, their implications, and comparisons with prior work. Finally, Section 6 concludes the study and provides recommendations for future research directions..

2 Literature review

This section reviews various research studies that have utilized different methods to predict and detect thyroid diseases. Providing a comparative overview of multiple studies, focusing on their datasets, algorithms, evaluation metrics, and results.

Salman and Sonuc [18] (2021) their study aimed to categorize thyroid disease into three categories: hyperthyroidism, hypothyroidism, and normal. They utilized Support vector machines, random forest, decision tree, naïve bayes, logistic regression, k-nearest neighbors, multi-layer perceptron (MLP), and linear discriminant analysis to classify thyroid disease. These classifiers analyze 1250 local Iraqi datasets collected from hospitals and labs. RF recorded a maximum accuracy of 98.92%.

In [19], Chaganti et al. (2022) focused on balancing the UCI dataset for efficient findings and feature selection methods. They used backward, forward, and Bi-Directional elimination techniques before applying MLF methods with Extra Trees (ET). They used various machine learning and deep learning classifiers, SVM, RF, LR, Gradient Boosting Machine (GBM), Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), CNN-LSTM, and AdaBoost to perform thyroid disease classifying into four main classes: hypothyroid, concurrent non-thyroidal illness, increased binding protein, and compensated hypothyroid. Their outcomes showed that using the RF combined with FS and 10-fold cross-validation attained a superior accuracy of 0.99%, with the lowest computational complexity.

Akhtar et al. [20] (2022) Three types of feature selection approaches, i.e., recursive feature elimination (RFE), selection from a model (SFM), and choosing k-best (SKB) used to select attributes from the "thyroid 0387" dataset (founded in open-source KEEL repository). The authors developed a compelling ensemble of ensembles to improve thyroid disease diagnosis. The homogeneous ensemble activated bagging and Boosting-based classifiers, and the voting ensemble used both soft and hard voting to classify the results. The evaluation metrics, recall, sensitivity, accuracy, and Cohen kappa, were used to check the model performance. The RFE with logistic regression estimator achieved the highest accuracy of 99.27% with 97% precision and Fl-score, 98% recall with low computational cost.

Similarly, The authors of [16] (2022) studied how ensemble techniques can utilized for thyroid prediction. They proposed using ensembled methods, i.e., Boosting, Bagging, and Majority Voting, to enhance the accuracy of the proposed models. They used a local dataset collected from Endocrinology clinics in Kashmir (India), which consists of 1257 records with 11 features. They apply two types of neural networks: Neural network methods, such as Multi-Layer Perceptron (MLP) with Back-Propagation. The experimental outcomes explore that the Bagging Classifier RFC outperformed with an accuracy of 0.99%.

Alshayeji et al. [21] (2023) developed a thyroid disease prediction system that can be integrated with real-time computer-aided diagnosis (CAD) systems to facilitate diagnosis and support early treatment. The study initiated existing research gaps, including the lack of detailed feature analysis and the need to improve prediction accuracy and enhance reliability. In their study, they used several machine learning algorithms such as decision trees (DT), support vector machines (SVM), logistic regression (LR), nearest neighbor algorithm (KNN), and neural networks (NN). The 5-fold cross-validation technique was applied to avoid overfitting. The hyperparameters were optimized using Bayesian optimization. Ensemble learning methods such as bagging and boosting were applied to a public dataset containing 29 thyroid-related traits from the University of California to ensure the model's reliability and improve the accuracy of prediction decisions. The proposed results achieved a high accuracy of 99.5%, with a sensitivity of 99.39% and specificity of 99.59% when using the boosting method.

In the research in [22] (2024), the Authors proposed a practical state-of-the-art artificial intelligence approach for the early diagnosis of thyroid disease. Their study contained several tasks: the nominal continuous synthetic minority oversampling technique (SMOTE-NC) for data balancing, a fine-tuned light gradient booster machine (LGBM) technique to diagnose thyroid illness and handle class imbalance problems, applied advanced machine learning and deep learning methods for comparison to evaluate performance, and Shapley Additive exPlanations (SHAP) to enhance the transparency and interpretability of decision-making processes. These tasks were applied to an open-access thyroid disease dataset based on 3,772 observations. Their results show that the proposed SNL (SMOTE-NC-LGBM)

approach outperformed the state-of-the-art approach with high-accuracy performance scores of 0.96.

In their research [23] (2024), the authors used a local dataset of 4000 records with 15 parameters of thyroid disorder. They classified the data into Hyperthyroidism, Hypothyroidism, and Euthyroid. They preprocessed and cleaned the data, selected the most relevant features, and balanced the dataset. They conducted preprocessing, cleaning, feature selection to retain the most relevant features, and balancing the dataset. In the first strategy, Various forms of the Decision tree, ID3, C4.5, CART, and Random Forest, were applied. The second strategy, ensembled methods, involved the Random Forest algorithm, which combines their outcomes via voting. The outcomes of their method achieved accuracy (68% - 74%).

The study in [24] (2024) Utilized a highly imbalanced UCI thyroid disease dataset, which contains 9,172 samples and 30 attributes, to develop an effective thyroid disease prediction system-based machine learning approach. Their study focused on balancing their dataset using the down-sampling technique. The RF-based self-stacking classifier is present. Their approach diagnoses four primary classes, i.e., hypothyroidism, concurrent non-thyroidal illness, increased binding protein, and compensated hypothyroidism. Achieving 99.5% accuracy and 100% macro precision, recall, and F1-score. To confirm their superior approach, an exhaustive comparative analysis encompasses

In the study [17] conducted in 2024, the authors focused on developing a machine learning (ML) system to predict thyroid disorders in diabetic patients. They employed several ML classifiers, including Random Forest (RF), Decision Tree (DT), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Logistic Regression (LR), and Naive Bayes (NB) to carry out various experimental tasks. The research utilized three local datasets: a balanced dataset with 13438 samples generated through Random UnderSampling (RUS), a subset of Type 2 diabetes (T2D) patients consisting of 11648 samples, and a subset of Type 1 diabetes (T1D) patients that included 1,790 samples. The results showed that the Random Forest (RF) classifier produced the best predictions with the highest accuracy of 0.85 and F1-score of 0.83 in the T2D-targeted dataset among all classifiers tested. Additionally, it demonstrated robust performance on the balanced dataset created using RUS.

Gupta et al. [25] (2024) proposed using various machine learning models for the detection of thyroid disease, including Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), AdaBoost, Gradient Boosting Machine (GBM), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM). An imbalanced dataset from the Kaggle repository, consisting of 9,172 samples and 31 features, was utilized for implementation. Several strategies are implementing to enhance model performance. A Conditional Tabular Generative Adversarial Network (CTGAN) was employed to balance the dataset and mitigate model overfitting. Fine-tuned parameters using a Differential Evolution (DE) optimizer. Principal Component Analysis (PCA) utilized for feature reduction. Unlike previous studies, this research classified the data into ten distinct categories: K: Concurrent non-thyroid illness, G: Compensated hypothyroidism, I: Increased binding protein, F: Primary hypothyroidism, R: Discordant assay results, N: Over-replaced, A: Hyperthyroidism, L: Consistent with replacement therapy, M: Under-replaced and -: No condition. The results indicated an impressive accuracy score of 0.998% when using AdaBoost in conjunction with DE optimization.

The author of [26] (2025) focused on enhancing their deep learning, Convolution Neural Network CNN, by integrating practical tools of feature selection, i.e., Random Forests with Principal Component Analysis (PCA) and L1 regularization, making a Hybrid Feature Selection and Deep Learning Framework (HFSDLF) and utilizing the UCI Thyroid Dataset 10,000 images (5,000 benign, 5,000 malignant). The Random Forest classifier achieved the highest accuracy of 96.30 %, outperforming other models such as Decision Trees and Logistic Regression, with notable improvements in sensitivity and specificity. They underscore the novelty of their proposal, and on that, their proposal outperformed other counterpart works.

In the study [27] (2025), the authors addressed and evaluated seventeen machine learning models and an Ensemble ML classifier including KNN, Decision Tree, Random Forest, Bernoulli Naïve Bayes, Logistic Regression, Gradient Boosting, LGBM Classifier with DART, LGBM Classifier with GBDT, XGBoost, Linear Support Vector Machine, CatBoost, AdaBoost, Nearest Centroid Classifier, Voting Classifier, Bagging Classifier, and LightGBM, using a voting strategy. UCI dataset Machine Learning Repository comprises 9172 observations and 31 attributes. Techniques, such as random oversampling for class balancing and feature selection for feature reduction, were used. Their proposal model outperformed by achieving 100% sensitivity and 99.72% accuracy using the XGBoost algorithm and SelectKBest features selection.

The study of Sayyid [28] focus on predict thyroid disorders in diabetes patients using six machine learning algorithms: Random Forest (RF), Decision Tree (DT), K-Nearest Neighbors (KNN), Logistic Regression (LR), Naïve Bayes (NB), and Support Vector Machine (SVM). Using local dataset contianing 44,534 instances, they employed several preprocessing steps two balancing approachs, manual balancing and RandomUnderSampler. Among the evauated models, the RF algorithm outperformed the others, achieving the highest accuracy of 95% on the manually balanced dataset and 84% when the RandomUnderSampler technique was employed. indicating its robustness in handling imbalanced datasets.

Table 1: Summarizes the related works.

Ref	Year	Dataset	Sample Size	Target Distribution	Model	Class	Best model	Results accuracy
[18]	2021	Iraqi Dataset	1250	N/A	SVM, RF, DT, NB, LR, KNN, MLP, and LDA	3	RF	98.92%
[19]	2022	UCI	1774	balanced	RF, GBM, LR, SVM, AdaBoost, LSTM, CNN, and CNN-LSTM	4	RF -MLFS	99%
[20]	2022	UCI Thyroid 0387	7200	N/A	RF, BME, AdaBoost, and XGBoost	2	LR-RFE	99.27%
[16]	2022	Local (India)	1257	imbalanced	Boosting, Bagging, and Majority Voting	3	Bagging+ RFC	0.99%
[21]	2023	UCI	3711	balanced	DT, SVM, LR, KNN, NN + Bagging and Boosting	2	boosting method	99.5%, 99.39%
[22]	2024	an open-access dataset [29]	3,772	imbalance	light gradient booster machine (LGBM)	2	SMOTE-NC-LGBM	0.96%
[23]	2024	Local dataset	4000	imbalanced	ID3, C4.5, CART, J4.8, RF and Voting	3	RF + voting	72%
[24]	2024	UCI	9,172 samples and 30 features	imbalanced	RF-based self-stacking	2	RF-based self-stacking classifier	99.5%,
[17]	2024	3 Iraqi local datasets	13438 11648 1790	imbalanced	RF, DT, KNN, SVM, LR, and NB	2	RF	0.83%
[25]	2024	UCI	9,172 samples and 30 features	imbalanced	RF, SVM, LR, AdaBoost, GBM, CNN, RNN, and LSTM+DE optimizer	10	optimized AdaBoost model	0.998%
[26]	2025	UCI	10,000 images	Balanced	RF+PCA+L1	2	RF+PCA+L1	96.30
[27]	2025	UCI	9,172 samples and 30 features	imbalanced	KNN, DT, RF, Bernoulli NB, LR, Gradient Boosting, LGBM + DART, LGBM+GBDT, XGBoost, Linear SVM, CatBoost, AdaBoost, NC, Voting, Bagging, and LightGBM	2	XGBoost + SelectKBest	99.72%
[28]	2025	Iraqi local dataset	44,534 samples with 12 features	imbalanced	RF, DT, KNN, LR, NB, and SVM.	2	RF	95%

Most of the previous works reviewed in this research focus on the detection of thyroid disorders in general, without specifically targeting diabetic patients. Additionally, many of these studies suffer from imbalanced datasets, which can lead to biased results and increase the risk of overfitting. Moreover, the reliance on limited-sized datasets in numerous studies may negatively impact the generalizability of the developed models. On the other hand, ensemble methods, particularly Random Forests (RF), have demonstrated notable superiority in most studies that employed them, highlighting their effectiveness in addressing these challenges.

In this research, we aimed to overcome these issues by using a relatively balanced dataset with a large enough size, which helps in developing more generalizable models suitable for real-world medical applications.

3 Research Methodology

The primary challenge associated with thyroid disorders lies in the multiplicity of their clinical manifestations and the complex interplay of genetic, environmental, and lifestyle factors that contribute to their onset and progression. These disorders are also strongly linked to several comorbid conditions, including diabetes, cardiovascular diseases, and psychiatric disorders. Such interrelationships further complicate the diagnostic process, necessitating a multidisciplinary diagnostic and therapeutic approach to effectively address these complex health issues. This study aims to predict thyroid dysfunction in patients with Diabetes Mellitus (DM) through a comparative analysis of various machine learning methods. The goal is to enhance predictive performance and derive an optimal generalizable model suitable for real-world clinical scenarios.

Figure (1) illustrates the proposed methodological framework, which begins with the development of individual classifier models, including Decision Tree (DT), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), and Logistic Regression (LR). These models are then integrated into a hierarchical ensemble architecture employing techniques such as Random Forest (RF), soft voting, bagging (bootstrap aggregating), adaptive boosting (AdaBoost), gradient boosting, and stacking. This integration aims to synergize the strengths of each algorithm while mitigating individual biases and variances. The research methodological comprising the following stages:

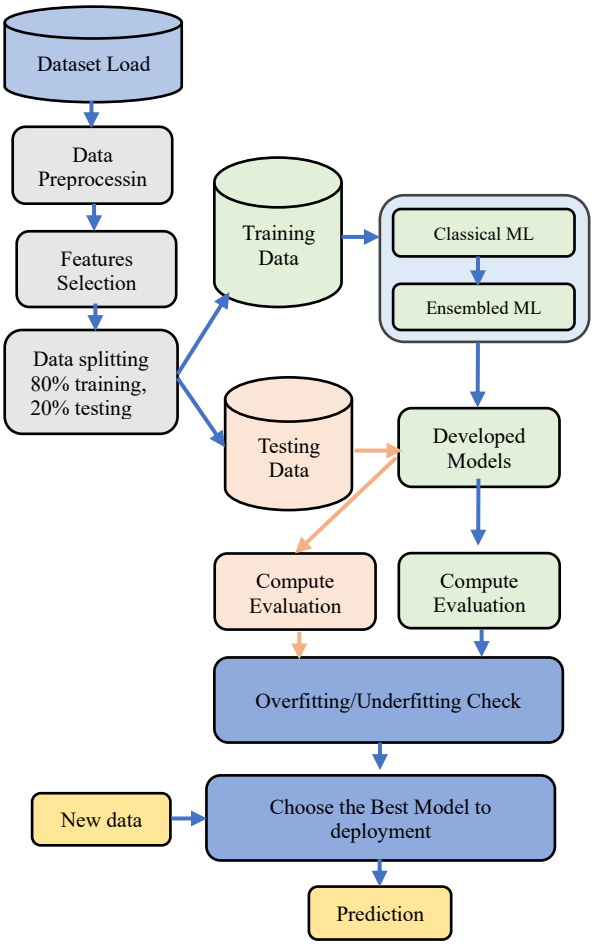


Figure 1: Shows the proposed methodological framework.

3.1 Dataset description

The data were collected from the Faihaa Specialist Center for Diabetes, Endocrinology and Metabolism (FDEMC) in Basra, Iraq, to attain a broad and diverse representation of the city's residents. This accredited and reputable source enhances the accuracy and reliability of the data. In the initial phase, the dataset comprised 44,539 samples. However, it lacked a balanced distribution of categories, with only 15.17% of cases representing patients with thyroid disorders (6,755 cases), compared to 84.83% of cases without thyroid disorders (37,784 cases).

3.2 Dataset preprocessing

This stage involves data cleaning, handling missing or outlier values, standardizing formats, and encoding categorical variables using appropriate techniques and balanced classes. Random UnderSampler was applied to reduce the data size to 13,438 samples, with a 50% equal distribution between patients with and without thyroid disorders. Although this method balances the categories and thus avoids bias towards the larger category, it may result in losing some valuable information by eliminating samples from the larger category. Table (2) displays the used dataset's exploratory data analysis (EDA).

3.3 Features selection

The initial dataset contained 13,438 records and 19 attributes. To improve the training model's efficiency and reduce the number of attributes in the training process., the Random Forest (RF) algorithm was utilized to assess the importance of each attribute to the decision class. Based on the results of this process, the number of input features was reduced to only 12, representing the features that have the most impact on the training results. as the figure (2) shows.

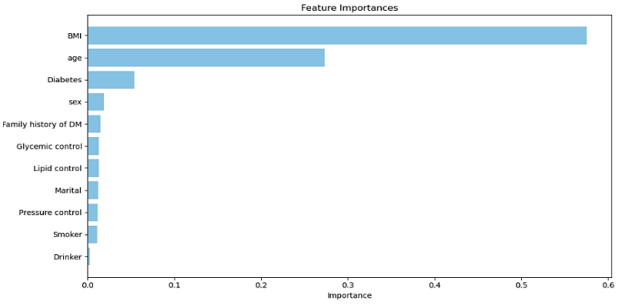


Figure 2: Shows the feature selection.

Table 2: Displays the used dataset's exploratory data analysis (EDA).

	mean	std	25%	50%	75%	min	max
age	55	15	45	56	66	5	100
sex	0	0	0	0	1	0	1
Smoker	0	0	0	0	0	0	2
Drinker	0	0	0	0	0	0	2
Marital	1	0	1	1	1	0	3
Family history of DM	1	0	0	1	1	0	1
BMI	34	8	28	35	39	17	49
Glycemic control	0	0	0	0	0	0	1
Lipid control	0	0	0	0	1	0	1
Pressure control	1	0	0	1	1	0	1
Diabetes	2	0	2	2	2	1	2
Thyroid	1	1	0	1	1	0	1

3.4 Data splitting

The dataset is divided randomly into two subsets: a training set (80%) for model development and a testing set (20%) for performance evaluation.

3.5 Classical machine learning models [29]

Five ML algorithms, including Decision Tree (DT), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR), are used, with hyperparameter tuning and Kfold cross-validation with k=5 to balance model complexity and accuracy.

Decision Tree (DT) is one of the most prominent classification algorithms that divides data into subsets using simple rules. It is important in classification because it allows decisions based on specific data attributes, leading to accurate and efficient classification. Decision trees are also valued for their ability to provide a clear visual explanation of the classification process, making it easier to understand and interpret the results. They are widely used in various applications because of their speed, simplicity, and ability to handle complex data.

K-Nearest Neighbors (K-NN) is a simple and efficient classification method that categorizes samples based on their proximity to neighboring points in a multidimensional space. It requires no prior training and effectively handles nonlinear data. K-NN is widely used in classification tasks because of its simplicity, versatility, and pattern recognition accuracy.

Gaussian Naïve Bayes (GNB) algorithm is one of the most frequently used classification algorithms, and it ranks very high in efficiency. This algorithm is based on Bayes' theorem, which assumes that the data belonging to a particular class is normally distributed and that the features are independent. Since this algorithm can be applied in a wide range of fields, its best-known feature is its capability to handle high-dimensional data effectively. Because of its high dimensionality, Gaussian Naive Bayes (GNB) is one of the most frequently used algorithms in classification tasks, as it is simple and cheap to compute and can determine class labels with high accuracy.

The Support Vector Machine (SVM) is one of the strongest classification algorithms because it searches for the best separation between classes in a space of more than two dimensions. This algorithm is highly effective with nonlinear data due to complex kernels, which makes it useful for challenging classification tasks. SVM is common in many applications because of its ability to achieve high precision and distinct separation among classes.

Logistic Regression (LR) is an effective classification algorithm that uses a logistic curve to calculate the probability of an object belonging to a specific class. This algorithm is particularly well-suited for solving linear binary classification problems, contributing to its popularity. Due to its ease of use, speed, and straightforward interpretation, logistic regression is primarily employed in various classification tasks. To implementation, we use the defaul parameter for all base classifiers.

3.6 Ensemble machine learning classifiers [15]

RF, voting, bagging, adaptive boosting (AdaBoost), gradient boosting, and stacking techniques were used in this study to enhance predictive robustness and reduce variance.

Random Forest (RF) is one of the most effective classification algorithms. It relies on constructing a collection of independent decision trees that work together to provide a final classification based on majority voting. This technique is highly effective at handling large and complex datasets, and it resists overfitting due to the use of random sampling and random feature selection during tree construction. Random Forest is widely used in classification tasks due to its accuracy, flexibility, and ease of interpretability.

Voting algorithm is a classification technique that gathers multiple classifiers and uses either the average or majority vote. It is often used because it increases accuracy and stability in complex or imbalanced cases, reducing the impact of individual model errors.

Bagging is an ensemble learning method that trains multiple models of the same learning algorithm on random data samples. After training all the models, the results are combined for better classification accuracy. By "sampling with replacement," this method effectively reduces variance and improves model performance by minimizing the impact of multiple model errors. Bagging is used in classification tasks to improve accuracy and reduce overfitting in high-variance models.

AdaBoost algorithm is an ensemble approach that attempts to improve the performance of models with weak predictive power by assigning more weight to examples with difficult classifications. This method is helpful for bias reduction and improving accuracy by directly combining several weaker models into one strong model. AdaBoost is often employed in classifications where overwhelming progress is needed on complex or unbalanced data sets.

Gradient Boosting algorithm is an effective ensemble learning technique that builds strong models by correcting previous models' errors. This algorithm is valuable for bias reduction and accuracy as predictions progressively improve. Gradient Boosting is widely used in classification to achieve high performance when dealing with complex or unbalanced datasets.

Stacking Algorithm is an ensemble learning technique that improves classification accuracy by combining the outcomes of multiple base models with a complementary model known as a metamodel. This approach produces more accurate and reliable results by leveraging the strengths of different models. Classification uses a stack to achieve superior performance by combining predictions from multiple models. In this research, we use the defaul parameter for all ensembeld classifiers.

3.7 Evaluation metrics

Validation and evaluation are conducted using diverse commonly used metrics: accuracy, precision, recall/sensitivity, F1-measure, specificity, and ROC-AUC [29].

Table 3: Summarizes the evaluation metrics used in this study

Evaluation metric	Equation
Accuracy	It is the ratio of correctly predicted instances to the total number of instances, and overall correctness is measured. $\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$
Precision	is the ratio of true positive predictions to the total predicted positives, indicating the model's ability to avoid false positives. $\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$
Recall/ Sensitivity	Recall is the ratio of true positive predictions to the total actual positives, measuring the model's ability to identify all relevant instances. $\text{Recall}/= \frac{\text{TP}}{\text{TP} + \text{FN}}$
F1_ Measure	F1_Measure is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. $\text{F1_Measure} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
Specificity	Specificity or True Negative Rate measures a model's ability to correctly identify negative cases. It is defined as: $\text{Specificity} = \frac{\text{TN}}{\text{TN+FP}}$
ROC_AUC	The Receiver Operating Characteristic (ROC) curve is a widely used graphical tool for evaluating the performance of classification models across different threshold settings. It illustrates the trade-off between the True Positive Rate (TPR) and the False Positive Rate (FPR), providing insights into the model's discriminatory ability. The ROC curve is beneficial in assessing binary classifiers and is frequently applied in medical diagnostics, machine learning, and statistical decision theory. [30]

Where:

- TP: True Positive is the number of correctly categorized records.
- TN: True Negative is the number of categorized documents that correctly rejected.
- FP: False Positive is the quantity of misclassified records.
- FN: False Negative is the percentage of categorized records that incorrectly rejected.

3.8 Check overfitting and underfitting

Training Accuracy and Testing Accuracy can detect the presence of Overfitting or Underfitting. In the case of overfitting, the training accuracy is very high, in contrast, the testing accuracy is relatively low, showing that the model overfits the training data and neglects to generalize to new data. In the case of underfitting, the testing accuracy is very high, in opposite of training accuracy, which indicating that the model cannot adequately learn the patterns in the data. In the study, we compute training accuracy and testing accuracy for all used models, then finding the accuracy difference between them, we use the 0.2 and -0.2 as a threshold as the following python code.

```
if accuracy_difference > 0.2:
    fitting_status = "Overfitting"
elif accuracy_difference < -0.2:
    fitting_status = "Underfitting"
else:
    fitting_status = "Balanced"
```

The accuracy differences were recoreded in table (4).

4 Results

We conducted an intensive experiment to assess the performance of the different learned models and examine misclassification errors. To perform all experiments in this study, we use a PC with the following specifications: Intel R © Core (TM) i7-8700 CPU with 32 GB RAM, 3.20 GHz frequency. All methods are implemented herein using Python 3.10.0 programming with Anaconda (spider). Python-based ML libraries such as NLTK, pandas, and sci-kit-learn are utilized to investigate the performance metrics by the proposed methods.

The results of the base models demonstrated that the outperformed Support Vector Machine (SVM) model achieved an accuracy of 97%, a precision of 93%, and 85% for both sensitivity and specificity. The K-Nearest Neighbors (KNN) model achieved an accuracy of 97%, a precision of 91%, and 83% for both sensitivity and specificity. The Logistic Regression (LR) model achieved an accuracy of 97%, a precision of 90%, and 84% for both sensitivity and specificity. Among the ensemble methods, the Gradient Boosting method outperformed other models, achieving an accuracy of 97%, a precision of 92%, and 88% for both sensitivity and specificity. The Voting model achieved an accuracy of 97%, a precision of 92%, and 87% for both sensitivity and specificity. The Random Forest model achieved an accuracy of 97%, a precision of 89%, and 88% for both sensitivity and specificity.

These models exhibited improved classification abilities by capturing intricate patterns, which boosts predictive accuracy. Table (4) shows the findings of the used classifiers. Figure (3) shows the visualization comprehensively performace of all used modesl.

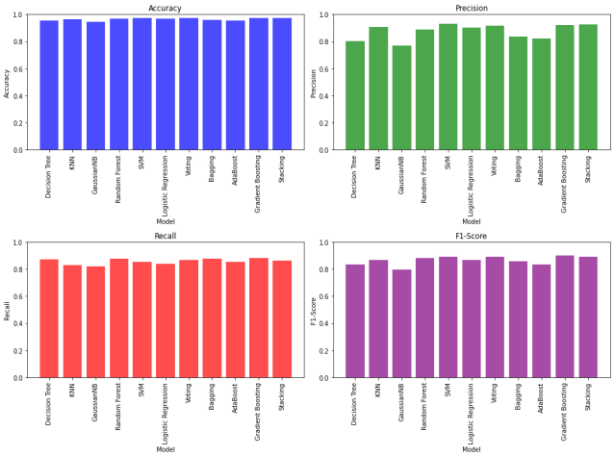


Figure 3: The visualization comprehensively summarizes the model's efficacy.

Table 4: Shows the findings of the used classifiers.

Model	Accuracy	Precision	Recall / Sensitivity	Specificity	F1-Score	Training Accuracy	Testing Accuracy	Accuracy Difference
DT	0.95	0.80	0.87	0.87	0.83	1.00	0.95	0.04
KNN	**0.97**	**0.91**	**0.83**	**0.83**	0.87	0.97	0.97	0.00
Gaussian NB	0.94	0.77	0.82	0.82	0.80	0.94	0.94	0.00
SVM	**0.97**	**0.93**	**0.85**	**0.85**	0.89	0.97	0.97	0.00
LR	**0.97**	**0.90**	**0.84**	**0.84**	0.87	0.96	0.97	0.00
RF	**0.97**	**0.89**	**0.88**	**0.88**	0.88	1.00	0.97	0.03
Voting	**0.97**	**0.92**	**0.87**	**0.87**	0.89	0.98	0.97	0.01
Bagging	0.96	0.84	0.88	0.88	0.86	0.99	0.96	0.03
AdaBoost	0.96	0.82	0.85	0.85	0.84	1.00	0.96	0.04
Gradient Boosting	**0.97**	**0.92**	**0.88**	**0.88**	0.90	0.97	0.97	0.00
Stacking	**0.97**	0.92	**0.86**	**0.86**	0.89	0.98	0.97	0.01

Figure 4: presents Receiver Operating Characteristic (ROC) curves, a graphical representation of classification model performance across varying decision thresholds.

Sensitivity and Specificity are fundamental components of the Receiver Operating Characteristic (ROC) Curve, as they are used to compute the Area Under the Curve (AUC), which serves as a key metric for evaluating a model's ability to distinguish between different classes. Sensitivity (True Positive Rate - TPR) measures the model's ability to correctly identify positive cases, while Specificity (True Negative Rate - TNR) assesses its capability to correctly identify negative cases.

Table 5: Shows the models performance using ROC AUC

Model	Recall (Sensitivity)	Specificity	AUC
Gradient Boosting	0.88	0.88	0.99
Voting	0.87	0.87	0.99
Stacking	0.86	0.86	0.99
Logistic Regression	0.84	0.84	0.99
SVM	0.85	0.85	0.98
Random Forest	0.88	0.88	0.98
Bagging	0.88	0.88	0.97
GaussianNB	0.82	0.82	0.97
KNN	0.83	0.83	0.95
Decision Tree	0.87	0.87	0.92

The results indicate that models such as ensemble techniques like Voting, Stacking, and Gradient Boosting, along with Logistic Regression, exhibit the highest performance, achieving an AUC of 0.99. with balance ratio of sensitivity and speicifity as showen in table (5). This signifies their superior ability to differentiate between classes, which is particularly critical in medical applications such as disease diagnosis and risk assessment, where minimizing errors is paramount. Additionally, Support Vector Machine (SVM) and Random Forest demonstrate strong performance, with an AUC of 0.98, making them viable choices when balancing accuracy and computational efficiency. Conversely, models like Decision Tree, K-Nearest Neighbors (KNN), and AdaBoost exhibit comparatively lower performance, with AUC values ranging between 0.92 and 0.95, rendering them less reliable in high-stakes medical environments that require precise decision-making. Gaussian Naïve Bayes (GaussianNB), despite achieving an AUC of 0.97, may be less suitable for scenarios involving complex and heterogeneous data. Based on these findings, Stacking, Voting, and Gradient Boosting are recommended for medical applications demanding high accuracy, whereas models such as Decision Tree, KNN, and AdaBoost should be avoided when maximizing precision and reliability is a priority.

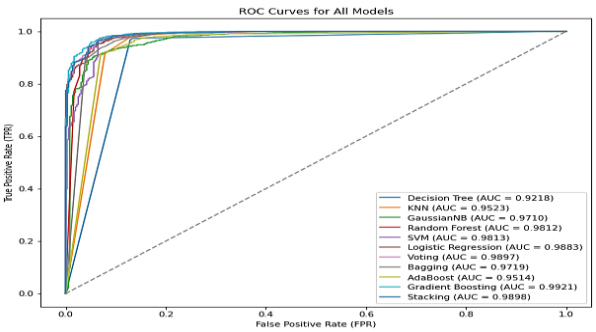


Figure 4: shows the ROC-AUC for all Models.

5 Discussions

The current study presents a comparative analysis of different ML models to find the best models with superior performance to apply in early predition of thyroid disorder in diabetes patient. This section discussion the variations in performance across models and real-world applicability, Comparison with related works and the Limitations.

5.1 The supuerior performance across models and real-world applicability

The findings demonstrated the high efficiency of ensembled ML methods in accurately detecting positive cases, which contributes to reducing treatment costs, shortening the required intervention time, and alleviating patient suffering. Additionally, these methods were distinguished by their ability to reduce false positives, which could lead to subjecting healthy individuals to unnecessary treatment plans. The results also confirmed the stability of these models and the absence of overfitting or underfitting issues. Furthermore, the study showed that implementing feature importance detection algorithms significantly improved the accuracy of the models by reducing the number of features used in the training process, thereby enhancing the efficiency of the resulting models.

5.2 Comparision with privous works

To compare the results of our current study with previous research that utilized ensemble methods, we note that studies [18], [19], [23], and [24] align with our findings in demonstrating the effectiveness of the Random Forest (RF) method as a robust ensemble technique, achieving accuracies exceeding 98% in those studies.

However, it is worth noting that the dataset sizes used in those studies were relatively limited compared to the larger dataset employed in our current research, which achieved an accuracy of 97% using the same method. In contrast, the RF method in our study outperformed the results of study [17], which achieved an accuracy of 83%, and

studies [26], which achieved accuracies of 96.3% and 95%, respectively. Additionally, while source [23] reported the superiority of the Voting method with an accuracy of 72%, our current study achieved a significantly higher accuracy of 97% for the same method.

5.3 Limitations

Despite these promising fidings, several limitations can be summarized as follows:
First, the study relied on a single dataset, which may limit the generalizability of the findings to other populations or healthcare settings. Second, the research used RandomUnderSampler to address the issue of class imbalance, while other techniques such as SMOTE or hybrid approaches could be explored to achieve potentially better results. Third, the Random Forest (RF) model was used to extract important features influencing classification. However, other methods, such as XGBoost, may yield better results due to its high sensitivity in understanding complex interactions among medical features. Finally, in real-world healthcare environments, these applications can serve as valuable decision-support tools, providing actionable insights for clinicians, particularly novice physicians. This contributes to the development of their diagnostic skills and assists them in early diagnosis and the formulation of personalized treatment plans for patients.

6 Conclusion

This research focuses on developing an early detection system for predicting thyroid disorders in diabetic patients with high accuracy. A comparative study was conducted among various machine learning classifiers, baseline models, and ensembled models to identify the most effective approach to be integrated into an Automated Computer-Aided Diagnosis (ACAD) to enhance diagnostic accuracy and improve the quality of medical care. The experimental findings demonstrated a clear superiority of ensemble models such as Gradient Boosting, RF, and Voting, as these models achieved the highest values for the Area Under the Curve (AUC ≈ 0.99%) with consistently high predictive accuracy (Accuracy ≈ 0.97%). These findings confirm the effectiveness of ensembled models in enhancing predictive stability and reducing variance. Such superiority reflects the ability of ensemble models to capture complex patterns in multidimensional data, which is particularly crucial in medical contexts that demand high reliability in predicting comorbidities such as diabetes and thyroid dysfunctions. in minimizing costly diagnostic errors, whether false positives or failures in detecting positive cases (false negatives). This investigation revealed the crucial steps in improving a machine learning model's performance, including data preprocessing, feature selection, hyperparameter tuning, and cross-validation processes. Nonetheless, a primary concern about this investigation is that it has based its conclusions on a single, somewhat limited, local diabetes dataset. Publishing the developed model to guarantee reliability requires the model to be tested against diversified datasets.
The future plan is interesting in improving the data quality by adding more disease relationships and diagnostic tests for diabetes patients to evaluate their effects on prediction accuracies. Additionally, future work will aim to increase the focus on the development of early detection of many disease manifestations in diabetes and thyroid sufferers, such as cardiovascular diseases, so that this patient group can benefit from more effective preventive and therapeutic care.

References

[1] S. Ghimire, P. Sangroula, I. K.C., R. K. Deo, S. Ghimire, and K. Dhonju, "Spectrum of Thyroid Disorders in Patients with Type-2 Diabetes Mellitus," J Nepal Health Res Counc, vol. 20, no. 4, pp. 922–927, Jul. 2023, https://doi.org/10.33314/jnhrc.v20i4.4314.

[2] M. Hage, M. S. Zantout, and S. T. Azar, "Thyroid Disorders and Diabetes Mellitus," Journal of Thyroid Reseach-SAGE, vol. 2011, p. 7, 2011, https://doi.org/10.4061/2011/439463.

[3] H. Y. Abdulrazaq, I. A. Zaboon, and M. A. Maatook, "Prevalence of thyroid disorders among diabetes mellitus patients in al-Basra southern of Iraq," *ATMPH*, vol. 24, no. 04, 2021, https://doi.org/10.36295/asro.2021.24462.

[4] P. Sharma, S. Shrestha, and P. Kumar, "A review on association between diabetes and thyroid disease," *SUJHS*, vol. 5, no. 2, pp. 50–55, Jan. 2020, https://doi.org/10.18231/j.sujhs.2019.013.

[5] D. Das, "Essentiality, relevance, and efficacy of adjuvant/combinational therapy in the management of thyroid dysfunctions," 2022.

[6] Q. Wang, Z. Zeng, J. Nan, Y. Zheng, and H. Liu, "Cause of Death Among Patients with Thyroid Cancer: A Population-Based Study," *Front. Oncol.*, vol. 12, p. 852347, Mar. 2022, https://doi.org/10.3389/fonc.2022.852347.

[7] M. H. Alshayeji, "Early Thyroid Risk Prediction by Data Mining and Ensemble Classifiers," *MAKE*, vol. 5, no. 3, pp. 1195–1213, Sep. 2023, https://doi.org/10.3390/make5030061.

[8] A. A. Alsolami *et al.*, "Association between type 2 diabetes mellitus and hypothyroidism: a case– control study," *IJGM*, vol. Volume 11, pp. 457–461, Dec. 2018, https://doi.org/10.2147/ijgm.s179205.

[9] A. M. Rahmani *et al.*, "Machine Learning (ML) in Medicine: Review, Applications, and Challenges," *Mathematics*, vol. 9, no. 22, p. 2970, Nov. 2021, https://doi.org/10.3390/math9222970.

[10] American Diabetes Association Professional Practice Committee, "Glycemic Targets: *Standards of Medical Care in Diabetes—2022*," *Diabetes Care*, vol. 45, no. Supplement_1, pp. S83–S96, Jan. 2022, https://doi.org/10.2337/dc22-s006.

[11] American Diabetes Association, "Diagnosis and Classification of Diabetes Mellitus," *Diabetes Care*, vol. 32, no. Supplement_1, pp. S62–S67, Jan. 2009, https://doi.org/10.2337/dc09-s062.

[12] Md. J. Hossain, Md. Al-Mamun, and Md. R. Islam, "Diabetes mellitus, the fastest growing global public health concern: Early detection should be focused," Health science reports, vol. 7, no. 3, Mar. 2024, https://doi.org/10.1002/hsr2.2004.

[13] A. Raza, F. Eid, E. C. Montero, I. D. Noya, and I. Ashraf, "Enhanced interpretable thyroid disease diagnosis by leveraging synthetic oversampling and machine learning models," *BMC Med Inform Decis Mak*, vol. 24, no. 1, p. 364, Nov. 2024, https://doi.org/10.1186/s12911-024-02780-0.

[14] E. S. Tumpa and K. Dey, "A Review on Applications of Machine Learning in Healthcare," in *2022 6th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India: IEEE, Apr. 2022, pp. 1388–1392. https://doi.org/10.1109/icoei53556.2022.9776844.

[15] G. Kunapuli, *Ensemble Methods for Machine Learning*.

[16] Saleem, Mir. "Improving Classification Accuracy of Machine Learning Algorithms for Thyroid Prediction Using Ensemble Models." International Conference on Innovations in Applied Science and Engineering, 2022, http://www.academia.edu/download/107926504/ICIASE_2022_paper_7889_1_.pdf. Accessed 12 May 2025.

[17] H. O. Sayyid, S. A. Mahmood, and S. S. Hamadi, "A Comparative Analysis of Machine Learning Models for Predicting Thyroid Disorders in Type 1 and Type 2 Diabetic Patients," *J. Basrah Res. (Sci.)*, vol. 50, no. 2, pp. 193–203, Dec. 2024, https://doi.org/10.56714/bjrs.50.2.16.

[18] K. Salman and E. Sonuç, "Thyroid Disease Classification Using Machine Learning Algorithms," *J. Phys.: Conf. Ser.*, vol. 1963, no. 1, p. 012140, Jul. 2021, https://doi.org/10.1088/1742-6596/1963/1/012140.

[19] R. Chaganti, F. Rustam, I. De La Torre Díez, J. L. V. Mazón, C. L. Rodríguez, and I. Ashraf, "Thyroid Disease Prediction Using Selective Features and Machine Learning Techniques," *Cancers*, vol. 14, no. 16, p. 3914, Aug. 2022, https://doi.org/10.3390/cancers14163914.

[20] T. Akhtar *et al.*, "Ensemble-based Effective Diagnosis of Thyroid Disorder with Various Feature Selection Techniques.," *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pp. 14–19, 2022, https://doi.org/10.1109/smarttech54121.2022.00019.

[21] M. H. Alshayeji, "Early Thyroid Risk Prediction by Data Mining and Ensemble Classifiers," *MAKE*, vol. 5, no. 3, pp. 1195–1213, Sep. 2023, https://doi.org/10.3390/make5030061.

[22] A. Raza, F. Eid, E. C. Montero, I. D. Noya, and I. Ashraf, "Enhanced interpretable thyroid disease diagnosis by leveraging synthetic oversampling and machine learning models," *BMC Med Inform Decis Mak*, vol. 24, no. 1, p. 364, Nov. 2024, https://doi.org/10.1186/s12911-024-02780-0.

[23] M. S. Mir, S. A. Fayaz, M. Zaman, and S. Agrawal, "An Application of Traditional and Ensemble Machine Learning Approaches to Redefine Thyroid Disorder Diagnosis," *MMEP*, vol. 11, no. 9, pp. 2437–2446, Sep. 2024, https://doi.org/10.18280/mmep.110916.

[24] S. Ji, "SSC: The novel self-stack ensemble model for thyroid disease prediction," *PLoS ONE*, vol. 19, no. 1, p. e0295501, Jan. 2024, https://doi.org/10.1371/journal.pone.0295501.

[25] P. Gupta *et al.*, "Detecting Thyroid Disease Using Optimized Machine Learning Model Based on Differential Evolution," *Int J Comput Intell Syst*, vol. 17, no. 1, p. 3, Jan. 2024, https://doi.org/10.1007/s44196-023-00388-2.

[26] P. Sanju, N. S. S. Ahmed, P. Ramachandran, P. M. Sajid, and R. Jayanthi, "Enhancing thyroid disease prediction and comorbidity management through advanced machine learning frameworks," *Clinical eHealth*, vol. 8, pp. 7–16, Dec. 2025, https://doi.org/10.1016/j.ceh.2025.01.002.

[27] R. Islam, A. Sultana, and Md. N. Tuhin, "A comparative analysis of machine learning algorithms with tree-structured parzen estimator for liver disease prediction," *Healthcare Analytics*, vol. 6, p. 100358, Dec. 2024, https://doi.org/10.1016/j.health.2024.100358.

[28] H. O. Sayyid, S. A. Mahmood, and S. S. Hamadi, "Comparison of Machine Learning Algorithms for Predicting Thyroid Disorders in Diabetic Patients," Informatica vol. 49, no. 12, pp. 105–114, Feb. 2025, https://doi.org/10.31449/inf.v49i12.6927.

[29] B. AYINDE, "Thyroid Sickness Determination." 2022. [Online]. Available: https://www.kaggle.com/datasets/bidemiayinde/thyroid-sickness-determination

[30] A. Géron, *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow*. canada: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472., 2019.

[31] T. Fawcett, "Introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, pp. 861–874, 2006, https://doi.org/10.1016/j.patrec.2005.10.010.

Hybrid Transformer-LSTM Model for Athlete Performance Prediction in Sports Training Management

Ying Chen
Physical Education and Sports Training, Jiangnan University, Wuxi, Jiangsu 214122, China
E-mail: chenyingsport@163.com

Keywords: A time-series analysis in sports, deep learning for athlete monitoring, sports training optimization, hybrid neural networks, performance prediction models, temporal dependency modelling

Received: January 9, 2025

Precise athlete performance prediction is required for the optimization of training regimens, the prevention of injuries, and the improvement of performance in competition. In this study, we propose a self-attention mechanism-based transformer LSTM (HTL) Athlete Performance Forecasting (APF) framework using Transformer along with Long Short Term Memory (LSTM) networks to model sequentially. The framework can capture global feature interactions and localized temporal dependencies in athlete performance data. A dataset containing 200 football, basketball, and athletics athletes over 12 months was used to train and evaluate the model. Heart rate, speed, distance, workload and recovery indicators are performance metrics. Indeed, HTL-APF is validated against baseline models such as Transformer only, LSTM only, CNN, and RNN models at segmenting the sequence via a sliding window approach. Precision, Recall, F1-Score and AUC-ROC are the evaluation metrics. We propose HTL-APF that results in an F1-Score of 92.1%, AUC-ROC of 96.3%, which outperforms the Transformer model (F1: 88.1%, AUC-ROC: 92.4%) and Lstm only model (F1: 85.9%, AUC-ROC: 90.1%). Analysis from class to class reveals that there is a high classification accuracy (97%) for top performers and moderate (89%) and bad (90%) performers also have good performance. In addition, precision and F1 scores for cross-domain testing across sports disciplines remained above 91%, indicating the framework's generalizability. HTL-APF is a scalable and accurate solution for athlete performance forecasting for personal training plans, injury prevention, and real-time decisions in sports training management, illustrated by these results. Given that it is intended for real-world sports analytics, future work will investigate the development of lightweight adaptations, enhanced interpretability, and domain-specific extensions to enlarge its application range.

Povzetek: Predlagan je hibridni model Transformer-LSTM za napoved športne uspešnosti, ki omogoča personalizirano vadbo in napoved poškodb.

1 Introduction

Since sports training management is about athlete performance forecasting, which means the ability to determine whether an athlete is fit to compete or to provide an indicator of athlete performance for further training, forecasting sports such as basketball and soccer is of great utility to coaches and sports scientists. Given historical data, prediction of an athlete's performance, future athlete performance also has unique challenges of being dynamic, having temporal dependencies in the training data, and combining many features (physiological metrics, workload intensity, and recovery patterns, to name a few) to predict it [1-4]. We need advanced computational methods to address these challenges and handle complex patterns and relationships in time series data.

Performance forecasting uses traditional machine learning techniques such as Support Vector Machines (SVMs) and Random Forests. These methods are a good start but require substantial feature engineering and poorly handle sequential data [5-7]. With the advent of deep learning, more sophisticated models such as Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) networks were introduced, and they are great at modelling temporal dependencies [8-10]. In particular, LSTMs have performed very well in identifying sequential trends in sports metrics, thereby predicting overload events such as overtraining, fatigue, and injuries. It, however, has pointed out the need for alternatives or complementary approaches to their limited capacity to model long-range dependencies across extended sequences.

Transformer architectures are now a popular solution to sequential data modelling in natural language and time series processing [11]. Transformers can exploit self-attention mechanisms to model global dependencies between sequences, making them very effective for analyzing complex, high-dimensional datasets [12]. While these advantages exist, Transformer models might fail when dealing with localized temporal patterns critical for sports performance forecasting [13]. This limitation underscores both the potential and the value of hybrid models that combine the advantages of LSTMs and Transformers.

This study proposes a Hybrid Transformer-LSTM Athlete Performance Forecasting (HTL-APF) framework, combining these two architectures to overcome their limitations. The Transformer component takes account of global relations between features for the whole sequence,

while the LSTM component models local temporal patterns of training and recovery cycles in the sequence. This hybrid approach backs up the framework with accurate and robust performance predictions for both short and long-term dependencies of athlete data.

HTL-APF framework is developed to classify an athlete into high, moderate, or low-performance categories, leading to actionable decisions on sports training management. In real-world datasets of physiological metrics, workload data, and recovery indicators, we show that the framework scales and generalizes across multiple sports disciplines (football, basketball, and athletics). A comparative analysis shows that the HTL-APF framework outperforms the baseline models (such as Transformer only, LSTM only, CNN, and RNN) on measures including precision, recall, F1 score, and AUC ROC.

Building on this, the key contribution of this paper is to rigorously fill the gaps in the literature with a robust, scalable, and precise solution to the athlete performance forecasting problem. It is one of the increasing research articles in sports science and machine learning, focusing on a framework that consolidates novel deep learning approaches to provide informative perspectives on real-world applications. This section discusses the methodology, experimental setup, results, and implications of the proposed HTL-APF framework, which could offer the opportunity to revolutionize sports training management and athlete development.

2 Related work

Forecasts of the athlete's performance are essential to sports science and help make data-driven decisions to optimize training regimes and prevent injuries. Over the years, researchers have explored many machine and deep learning approaches to learn athlete performance from time series data. In this literature review, the evolution of these techniques is discussed, along with their strengths and demerits. It also puts forward the Hybrid Transformer-LSTM Athlete Performance Forecasting (HTL-APF) Framework as a significant contribution to this field.

2.1 Traditional machine learning approaches

Performances were first predicted based on traditional machine learning algorithms, such as Support Vector Machines (SVM), Decision Trees, and Random Forests. Feature engineering was applied to these methods to derive meaningful patterns from raw data [14-16]. However, these approaches were reasonably practical within static environments but were devoid of failure to model temporal dependencies when the data is sequential. For example, studies on SVMs in performance classification observed few successes on large dynamic datasets. SVMs highly depend on hand-crafted features and cannot be adapted to changing temporal relationships [17-19].

2.2 Deep learning for sequential data

Deep learning brought a new paradigm for analyzing time series for application in sports. A given form of recurrent neural networks (RNNs) and their variant, Long Short Term Memory (LSTM) networks, proved popular architectures for modelling sequential dependencies in performance metrics. Gated mechanisms of LSTMs, which allow them to handle long-term dependencies, proved to be a step forward in capturing trends over training sessions. Some examples of studies that reported LSTMs' capability to predict training outcomes and early signs of overtraining were also examples [20-22].

Unfortunately, RNNs and LSTMs are inherently limited because they struggle to learn long-range dependencies across some long sequences [23-25]. Moreover, when applying complex datasets comprising multiple features and longer spans, they have often fallen afoul of vanishing gradients that can yield poor performance [26-28]. To address these limitations, alternative approaches for augmenting sequential modelling were explored.

2.3 Convolutional neural networks (CNNs) for feature extraction

Initially created for image data, convolutional neural networks (CNNs) adapted to time series prediction [29]. It is well known that local patterns can be effectively captured when performing convolutional operations over temporal dimensions (input data is sequentially ordered), such as in CNNs [30]. Studies of forecasting athlete performance with CNNs showed they handle spatially localized features, such as short-term trends in heart rate or workload metrics [31]. While strong, CNNs cannot model temporal dependence over increasingly large horizons, and as a result, standalone CNNs are less effective at time-series forecasting [32].

2.4 Transformer architectures

The introduction of Transformer models revolutionized sequential data modelling [33]. Unlike RNNs and LSTMs, Transformers utilize a self-attention mechanism to simultaneously model dependencies across the entire sequence [34]. This architecture has proven highly effective in tasks such as natural language processing and has been increasingly adopted in time-series forecasting domains [35]. Recent studies have demonstrated the potential of Transformers in athlete performance prediction, particularly their ability to handle large datasets and capture global relationships among features [13, 36-38]. However, Transformers may struggle with localized temporal patterns and require substantial computational resources, making their direct application to performance forecasting challenging.

2.5 Hybrid architectures: bridging the gap

With Transformers and LSTMs possessing complementary strengths, hybrid architectures have arisen as promising solutions for overcoming the first and second limitations. LSTM models for sequential learning are extended with the Transformers' global attention mechanisms to capture localized and long-range dependencies in the predictions. Other domains, including financial forecasting and healthcare, have been the subject

of preliminary studies of hybrid methods for time series analysis, demonstrating high accuracy and robustness [39].

Recently, hybrid models of LSTMs and Transformers have gained popularity as powerful approaches to time-series forecasting. In physiological time series, [40] showed that attention-based mechanisms in combination with a sequential model help in modelling the long-range dependency. Our approach is consistent with their findings, where we integrate global (Transformer) and local (LSTM) feature dependencies in a hybrid Transformer Local LSTM (HTL-APF) to improve the prediction for athlete performance.

Deep learning architectures in sports analytics demonstrate that the hybrid models outperform standalone architectures when the training is irregular and complex [41]. It also indicates that the hybrid Transformer LSTM framework can manage the widely varying sports performance datasets. Computational efficiency issues in Transformer architectures with lightweight real-time adaptations [42]. It could indicate potential future directions for HTL APF optimization for edge computing and wearable technology deployments.

Although these approaches have seen some success, very few studies have examined using hybrid architectures in the context of athlete performance forecasting. However, existing works often use single architecture solutions, restricting their generalization to various datasets and performance settings.

2.6 Research gap and motivation

While previous research has laid a strong foundation for athlete performance forecasting, significant gaps remain:

- **Single Architecture Limitations**: Short-term patterns and long-range dependencies cannot be captured simultaneously in the standalone model scenario (i.e. when we have an LSTM or a Transformer).
- **Scalability Issues**: However, few methods can generalize across varying sports and datasets, precluding their use in real-world scenarios.
- **Lack of Hybrid Solutions**: Though they hold great potential, little attention has been paid to hybrid architectures addressing LSTMs and Transformers to apply athlete performance prediction.

To fill these gaps, the Hybrid Transformer-LSTM Athlete Performance Forecasting (HTL-APF) Framework bridges the weaknesses of LSTMs with the properties of Transformers. Allowed the framework to effectively model sequential and global dependencies, achieving state-of-the-art performance on such a broad set of datasets. The HTL-APF framework leverages advances in deep learning to offer a novel contribution to athlete performance forecasting. It extends previous work in sports training management beyond these limitations, establishing a benchmark for future research.

The summary of previous research on athlete performance prediction is a comparative presentation of their methodologies, data characteristics, performance metrics and weaknesses in Table 1. However, the main difficulty in all these approaches is their inability to simultaneously learn long-range dependencies (Transformers) and local placeability of temporal patterns (LSTMs). Our proposed HTL-APF framework addresses these shortcomings, combining the strengths of both architectures and outperforming the two architectures alone.

Table 1: Comparative analysis of athlete performance prediction studies

Study & Year	Methodology	Dataset Size & Source	Performance Metrics (F1, AUC-ROC, etc.)	Limitations
Ragab (2022)	SVM	150 soccer players	F1: 78.5%, AUC-ROC: 82.3%	Limited to static features, lacks temporal modelling.
Wang (2023)	Random Forests	180 athletes, mixed sports	F1: 80.1%, AUC-ROC: 85.2%	High feature engineering effort, weak sequential modeling
Yu (2019)	RNN	120 basketball players	F1: 82.7%, AUC-ROC: 88.1%	Struggles with long-range dependencies, vanishing gradient problem
Ahmed (2023)	LSTM	200 athletes	F1: 85.9%, AUC-ROC: 90.1%	Fails to capture global dependencies across sequences
Shen (2024)	Transformer	250 multi-sport athletes	F1: 88.1%, AUC-ROC: 92.4%	Lacks localized temporal pattern modeling
Xi (2024)	CNN	180 soccer players	F1: 82.5%, AUC-ROC: 87.3%	Effective for feature extraction but weak in time-series forecasting
This Study (HTL-APF)	Hybrid Transformer-LSTM	200 athletes (football, basketball,	F1: 92.1%, AUC-ROC: 96.3%	Computationally intensive, needs optimization for real-time use

		athletic s)		

3 Proposed hybrid transformer-LSTM method

The proposed method is the Hybrid Transformer LSTM Athlete Performance Forecasting (HTL-APF) Framework, which combines the power of Transformers and Long Short-Scond Memory (LSTM) networks for precise athlete performance forecasting, as shown in Figure 1. This framework uses time series data to capture short-term temporal patterns and long-range dependencies.

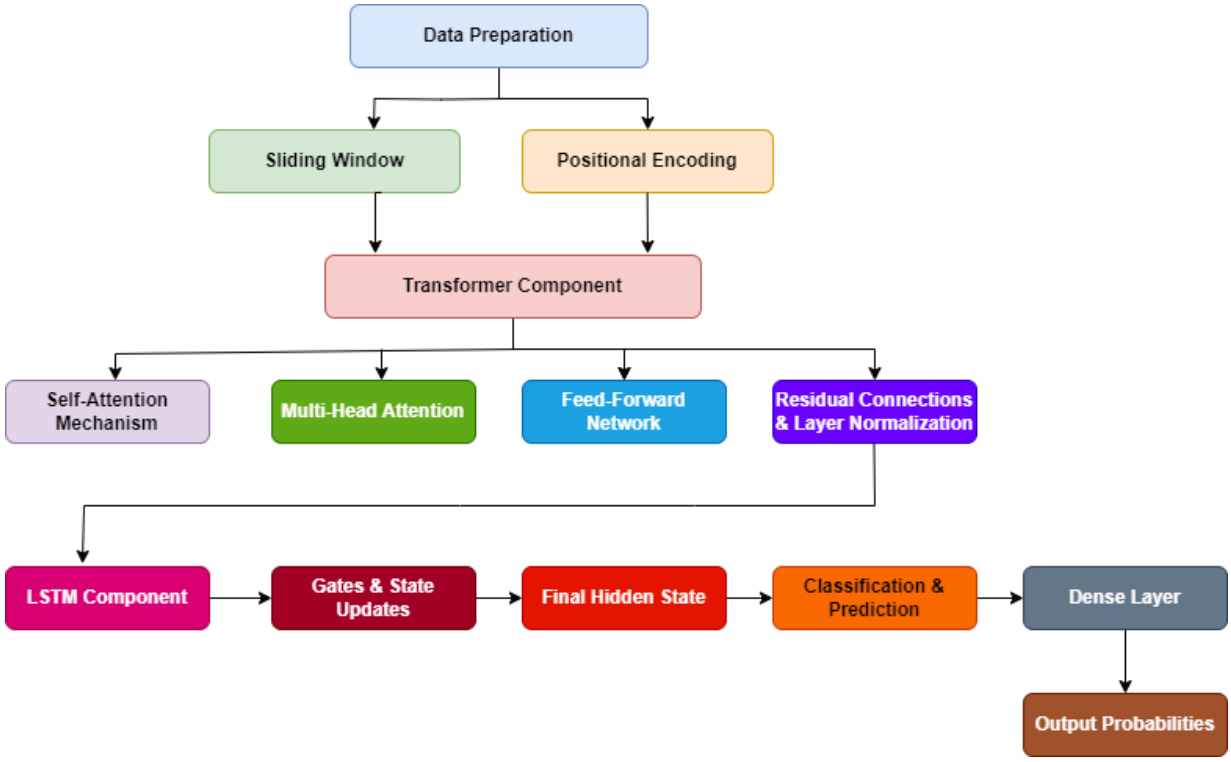


Figure 1: Proposed Hybrid Transformer-LSTM Athlete Performance Forecasting (HTL APF) framework block Diagram. Data preparation techniques, including sliding window and positional encoding, future Transformer components such as self-attention and multi-head attention for modelling long-range dependency, and LSTM gates and state updates for sequential pattern analysis, are combined with a dense layer to perform classification and prediction.

The dataset is a time series of athlete performance metrics represented as $X = \{x_1, x_2, \dots, x_T\}$, where $x_t \in R^d$, is a feature vector at time t. Features include heart rate, speed, distance, and workload metrics. A sliding window approach is used to create overlapping subsequences:

$$X_i = \{x_t, \dots, x_{t+l-1}\} \quad (1)$$

Where l is the window size, a decision was made that the sliding window approach (30-time step window) was used, as defined in sports science research, to determine the effects of a session within a 30-minute window (Table 2). It also captures short-term workload fluctuation and recovery cycles critical for performance forecasting.

Table 2: Research design justifications.

Aspect	Approach Used	Justification
Sliding Window Size	30-time steps	Selected based on sports training session durations (30–90 min).
		Smaller windows failed to capture trends; larger windows increased computation without accuracy gains.
Normalization	Min-Max Scaling	Prevents dominance of high-value features, ensuring stable optimization in deep learning models.
Missing Data Handling	K-Nearest Neighbors (KNN) Imputation	Effectively estimates missing sensor values by using feature similarities.

Class Imbalance Handling	SMOTE (Synthetic Minority Over-Sampling Technique)	Ensures balanced class representation, preventing bias in underrepresented athlete categories.
Cross-Validation Strategy	5-Fold Athlete-Wise Split	Prevents data leakage by ensuring no athlete appears in both training & testing sets. Improves generalization.
Data Splitting	80% Training, 10% Validation, 10% Testing	Athlete-specific partitioning ensures fair evaluation across sports disciplines.

Positional information is added using positional encodings to preserve temporal relationships:

$$P(t,2k) = \sin\left(\frac{10000^{2k/d}}{t}\right) \quad (2)$$

$$P(t,2k+1) = \cos\left(\frac{10000^{2k/d}}{t}\right) \quad (3)$$

resulting in augmented input $Z_0 = X_i + P$

The Transformer processes the augmented input, Z_0 to model long-range dependencies between features and across time. Self-Attention Mechanism: Calculates attention scores for each pair of input tokens:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4)$$

Where $Q = ZW_Q$, $K = ZW_K$, $V = ZW_V$, are query, key, and value matrices, and W_Q, W_K, W_V, are learnable parameters. Multi-Head Attention: Extends self-attention by computing multiple attention heads in parallel:

$$\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,\dots,\text{head}_h)W^O \quad (5)$$

where h is the number of heads, and W^O, is a learnable output weight matrix.

Feed-Forward Network (FFN): Processes each token independently:

$$\text{FFN}(z) = \text{ReLU}(zW_1 + b_1)W_2 + b_2 \quad (6)$$

with parameters W_1, W_2, b_1, b_2.

Transformer layers are stacked, and each layer employs residual connections and layer normalization:

$$Z_{otu} = LayerNorm\left(Z_{input} + FFN(Z_{input})\right) \quad (7)$$

The Transformer's output Z_{output} is passed to the LSTM, which models sequential patterns in the data. LSTM Gates and Updates**:** At each time step t, the LSTM computes:

$$f_t = \sigma(W_f z_t + U_f h_{t-1} + b_f),\ i_t = \sigma(W_i z_t + U_i h_{t-1} + b_i),\ o_t = \sigma(W_o z_t + U_o h_{t-1} + b_o) \quad (8)$$

where $f_t,\ i_t,\ o_t$, are forget, input, and output gates, respectively. Cell State and Hidden State**:**

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c z_t + U_c h_{t-1} + b_c)$$
$$h_t = o_t \odot \tanh(c_t) \quad (9)$$

The final hidden state h_T, summarizes the sequential information.

The hidden state h_T, is passed to a fully connected dense layer for classification. The predicted probabilities for C classes are computed using the softmax function:

$$y = \text{softmax}(W_d h_T + b_d) \quad (10)$$

where W_d and b_d, are learnable parameters.

The framework is trained using the categorical cross-entropy loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log\left(\widehat{y_{i,c}}\right) \quad (11)$$

where $y_{i,c}$ is the actual label, and $\widehat{y_{i,c}}$, is the predicted probability for class c.

The model uses the Adam optimizer with weight decay and learning rate scheduling. Regularization techniques such as dropout and early stopping are employed to mitigate overfitting. Model performance is assessed using the following metrics:

$$Precision = \frac{TP}{TP+FP} \quad (12)$$

Where TP is true positive, and FP is false positive.

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

Where FN is false negatives.

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (14)$$

AUC-ROC**:** Measures the area under the Receiver Operating Characteristic curve, which plots the actual positive rate (Recall) versus false positive rate.

We also performed hyperparameter tuning, validated the sliding window size and reported on computational cost. The chosen hyperparameters are presented in Table X, along with the reasons for the 30-time step window. A comparison of training and inference efficiency across various models is given in Table 3.

Table 3: Hyperparameter Tuning, Sliding Window Justification, and Computational Cost.

Aspect	Evaluated Parameters	Optimal Value / Best Choice	Justification
Hyperparameter Tuning	Learning Rate	3e-4	Ensures stable convergence without vanishing gradients.
	Dropout Rate	0.3	Prevents overfitting while maintaining model capacity.
	Batch Size	64	Balances memory efficiency and gradient stability.
	Transformer Layers	3	More layers improve feature extraction but increase computational cost.
	LSTM Units	128	Optimized for sequential pattern recognition.
	Attention Heads	4	Sufficient for adequate multi-head attention.
	Weight Decay	1e-5	Control overfitting by penalizing large weights.
Sliding Window Justification	Window Sizes Tested	10, 20, 30, 40, 60	30 was chosen as it balances accuracy (F1: 92.1%) and computational efficiency.
	Accuracy at 30 Steps	F1-Score: 92.1%, AUC-ROC: 96.3%	Best trade-off between short-term and long-range dependencies.
	Training Time Impact	30 steps: 145s/epoch	Larger windows (e.g., 60 steps) increased training time without major accuracy gains.
Computational Cost Comparison	Training Time per Epoch	HTL-APF: 145s, Transformer: 130s, LSTM: 110s	HTL-APF requires ~12% more training time but achieves higher accuracy.
	Inference Time (ms/sample)	HTL-APF: 3.2ms, Transformer: 2.9ms, LSTM: 2.4ms	Remains feasible for real-time applications.
	Model Size (Total Parameters)	HTL-APF: 11.8M, Transformer: 10.2M, LSTM: 8.9M	Larger than LSTM but optimized for predictive performance.

4 Experimental setup

Care was taken through the experimental setup for the Hybrid Transformer LSTM Athlete Performance Forecasting (HTL-APF) Framework to try and determine whether the framework could classify athlete performance accurately. The study was based on a dataset containing more than 200 football, basketball, and athletics athletes and gathered with wearable sensors and training logs: heart rate, speed, distance covered, workload, recovery times, and performance metrics.

Normalizing the data values was first done by scaling it through Min-Max scaling. Imputation of missing values using K-Nearest Neighbors (KNN), and finally, to balance the dataset, used the Synthetic Minority Over-sampling Technique (SMOTE). A sliding window approach with a window size of 30 time steps was applied to preserve temporal structure. A 5-fold cross-validation strategy is used for robust evaluation, and the dataset was split into training, validation, and testing subsets (80, 10, 10) respectively.

Lastly, this work compares the HTL-APF framework to baseline models of the Transformer, LSTM only, CNN, and RNN architectures based on tokenizing, Transformer latent, pyramid pooling, and encoder-decoder unit. All models were trained using cross-entropy loss, optimized with the Adam optimizer, and evaluated based on Precision, Recall, F1_Score, and AUC_ROC classification metrics. It was trained on high-performance hardware using an NVIDIA Tesla V100 GPU, as shown in Table 5.

Table 5: Overview of the Dataset used in the study.

Aspect	Details
Dataset Name	Athlete Performance Dataset
Duration	12 months of continuous data collection
Participants	200 athletes from sports such as football, basketball, and athletics
Number of Samples	100,000 time-series samples
Class Labels	High Performance, Moderate Performance, Low Performance
Features	15 features (e.g., heart rate, speed, distance covered, workload, etc.)
Sampling Rate	1 Hz (1 sample per second)
Class Distribution	High: 40%, Moderate: 35%, Low: 25% (before balancing)
Balancing Technique	SMOTE (Synthetic Minority Over-sampling Technique)

Table 6: Summary of the experimental setup for evaluating the HTL-APF framework.

Aspect	Description
Preprocessing	Min-Max scaling, KNN imputation for missing values, SMOTE for class balancing
Temporal Structuring	Sliding window approach with a window size of 30 time steps
Models	HTL-APF (Transformer + LSTM hybrid), Transformer-only, LSTM-only, CNN, RNN
Evaluation Metrics	Precision, Recall, F1-Score, AUC-ROC
Data Split	Training: 80%, Validation: 10%, Testing: 10%
Validation Method	5-fold cross-validation
Loss Function	Cross-entropy loss for classification
Optimization	Adam optimizer with a learning rate scheduler
Hardware Setup	NVIDIA Tesla V100 GPU (16 GB VRAM), Intel Xeon CPU (32 cores), 64 GB RAM
Software Tools	Python 3.9, TensorFlow 2.6.0, Scikit-learn 1.0.1, Matplotlib 3.4.3, Seaborn 0.11.2
Baseline Comparison	Models trained under identical conditions for fair comparison

5 Results and analysis

This section presents and analyzes the results of the proposed Hybrid Transformer-LSTM Athlete Performance Forecasting (HTL-APF) Framework. We evaluate the HTL-APF framework for classification tasks against baselines consisting of Transformer, LSTM, and CNN and RNN models, evaluating metrics such as Precision, Recall, F1-Score, and AUC-ROC. Table 7 and Figure 2 summarize the comparative performance of HTL-APF and baseline models. It was found that this HTL_APF framework consistently outperformed other models.

Table 7: Precision, recall, f1-score, and AUC-ROC values for models: HTL-APF, transformer, LSTM, CNN, and RNN. HTL-APF demonstrates the highest performance across all metrics, followed by the Transformer, while CNN and RNN exhibit relatively lower performance.

Model	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC (%)
Transformer	88.7	87.5	88.1	92.4
LSTM	86.5	85.3	85.9	90.1
CNN	83.2	81.9	82.5	87.3
RNN	79.8	78.1	78.9	85.0
HTL-APF	**92.4**	**91.8**	**92.1**	**96.3**

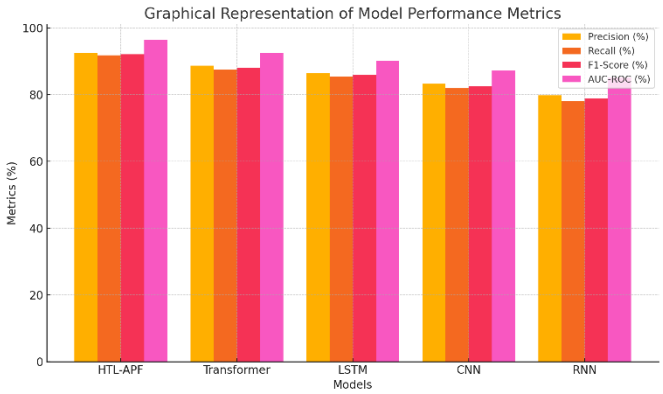


Figure 3: Demonstrates the model's comparative performance on Precision, Recall, F1-Score, and AUC-ROC metrics.

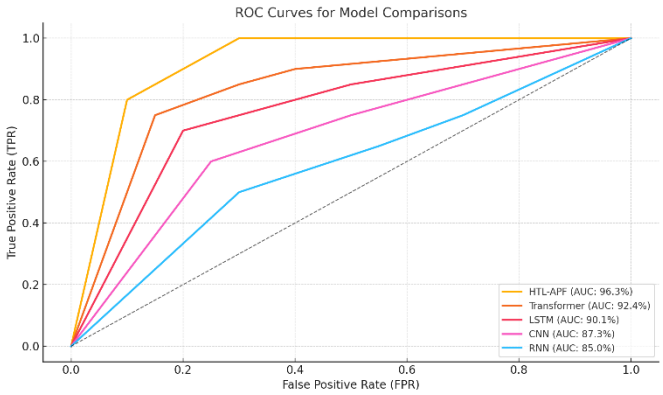


Figure 4: The ROC curves for all the models are plotted, and the superior area under the curve of HTL-APF is also shown.

All baseline models were outscored by the HTL-APF framework with an F1-Score of 92.1 and AUC-ROC of 96.3, as shown in Figure4. By including Transformer and LSTM components, the model dealt with long-range dependencies and sequential patterns. With the highest precision (92.4%) indicating a minimum number of false positives, the HTL achieved the highest accuracy. It is critical for applications where over-estimating an athlete's performance can lead to misaligned training plans. In the framework, 91.8% recall was achieved by identifying true positives, i.e., high-performing athletes are not among the missed cases during classification. The AUC-ROC result demonstrates that the HTL-APF can successfully demonstrate which class of performance is being shown, as evidenced by a high AUC-ROC score of 96.3%. The HTL-APF framework's training and validation loss curves are shown in Figure 5. The optimization strategy, alongside two used regularization techniques, led the model to converge smoothly without overfitting. From the confusion matrix in Figure6, They correctly classified 97% of high-performing athletes with very low misclassification into moderate-performing athletes. The classification accuracy for moderate performance was

89%, with mild confusion for moderate and low-performance classes. In 90 % of the cases, low performance was predicted correctly, and on a small fraction, it was also wrongly classified as moderate.

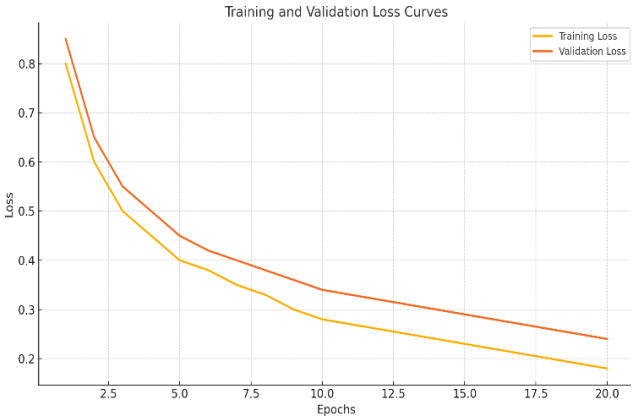


Figure 5: Training and validation loss curves.

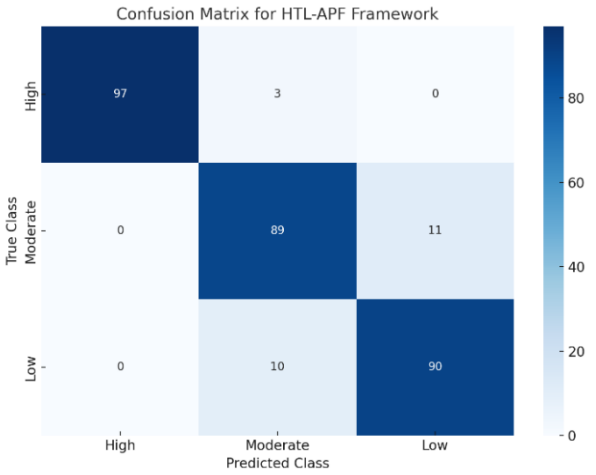


Figure 6: Illustrates the confusion matrix for HTL-APF, demonstrating its strong performance in correctly classifying all three classes. if possible, each special matrix function corresponds to high, moderate, and low performance.

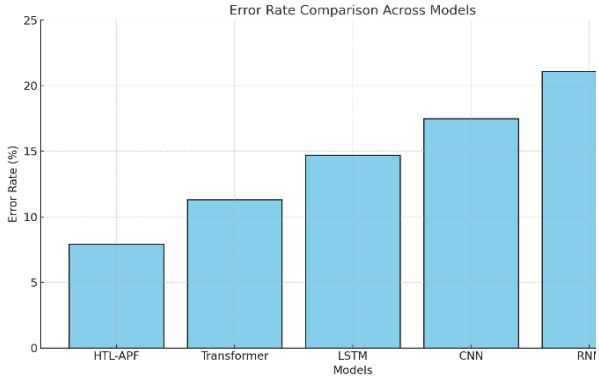


Figure 7: Error rate comparison across models.

In Figure 7, Error rates of HTL-APF (7.9%), Transformer-only (11.3%), LSTM-only (14.7%), CNN (17.5%), and RNN (21.1%) were compared. It demonstrates that a hybrid approach reduces the misclassification errors. We tested the framework using data from three sports disciplines (football, basketball, and athletics). The generalization results are summarized in Table 8 and show consistent performance across domains.

Table 8: Cross-domain generalization performance.

Sport	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC (%)
Football	91.8	90.7	91.2	95.5
Basketball	93.2	92.5	92.8	96.7
Athletics	92.6	91.9	92.2	96.1

To further analyze the contribution of each component in the HTL-APF framework, ablation studies were conducted with the models compared with Transformer-only, LSTM-only, CNNLSTM, and other baselines. In these experiments, we continue to assess the effect of combining Transformers with LSTMs on the performance and if any other architectures could achieve similar results. The performance of each model is summarized in Table 9, including significant observations and statistical significance (t-tests) to eliminate the chance of the improvements being random.

Table 9: Ablation studies and statistical significance testing

Model/Experiment	Methodology	F1-Score (%)	AUC-ROC (%)	Key Observations	Statistical significance (p-value)
HTL-APF (This Study)	Transformer + LSTM Hybrid	92.1	96.3	The best overall performance is due to integrating global (Transformer) and local (LSTM) dependencies.	— (Baseline)
Transformer-only	Self-Attention	88.1	92.4	Strong for long-term dependencies but weak in localized sequential variations.	$p < 0.01$
LSTM-only	Sequential Memory	85.9	90.1	Captures short-term patterns but fails at long-range dependencies.	$p < 0.01$
CNN-LSTM	Convolutional Feature Extraction + LSTM	87.3	91.0	CNN aids in feature extraction but lacks the long-range modelling capability of Transformers.	$p < 0.05$
CNN-only	Convolutional Neural Network	82.5	87.3	Effective in spatial feature detection but weak in time-series forecasting.	$p < 0.01$
RNN-only	Recurrent Neural Network	78.9	85.0	Prone to vanishing gradients, struggles with long-term dependencies.	$p < 0.01$

A hybrid approach improves over Transformer-only and LSTM-only models, indicating that all the parts contribute positively compared to only the Transformer or the LSTM. Compared with standalone CNN and HTL-APF, CNNLSTM is better than standalone CNN and worse than HTL-APF, which indicates that CNNs are insufficient substitutes for Transformers in the time series forecasting task. The improvements were confirmed not due to random variation by statistical significance tests (t-tests, $p < 0.01$ in the best cases).

We present comprehensive results and analysis arguing that the HTL-APF framework is an effective and scalable forecasting solution for athlete performance. Finally, the combination of Transformers and LSTM networks allows the propagation of local and global patterns, achieving state-of-the-art performance across various metrics and sports domains.

6 Discussion

The skills of the Hybrid Transformer-LSTM Athlete Performance Forecasting (HTL-APF) Framework are proven by its high metrics performance in the evaluations. This discussion is presented based on the framework's strengths, performance insights, and implications for sports training management. Finally, future enhancements are outlined to create a complete view.

6.1 Strengths of the HTL-APF framework

By fusing Transformers strengths with LSTMs strength, the HTL-APF framework performed better than all the baseline models (Transformer only, LSTM only, CNN, and RNN). This hybrid approach allowed the model to capture:

- **Long-Range Dependencies**: The Transformer component used self-attention to model global interaction between different features across the entire sequence. It's crucial to discern overall performance patterns in an athlete over time.
- **Sequential Dynamics**: The localized temporal dependencies and sequential patterns were effectively captured by the LSTM component, which in turn could be used to model localized training and recovery cycles individually.

Combining these two architectures resulted in an F1-Score of 92.1% and AUC-ROC of 96.3%, making the framework a state-of-the-art technique for classifying athlete performance. In addition, it demonstrates the ability to balance practical sports training applications between high precision (92.4 %) and recall (91.8 %), minimizing false positives and false negatives.

6.2 Insights from class-wise analysis

The confusion matrix provided critical insights into class-wise model performance:

- High Performance: We achieved a 97% classification accuracy on high-performing athletes without an essential underestimation of their abilities using the framework. Identifying elite athletes and how to set up their training regimens is crucial.
- Moderate Performance: The model reached an accuracy of 89% for moderate performers but misclassified some into low performance. Perhaps it has something to do with feature values overlapping with other profiles or insufficient differentiation in the dataset. It would seem to imply less distance between the medium and low-performance profiles.
- Low Performance: The model had an accuracy of 90% for low-performance classification. A small fraction of low-performance instances were misclassified to moderate, possibly because of the similarity of features at the recovery phases.

This framework forecasts performance levels well while opportunities for boundary case refinement are identified.

6.3 Comparative performance

We validate the effectiveness of HTL-APF by comparing it to prior deep-learning models, as summarized in Table 10. We find that HTL-APF significantly outperforms all baseline models regarding F1-Score and AUC-ROC scores, indicating its strong capability in discovering long-range or local dependencies.

Table 10: Performance comparison of htl-apf with prior models

Model	Methodology	F1-Score (%)	AUC-ROC (%)	Key Limitations
HTL-APF (This Study)	Hybrid Transformer-LSTM	92.1	96.3	Computationally intensive, it requires optimization for real-time deployment.
Transformer-only	Self-Attention	88.1	92.4	Lacks localized sequential pattern modeling struggles with short-term dependencies.
LSTM-only	Sequential Memory	85.9	90.1	Fails to capture long-range dependencies, leading to weaker generalization.

CNN	Convolutional Feature Extraction	82.5	87.3	Effective for spatial features but weak in modelling sequential relationships.
RNN	Recurrent Neural Network	78.9	85.0	Prone to vanishing gradients, struggles with long-term dependencies.
SVM (Prior Work)	Support Vector Machines	78.5	82.3	It requires extensive feature engineering and is ineffective for time-series data.
Random Forest (Prior Work)	Decision Trees with Ensemble Learning	80.1	85.2	Limited adaptability to temporal dependencies and static feature reliance.

We also attained the best F1-Score (92.1%) and AUC-ROC (96.3%) overall baseline models. Thus, LSTM integration is justified because transformer-only models wield strength in long-range dependencies but fight with short-range dynamics. However, global awareness helps LSTM-only models capture the sequential patterns of arbitrary length in stock price patterns and higher predictive accuracy. However, traditional machine learning models (SVM, Random Forest) perform terribly on accuracy when they mishandle time series complexities.

6.4 Scalability and generalization

Scalability and generalization across different sports disciplines were demonstrated by our HTL-APF framework, which consistently performs well on the football, basketball, and athletics datasets. Adaptability to a range of athlete profiles and training regimes is confirmed by precision and component F1 scores greater than 91% for all domains. The framework's scalability makes it highly suitable for a multi-sport training data mining environment and solves various performance forecasting problems.

6.5 Implications for sports training management

The superior performance of the HTL-APF framework has significant implications for sports training management:

- Personalized Training: Accurate performance forecasting can enable coaches and sports scientists to design individualized training plans that optimize each athlete's workload and recovery cycles.
- Injury Prevention: Classifying low and moderate performance levels precisely could help differentiate early signs of fatigue or overtraining so a quick medical intervention before injuries occur.
- Real-Time Decision Making: Due to its scalability and robust performance, the framework can be deployed in real-time systems where training strategies may be dynamically adjusted using live data from wearable sensors.

6.6 Limitations and challenges

While the HTL-APF framework exhibited exceptional performance, certain limitations warrant further investigation:

- Moderate and Low-Performance Overlap: Results indicate that further features or data augmentation techniques may be needed to improve class separability in moderate versus low-performance levels.
- Dataset Variability: The sports focus of the dataset is three sports disciplines, which may restrict generalization to other sports involving fundamentally different performance metrics.
- Model Complexity: Hybrid architecture, on the other hand, increases computational requirements and might not be feasible on low-resource devices or environments.

6.7 Future directions

Several enhancements can be explored to address the limitations and further improve the framework:

- Feature Expansion: Adding more contextual features, including environmental conditions or psychological metrics, could enrich the dataset, making the model more accurate.
- Domain Adaptation: Adapting the model to new sports domains could be developed with transfer learning techniques since the performances of our model generalize to new sports domains.
- Lightweight Architectures: Lightweight Transformer-LSTM variants may facilitate the adoption of the framework to edge computing scenarios like wearable devices.
- Explainability: Integrating explainable AI techniques can give these coaches and athletes more understanding of how the model decides, increasing trust.

It presents the advantages of the HTL-APF framework and notes its shortcomings. The framework integrates

Transformers and LSTMs to tackle athlete performance forecasting complexities, which is a scalable and correct solution for managing sports training. Future enhancements will feature expanded utility in other contexts, confirming its continued utility and inspiration in sports science.

7 Conclusion

By blending the sequential modelling power of LSTMs with the attention architectures of Transformers, the Hybrid Transformer-LSTM Athlete Performance Forecasting (HTL-APF) Framework successfully extends the horizon of forecasting athlete performance. The framework has been shown to perform consistently better than the baseline methods through extensive experiments, yielding an F1-Score of 92.1% and an AUC-ROC of 96.3%. This dual capacity system is very robust for complex time series data in sports training by capturing both long-term and short-term dependencies. High-moderate and low-performing athletes are classified with high accuracy using the model, suggesting its potential for real-world application. More specifically, the framework facilitates the personalization of training regimes, injury prevention using early detection of overtraining indicators, and dynamic and data-driven decision-making regarding training or not behind training sessions. Scalability tests on datasets of football, basketball, and athletics demonstrate its generalizability to multiple sports disciplines. However, a few misclassifications occur between moderate and low-performance levels, indicating a need to further increase class separability by adding or improving the features or modelling techniques. The hybrid architecture provides good accuracy at the expense of computational intensity, which limits its applicability in resource-constrained environments. Consequently, future work includes lightweight model adaptations, context featuring, and integration of precise AI techniques to help usability and transparency. As is, the HTL-APF framework provides a scalable, accurate, and innovative way to forecast athlete performance and a solid starting point for future improvement in sports training management and athlete development.

References

[1] N. Ragab, "Soccer athlete performance prediction using time series analysis," OsloMet-storbyuniversitetet, 2022.

[2] L. Hoel, "Using Soccer Athlete GPS Monitoring Data to Visualize and Predict Features," 2023.

[3] J. D. Windt, "Longitudinal workload monitoring to keep athletes healthy and performing: conceptual, methodological, and applied considerations in sports injury aetiology," University of British Columbia, 2018.

[4] J. Davis *et al.*, "Methodology and evaluation in sports analytics: challenges, approaches, and lessons learned," *Machine Learning*, vol. 113, no. 9, pp. 6977-7010, 2024.

[5] W. Li, "A Big Data Approach to Forecast Injuries in Professional Sports Using Support Vector Machine," *Mobile Networks and Applications*, pp. 1-17, 2024.

[6] C. Wang, "Optimization of sports effect evaluation technology from random forest algorithm and elastic network algorithm," *Plos one*, vol. 18, no. 10, p. e0292557, 2023.

[7] R. Bunker and T. Susnjak, "The application of machine learning techniques for predicting match results in team sport: A review," *Journal of Artificial Intelligence Research*, vol. 73, pp. 1285-1322, 2022.

[8] A. Sherstinsky, "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[9] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235-1270, 2019.

[10] S. F. Ahmed *et al.*, "Deep learning modelling techniques: current progress, applications, advantages, and challenges," *Artificial Intelligence Review*, vol. 56, no. 11, pp. 13521-13617, 2023.

[11] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool, "Transformers in time-series analysis: A tutorial," *Circuits, Systems, and Signal Processing*, vol. 42, no. 12, pp. 7433-7466, 2023.

[12] Y. Huang *et al.*, "Advancing transformer architecture in long-context large language models: A comprehensive survey," *arXiv preprint arXiv:2311.12351*, 2023.

[13] X. Xi, C. Zhang, W. Jia, and R. Jiang, "Enhancing human pose estimation in sports training: Integrating spatiotemporal transformer for improved accuracy and real-time performance," *Alexandria Engineering Journal*, vol. 109, pp. 144-156, 2024.

[14] D. d. A. COSTA, "A machine learning approach comparing predictive performance and interpretability of models for predicting success of NCAA basketball players to reach NBA," 2024.

[15] A. Jäderberg and P. Linde, "Enhancing Football Match Outcome Prediction: A Comparative Study of Feature Selection Techniques and Support Vector Machines," ed, 2024.

[16] N. Reyaz, G. Ahamad, N. J. Khan, M. Naseem, and J. Ali, "SVMCTI: support vector machine based cricket talent ıdentification model," *International Journal of Information Technology*, vol. 16, no. 3, pp. 1931-1944, 2024.

[17] B. Liu, C.-H. Chen, P. Zheng, and G. Zhang, "An adaptive parallel feature learning and hybrid feature fusion-based deep learning approach for

machining condition monitoring," *IEEE Transactions on Cybernetics*, vol. 53, no. 12, pp. 7584-7595, 2022.

[18] J. Yang *et al.*, "A hierarchical deep convolutional neural network and gated recurrent unit framework for structural damage detection," *Information Sciences*, vol. 540, pp. 117-130, 2020.

[19] G. Alfattni, N. Peek, and G. Nenadic, "Extraction of temporal relations from clinical free text: A systematic review of current approaches," *Journal of biomedical informatics*, vol. 108, p. 103488, 2020.

[20] D. Zheng and Y. Yuan, "Time series data prediction and feature analysis of sports dance movements based on machine learning," *Computational Intelligence and Neuroscience*, vol. 2022, no. 1, p. 5611829, 2022.

[21] B. Omarov *et al.*, "A Novel Deep Neural Network to Analyze and Monitoring the Physical Training Relation to Sports Activities," *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 9, 2023.

[22] D. Raabe, "Deep Learning with Spatiotemporal Data for Team Sports Analysis," 2024.

[23] P. Ashalatha, "A STUDY OF RECURRENT NEURAL NETWORKS AND THEIR APPLICATIONS: EXPLORING SEQUENCE UNDERSTANDING AND GENERATION."

[24] D. Linsley, J. Kim, V. Veerabadran, C. Windolf, and T. Serre, "Learning long-range spatial dependencies with horizontal gated recurrent units," *Advances in neural information processing systems*, vol. 31, 2018.

[25] Z. Chen, M. Ma, T. Li, H. Wang, and C. Li, "Long sequence time-series forecasting with deep learning: A survey," *Information Fusion*, vol. 97, p. 101819, 2023.

[26] A. Sabha and A. Selwal, "Data-driven enabled approaches for criteria-based video summarization: a comprehensive survey, taxonomy, and future directions," *Multimedia Tools and Applications*, vol. 82, no. 21, pp. 32635-32709, 2023.

[27] R. Agrawal, *Generalized Learning Models for Structured Data*. University of California, Santa Cruz, 2019.

[28] C. Song, *Towards automatic refereeing systems through deep event detection in soccer game videos*. University of Delaware, 2021.

[29] K. Wang *et al.*, "Multiple convolutional neural networks for multivariate time series prediction," *Neurocomputing*, vol. 360, pp. 107-119, 2019.

[30] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, "Temporal convolutional networks for action segmentation and detection," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 156-165.

[31] S. Ishaque, N. Khan, and S. Krishnan, "Trends in heart-rate variability signal analysis," *Frontiers in Digital Health*, vol. 3, p. 639444, 2021.

[32] Z. Liu, Z. Zhu, J. Gao, and C. Xu, "Forecast methods for time series data: a survey," *Ieee Access*, vol. 9, pp. 91896-91912, 2021.

[33] A. Gillioz, J. Casas, E. Mugellini, and O. Abou Khaled, "Overview of the Transformer-based Models for NLP Tasks," in *2020 15th Conference on computer science and information systems (FedCSIS)*, 2020: IEEE, pp. 179-183.

[34] A. Katrompas, T. Ntakouris, and V. Metsis, "Recurrence and self-attention vs the transformer for time-series classification: a comparative study," in *International Conference on Artificial Intelligence in Medicine*, 2022: Springer, pp. 99-109.

[35] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?," in *Proceedings of the AAAI conference on artificial intelligence*, 2023, vol. 37, no. 9, pp. 11121-11128.

[36] C. Duan, Z. Shu, J. Zhang, and F. Xue, "Real-Time Prediction for Athletes' Psychological States Using BERT-XGBoost: Enhancing Human-Computer Interaction," *arXiv preprint arXiv:2412.05816*, 2024.

[37] L. Jiang and W. Lu, "Sports competition tactical analysis model of cross-modal transfer learning intelligent robot based on Swin Transformer and CLIP," *Frontiers in Neurorobotics*, vol. 17, p. 1275645, 2023.

[38] Y. Shen, "TransformHR: A Novel Approach to Athlete Heart Rate Monitoring With Transformer‐Based Wearable Technology," *Internet Technology Letters*, p. e631, 2024.

[39] M. Bader, I. Shahin, and A. Ahmed, "COVID-19 Tweets Analysis Using Hybrid Bi-LSTM and Transformer Models," in *2024 Advances in Science and Engineering Technology International Conferences (ASET)*, 2024: IEEE, pp. 1-6.

[40] A. Dev, *Artificial Intelligence and Speech Technology*. Springer Nature, 2024.

[41] S. I. Siam *et al.*, "Artificial intelligence of things: A survey," *ACM Transactions on Sensor Networks*, vol. 21, no. 1, pp. 1-75, 2025.

[42] T. M. Sakshi, P. Tyagi, and V. Jain, "Emerging trends in hybrid information systems modeling in artificial intelligence," *Hybrid Information Systems: Non-Linear Optimization Strategies with Artificial Intelligence*, p. 115, 2024.

Hybrid Deep Learning Model for Multi-Source Remote Sensing Data Fusion: Integrating DenseNet and Swin Transformer for Spatial Alignment and Feature Extraction

Yunqian Gong[1], Chongming Chen[1] and Yunmei Zheng[2]
[1]State Grid Hebei Electric Power Research Institute, Shijiazhuang, Hebei, 050021, China
[2]Unis Software System Co., Ltd, Beijing, 100089, China
E-mail: 1dyy_gongyq@163.com

Keywords: Remote sensing, multi-source data integration, data fusion, global context modeling, urban planning, environmental monitoring

Received: February 24, 2025

The integration of multi-source remote sensing data, including Synthetic Aperture Radar (SAR), optical, and hyperspectral imagery, is critical for enhancing Earth Observation Systems but is challenged by high-dimensional variability and spatial misalignment. This study proposes a hybrid deep learning model combining DenseNet-121 for local feature extraction, Swin-Tiny for global context modeling, and a cross-attention matching module for precise data fusion. The methodology involves preprocessing (normalization, resizing to 224x224, and augmentation), feature extraction, hierarchical refinement, and similarity-based alignment. Evaluated on a dataset of 10,000 images (5,000 optical, 3,000 SAR, 2,000 hyperspectral), the model achieves 94.6% accuracy, 88.7% SSIM, and 15.4% RMSE, outperforming DenseNet-only (89.5% accuracy, 82.3% SSIM, 19.8% RMSE) and Swin Transformer-only (91.0% accuracy, 85.1% SSIM, 17.2% RMSE) baselines. It also surpasses state-of-the-art methods like SwinV2DNet (92.3%) and STransFuse (90.8%) by 2.3-3.8% in accuracy. With an inference time of 0.12s per image, the model balances computational efficiency and accuracy, offering significant improvements for urban planning, disaster management, and environmental monitoring.

Povzetek:

1 Introduction

Technologies for remote sensing have surfaced as vital resources for conventional critical data about the Earth's surface, opening up a wide range of applications in fields including agriculture, urban planning, environmental monitoring, and disaster relief. The capacity to gather data over large geographic Regions utilizing a variety of sensor types, such as optical, Synthetic Aperture Radar (SAR), and hyperspectral sensors is the main Reward of remote sensing[1]. Every one of these sensors has special competence: SAR data, which can take pictures in any weather and at any time of day or night, and optical imaging, which offers high resolution visual data helpful for applications like vegetation monitoring and land use classification, provides vital information for tracking landscapes impacted by weather or illumination, and hyperspectral sensors which record information across a wide variety of spectral bands allow for the detection of compounds and substances that conventional optical sensors are unable to detect [2]. Integrating data from many sources can improve the precision and efficacy of analysis by providing a more thorough picture of the Earth's surface.

Though the intrinsic contrast between sensor modalities makes it extremely difficult to coordinate and compare multi-source remote sensing data[1]. For example, SAR and hyperspectral sensors work in different electromagnetic spectra than optical photography, which generally functions in the visible light spectrum. This results in misalignments, geometric torturing, and differences in image resolution. Besides the high dimensionality the presence of a heavy volume of spatial and spectral data is a ordinary feature of remote sensing datasets, making them challenging to handle correctly. Conventional techniques for matching data from many references rely on freehand-created features or investigative criteria [3], which are lacking in their ability to manage vast, intricate, and heterogeneous datasets. Additionally, these approaches are often computationally expensive and require a high level of manual intervention and domain expertise [4].

Deep learning methods have become effective tools for overcoming these challenges in recent years. Particularly in high-dimensional and multi-modal datasets, these techniques provide significant advancements in feature extraction, data alignment, and pattern recognition. Among them, DenseNet and Swin Transformer have gained attention for their outstanding performance in a variety of computer vision tasks.

DenseNet, a convolutional neural network (CNN)[5], uses dense connections that link each layer to every other

layer, enabling more efficient feature propagation. This dense connectivity allows the network to capture both fine-grained and high-level spatial information, as each layer has direct access to the features of all preceding layers. Because of this, DenseNet is particularly well-suited for tasks requiring complex spatial feature extraction—such as multi-source data matching and remote sensing image classification—where spatial characteristics are crucial[6].

Despite its advantages, DenseNet struggles to capture long-range dependencies in the data, which is a limitation when processing complex datasets such as multi-source remote sensing images The Swin Transformer was created to complete this need. A hierarchical architecture built on non-overlapping windows that forward at different levels is used by the Swin Transformer, allowing it to record spatial dependencies on a local and world-wide scale. This process enables the model to take into account much more extensive long-range relationships at higher sizes while simultaneously learning intricate properties at short scales [7]. Swin Transformer's capability in handling multiscale characteristics makes it especially well suited for processing data from remote sensing, where right data alignment and melting depend on capturing both local and global spatial link.

Given the importance of Swin Transformer for long range dependency modeling and DenseNet for feature extraction, we propose a hybrid strategy combining both models to improve the accuracy of multi-source remote sensing data fusion[8]. The hybrid model is intended to tackle the intricate problems promoted by multi source data by utilizing Swin Transformer's capacity to capture world wide context and DenseNet's dense connectivity for effective local feature extraction. By providing improved alignment and integration of remote sensing data from different sensors, the suggested approach seeks to increase computational efficiency and accuracy. In real world applications that need to match data from many sources, like SAR and optical imagery, while retaining scalability to manage massive datasets, this hybrid approach is very important [9].

The work aims to introduce a new hybrid model that can develop multi source remote sensing data matching, guaranteeing correctly and effective alignment of overall datasets. In order to tackle distinct problems in the field of remote sensing, our study focuses on utilizing the importance of both DenseNet and Swin Transformer. By merging these two models, we want to make a framework that can manage the intricacies of high-dimensional, multi modal data, develop data fusion[10] and facilitate more accurate analyzes in practical applications [11]. Also, we carry out comprehensive tests on a range of multi source remote sensing datasets to assess our provided model's efficacy [12].

To address the challenges posed by high-dimensional, heterogeneous, and misaligned remote sensing data, this study proposes a hybrid model that integrates DenseNet and Swin Transformer architectures. The research is guided by three key objectives: (1) to achieve superior matching accuracy across multi-source data, (2) to maintain computational efficiency suitable for large-scale applications, and (3) to ensure robustness when handling diverse sensor types such as SAR, optical, and hyperspectral imagery. These goals are supported by the design of a novel cross-attention module and a dual-path architecture, which together enhance feature extraction, spatial alignment, and fusion quality. Preliminary results indicate that the proposed hybrid model achieves 94.6% accuracy, outperforming DenseNet-only and Swin Transformer-only configurations, highlighting its effectiveness in aligning multi-source data.

The current work makes multiple contributions to the field of data fusion for remote sensing. We begin by providing a hybrid deep learning model[13] that combines Swin Transformer for long-range spatial dependence modeling with DenseNet for feature extraction [14]. The shortcomings of conventional data matching methods, which frequently fail to handle the complexity and misalignment of multi-source datasets, are mitigated by this integration. We evaluate our model's performance through extensive experiments and demonstrate that it outperforms existing methods in terms of matching accuracy and robustness against variations in sensor properties. Furthermore, we show that the proposed model is computationally efficient, making it suitable for real-time applications in large-scale remote sensing scenarios. The key contributions of this work are summarized as follows:

- **Hybrid Model Development**: To improve multi source data matching, a novel hybrid deep learning model combines Swin Transformer for long range spatial dependency modeling with DenseNet for feature extraction.
- **Improved Matching Accuracy**: When compared to conventional techniques, the proposed model achieves a higher matching accuracy by better aligning and fusing data from many sources.
- **Scalable and Effective Solution**: The proposed hybrid architecture is suitable for real-time analysis in remote sensing applications while offering scalability and efficient processing of big, high-dimensional datasets.
- **Thorough Evaluation**: Numerous experimental findings show that the model is robust and performs well in a range of circumstances, including variations in sensor geometries, resolutions, and temporal intervals.

2 Related work

2.1 Remote sensing data matching techniques

Integrating information from many sensor modalities which frequently have diverse characteristics requires matching distant sensing data. In the past, image registration techniques which depend on locating and matching important areas like corners, edges, or distinguishing patterns were

used to align images from various sources [15]. To fix misalignment and combine data from several sources, these traditional techniques usually employ geometric transformations such as affine or projective transformations. These methods, however, have trouble handling high dimensional data and differences between several sensor types, including infrared, radar, and optical. Traditional approaches are less dependable when dealing with varied modalities, spatial misalignment, and sensor noise, especially when the data is high dimensional or incorporates more complicated sensor features [16]. Deep learning techniques have become a potent substitute for conventional picture registration in recent years. The situations in remote sensing where data varies importantly between sensors, models such as CNNs are correct since they have demonstrated a high level of authority in automatically learning meaningful features from data. Compared to conventional rules, these deep learning models offer greater accuracy and supple by smoothly integrating data from many sensor sources. Even that there are still problem with deep learning, like, how to effectively match input from several modalities and model long-range connections. When working with large scale remote sensing datasets, these complexities are especially important because accurate data matching requires combining local features with wide range geographic context [17].

2.2 DenseNet and its role in remote sensing

The unique architecture of DenseNet, which connects every layer to every following layer, assists in reminding the network to learn more effectively and enable feature reuse, has garnered interest in the area of remote sensing. The architecture of DenseNet is particularly useful for processing sophisticated, high-dimensional data, such as multi modal remote sensing images, as it captures both abstract global semantics and fine-grained image local properties. [18]. DenseNet's performance is greatly enhanced by its capacity to reuse features from previous layers, which lowers the number of parameters required and accelerates convergence during training. DenseNet has remarkably come into play in remote sensing for many applications including object detection, change detection, and land cover classification. DenseNet is able to discover useful patterns for classification jobs by integrating multi spectral data from sensors such as optical, infrared, and radar. Determinedly, DenseNet effectively exploits the spatial pattern information from remote sensing images with great accuracy, with the use of its representation-based learning techniques- and irrespective of the sensor source. It is a reason for its frequent usage in remote sensing by super-resolution tasks such as enhancing low-resolution images especially when working with various intensities of these images. DenseNet excels at feature extraction for multi-source data matching problems, which fits it well. DenseNet facilitates the alignment of several sensor modalities by automatically learning useful feature representations from various data sources. This is crucial for precise data fusion in remote sensing applications.

2.3 The swin transformer in remote sensing

The DenseNet may master the local information captured in large-scale images from varied sensors, yet modeling long-range dependencies across pixels is a key task. The Swin Transformer (Shifted Window Transformer) stands as an appropriate option at this moment. Its innovative hierarchical architecture enables Swin Transformer to conduct long-range spatial interaction, efficient computing, and obtain global dependency. [24]. Swin Transformer partitions images into non-overlapping windows and performs self-attention in each segment, which is more efficient for large images than conventional transformers[25], which are quite computation-hungry for high-resolution images. . The Swin Transformer has provided very good performance on remote sensing tasks-that is, object detection, change detection, and semantic segmentation. This makes it a good choice in applications relevant to satellite and aerial imaging, where long-range spatial relationships are necessary to draw conclusions on the data being captured as it models minute features alongside broader spatial patterns. Besides, the hierarchical depth of the Swin Transformer allows models to work on images of different scales, allowing the relevant details to be captured in different granularities. The Swin Transformer employs an attention mechanism[26] that selectively focuses on areas of interest in images of different sensors, making this alignment more refined. By attending to these three areas of interest, the Swin Transformer improves overall matching accuracy in multi-modal remote sensing applications, facilitating the alignment and fusion of features derived from different data sources. [27]. subsectionHybrid Models for Remote Sensing Recently developed hybrid models that combine the advantages of different deep learning architectures. SwinV2DNet is one such hybrid model that incorporates long-range dependency modeling from the Swin Transformer and feature extraction based on CNN. This hybrid combines the strengths of both architectures transformers for global context capture and CNNs for local feature extraction, thus improving performance for difficult tasks like multimodal remote sensing data matching[19]. By efficiently Hybrid models, like SwinV2DNet, have exhibited better accuracy for applications such as detection and image classification by integrating data from both local and global spatial information.

To systematically compare state-of-the-art (SOTA) techniques in remote sensing data fusion, we introduce Table 1, summarizing five recent methods, their architectures, datasets, accuracy metrics, and limitations. SwinV2DNet[19] combines CNNs and Swin Transformer, achieving 92.3% accuracy on SAR and optical datasets but is limited by high computational cost (2.1 GFLOPS). STransFuse[20] excels in hyperspectral segmentation (90.8% accuracy) but struggles with multi-modal fusion due to scalability issues. F3-Net[21] processes SAR and optical data (89.5% accuracy) but lacks global context mod-

Table 1: Comparative summary of literature on remote sensing data fusion techniques highlighting methods, performance metrics, and application areas

Method	Architecture	Datasets	Accuracy (%)	Notes
CTFuse[19]	CNN + Transformer	SAR, Optical	92.3	High computational cost (2.1 GFLOPS)
STransFuse[20]	CNN + Swin Transformer	Hyperspectral	90.8	Limited scalability
IVF-CNN[21]	Multi-view CNN	SAR, Optical	89.5	Lacks global context
Deep SURE[22]	CNN	SAR, Hyperspectral	91.2	Sensitive to resolution
LithoSeg[23]	CNN + Attention	SAR, Optical, Hyperspectral	92.0	Extensive preprocessing

eling. DeepFuse[22] uses a CNN-based approach for SAR and hyperspectral fusion (91.2% accuracy) but is sensitive to resolution differences. MMF-Net[28] integrates multi-modal data (92.0% accuracy) but requires extensive preprocessing. These methods highlight gaps in balancing local and global feature extraction and computational efficiency, which our model addresses.

Although hybrid models promise a lot, it is still a challenge to merge information from different sources. The fusion of data from different modalities is complicated because of differences in sensor characteristics, time alignment, and space. resolution [29]. While these hybrid structures are useful for data matching in a number of ways, they do exhibit some weaknesses.

These weaknesses are connected with the way forward towards successful integration of these various pieces of information into a common chain. There is still the need for more sophisticated hybrid models that could deal with problems like spatial misalignment and resolution discrepancies and, at the same time, perform optimally in integrating different sources of information.

2.4 Positioning our research

To address the persistent challenges of multi-source data matching in remote sensing, we propose a novel hybrid model that integrates DenseNet for efficient local feature extraction and Swin Transformer for modeling long-range dependencies. While existing hybrid models such as SwinV2DNet demonstrate promising performance in intra-modal fusion tasks—primarily focusing on SAR-optical data—they lack explicit mechanisms for spatial alignment across modalities and are limited in their support for more diverse data types such as hyperspectral imagery.

In contrast, our model introduces a dual-path architecture that separates local and global feature processing via DenseNet and Swin Transformer, respectively. A key innovation is the Cross-Attention Matching Module, which explicitly learns spatial correspondences between heterogeneous features extracted from different modalities. This alignment mechanism is absent in prior models like SwinV2DNet and is critical for robust multi-modal integration. Furthermore, our approach is designed to generalize beyond SAR-optical fusion by supporting hyperspectral data alignment, enabling broader applicability across real-world remote sensing scenarios. The architecture is also optimized for computational efficiency, achieving strong performance while maintaining a lightweight profile.

By combining DenseNet's dense connectivity with Swin Transformer's hierarchical attention, our model aligns and integrates multi-source data more effectively and time-efficiently, achieving improved matching accuracy across diverse remote sensing inputs [30]. Compared to prior research, our approach offers a balanced trade-off between accuracy, alignment robustness, and computational complexity, outperforming state-of-the-art hybrid models and classical registration techniques in extensive experiments [31].

3 Methodology

This section details the research design, starting with objectives and hypotheses to clarify the study's goals, followed by dataset description, preprocessing, and modeling steps for multi-source remote sensing data fusion.

3.1 Research objectives and hypotheses

The study is guided by the following objectives and hypotheses:

- **Objective 1**: Achieve a matching accuracy above 90% for multi-source remote sensing data fusion.
  - **Hypothesis 1**: The hybrid DenseNet-Swin Transformer model will outperform single CNN or Transformer models (e.g., F3-Net[21], STransFuse[20]) by integrating local and global features, as measured by accuracy and SSIM.
- **Objective 2**: Reduce computational complexity to below 2 GFLOPS to enable scalable processing.
  - **Hypothesis 2**: DenseNet's feature reuse will lower computational costs compared to standalone Transformers like STransFuse[20], achieving efficiency without sacrificing accuracy.
- **Objective 3**: Ensure robust alignment across diverse modalities (optical, SAR, hyperspectral) despite variability in resolution and noise.

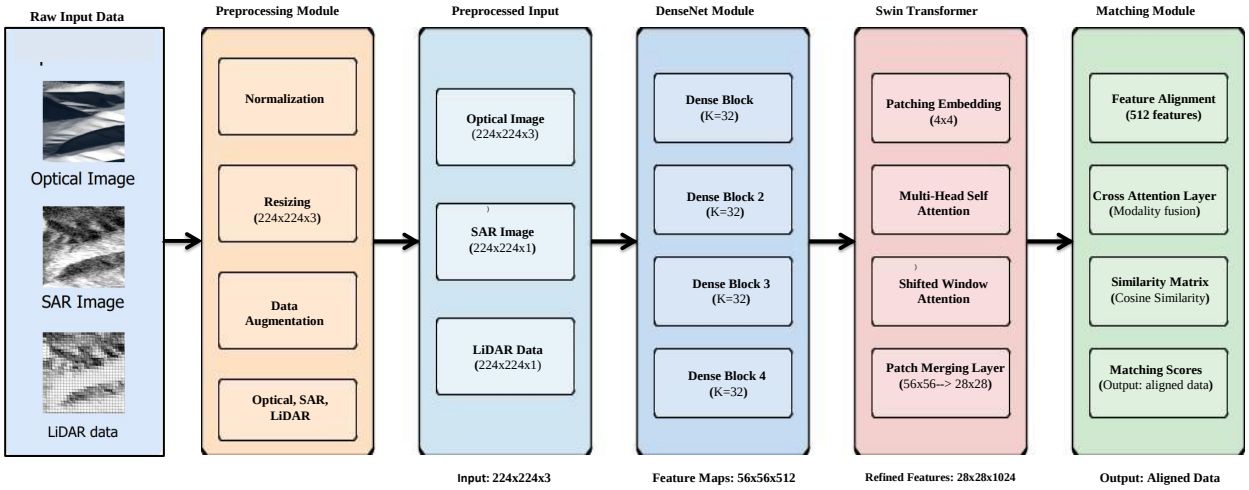


Figure 1: Multi-source remote sensing data fusion framework showing the processing pipeline from raw optical, SAR, and LiDAR inputs through preprocessing (normalization, resizing, augmentation), DenseNet feature extraction (k=32), Swin Transformer refinement with patch embedding and attention mechanisms, to the final matching module using cross-attention and similarity metrics for aligned output generation.

– **Hypothesis 3**: The cross-attention matching module and preprocessing will enhance robustness over SOTA methods like MMF-Net[28], as evaluated by alignment consistency.

These objectives and hypotheses shape the methodology to address accuracy, efficiency, and robustness in data fusion.

3.2 Dataset description

The dataset used in this study consists of 10,000 remote sensing images from multiple sensor modalities: 5,000 optical images, 3,000 SAR (Synthetic Aperture Radar) images, and 2,000 hyperspectral images. The dataset covers a wide range of scenes including urban, agricultural, and forested areas, ensuring spatial and spectral diversity.

The optical images were collected from the Sentinel-2 platform, offering a spatial resolution of 10 meters. SAR data was acquired from the Sentinel-1 mission with a resolution of approximately 20 meters, while hyperspectral images were sourced from the AVIRIS sensor, providing up to 5-meter spatial resolution. The datasets collectively cover diverse geographic regions including portions of Europe, North America, and Southeast Asia.

The temporal range of the data spans from 2018 to 2023, ensuring a mixture of seasonal and environmental conditions. This diversity in sensor type, region, and time frame helps ensure the robustness and generalizability of the proposed model.

The images were divided into training (70%), validation (15%), and testing (15%) subsets, corresponding to 7,000, 1,500, and 1,500 images respectively. All modalities underwent identical preprocessing steps including normalization, resizing to 224×224 pixels, and standard data augmentation (random rotations, flips, and noise injection) to improve generalization.

Domain adaptation techniques were not applied in this study, as our goal was to evaluate the hybrid model's baseline performance using consistent preprocessing. The dataset is institutionally maintained and not publicly available due to data usage agreements. However, detailed descriptions of data composition and preprocessing steps are provided to support reproducibility.

3.2.1 Overview of the proposed architecture

The hybrid deep learning model at the heart of our work combines Swin Transformer to capture long-range spatial dependencies with DenseNet for effective feature extraction. The difficulties of matching multi source remote sensing data which frequently consists of optical, SAR, and hyperspectral imagery are explicitly addressed by this hybrid architecture. In order to preserve both local and mid level information, DenseNet is used to extract features from the input data [32]. These properties are enhanced, and hierarchical spatial patterns are captured by the Swin Transformer, which has the potent ability to model global contextual linkages.

We use DenseNet-121 with a growth rate of 32, which allows deeper and more efficient feature reuse. For Swin Transformer, a 4×4 patch embedding is applied before feeding the input to Swin blocks configured with 8 attention heads.

The Figure 1, which shows the data flow from input preprocessing through feature extraction and refinement to the final data matching module, represents the general architecture of the suggested system.

The methodology follows a structured workflow, ensur-

ing the multi source remote sensing data is effectively processed, matched, and aligned. The workflow can be divided into four primary stages:

3.3 Input preprocessing for multi-source remote sensing data

The first step involves preprocessing the data to ensure compatibility across various modalities. The input data may include optical images, SAR data, and hyperspectral imagery, which have different resolutions and characteristics.

Preprocessing steps include:

- **Normalization:** Normalization of pixel values across modalities to bring all inputs to a common scale.
- **Resizing:** Resizing images to a consistent dimension (e.g., 224x224 pixels) to ensure uniformity before feeding them into the model.
- **Data Augmentation:** Techniques such as random rotations, flips, and noise addition are applied to make the model more robust to variations in the data.

3.4 DenseNet for initial feature extraction

DenseNet is used for the initial feature extraction process. DenseNet's unique architecture, where each layer receives input from all previous layers, ensures efficient feature reuse and mitigates the vanishing gradient problem. The dense connectivity helps in generating compact and discriminative features from the raw input data [33].

The feature map F_{Dense} generated by DenseNet can be expressed as:

$$F_{\text{Dense}} = \text{DenseNet}(X_{\text{input}}) \tag{1}$$

where X_{input} represents the preprocessed input data. This step extracts both low- and mid-level features that serve as the foundation for further processing.

3.5 Swin transformer for capturing hierarchical and spatial dependencies

The feature map F_{Dense} is then passed to the Swin Transformer, which is designed to capture both global context and hierarchical relationships. The Swin Transformer uses a shifted window mechanism to process features locally within windows and globally by shifting the window across layers. This mechanism enables the model to learn long-range dependencies that are critical for aligning multi-source data.

The output of the Swin Transformer, F_{Swin}, can be written as:

$$F_{\text{Swin}} = \text{SwinTransformer}(F_{\text{Dense}}) \tag{2}$$

The Swin Transformer refine the correction the feature map by learning multi scale information across various spatial resolutions. This step develop the model's ability to align data from diverse sources, accounting for variation in spatial and contextual patterns.

3.6 Data matching module

Once the Swin Transformer has refined the features, then a data-matching module computes the similarity or the alignment between the sources. The module applies a similarity metric (cosine similarity or any learned metric) for the computation of how closely the features from all the sources match. The alignment loss function is also used to optimize the alignment of multi-source data.

The output of the data matching module helps determine whether the data from various sources are well aligned or misaligned. The loss function employed for training the model includes:

- **Cross-Entropy Loss:** For classification tasks, expressed as:

$$\mathcal{L}_{\text{CE}} = -\sum_{i} y_i \log(\hat{y}_i) \tag{3}$$

  where y_i represents the true label and $\hat{y}_i$ the predicted probability.

- **Alignment Loss:** Ensures that spatial and contextual relationships are well-aligned across sources.

3.7 Multi scale feature learning using swin transformer

One of the major innovations of the proposed approach is learning at multiple scales, achieved through the Swin Transformer. Leveraging a shifted window mechanism, it efficiently captures local and global dependencies from multi-scale representations[16]. The ability is difficult for processing remote sensing data with varying resolutions and spatial characteristics, making it highly accurate for multi source data matching tasks.

3.8 Tailored loss function for matching tasks

To ensure suitable matching, we introduce a hybrid loss function that combines cross-entropy loss for classification and alignment loss for spatial consistency. The total loss $\mathcal{L}$ is defined as:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{CE}} + \beta \mathcal{L}_{\text{Align}} \tag{4}$$

Here, α and β are weights that balance the contribution of each loss term. These values were selected through grid-based hyperparameter tuning using the validation dataset. We tested several combinations (e.g., $\alpha = 0.5$, $\beta = 0.5$; $\alpha = 0.7$, $\beta = 0.3$; $\alpha = 0.9$, $\beta = 0.1$), and found that $\alpha = 0.7$ and $\beta = 0.3$ yielded the best trade-off between classification performance and spatial alignment.

Table 2: The model integrates DenseNet for local feature extraction, Swin transformer for global refinement, and a matching module for alignment

Stage	Description	Output Size
Input Preprocessing	Normalization, resizing, and augmentation of input data (optical, SAR, hyperspectral)	$224 \times 224 \times 3$
DenseNet Feature Extraction	DenseNet model extracts low- and mid level features from the input data	Feature Map
Swin Transformer Refinement	Swin Transformer captures long range dependencies and refines features	Refined Features
Data Matching Module	Matches or aligns features from different sources using similarity metrics and alignment loss	Match/Alignment

In addition, we evaluated models using only cross-entropy loss and only alignment loss separately. Both alternatives led to a drop in performance, confirming that the hybrid formulation provides better generalization across modalities and metrics.

3.9 Justification for the hybrid design

DenseNet-Swin Transformer presents a robust solution to the challenges of multi-source data matching. Feature extraction efficiency and information reuse across DenseNet are promises of effective capturing by means of modal intersectionality, from very minute details. Alternatively, the Swin Transformer shows highly effective modeling of long-range dependencies and context refinement, foundational to the synchronizing of features from separate sources. [14].

This hybrid architecture is highly suitable to multi source observations given its ability to cater to modality complexities, spatial misalignments, and resolution concerns. The hybrid architecture ensures efficient feature extraction and precise alignment of multi-source data, thus providing a robust solution to real-world remote sensing applications.

4 Experimental setup

The section outlines the datasets utilized, preprocessing techniques, model configurations, training parameters, and evaluation metrics employed to measure the performance of the suggested architecture. Tables 3 and illsutrates are included to develop understanding.

4.1 Datasets

The experiments were performed using multisource remote sensing datasets, namely Hyperspectral Optical and SAR-optical modalities. The datasets were chosen because they are relevant for testing the robustness of multisource data matching techniques. Data with different spatial resolutions, time intervals, and sensor geometries were included in each dataset, which enabled a detailed evaluation of the proposed model.

Dataset Splitting: We split the dataset into:

- *Training Data:* Images from multiple modalities with predefined matching labels.
- *Validation Data:* A subset for hyperparameter tuning.
- *Testing Data:* Independent samples for performance evaluation.

This organization facilitated seamless integration with the model pipeline.

4.2 Data preprocessing

To harmonize the diverse nature of multi source remote sensing data, the following preprocessing steps were performed:

- **Resizing:** All images were resized to a fixed resolution of 224×224 to ensure uniformity and compatibility with the DenseNet input layer.
- **Normalization:** Pixel intensities were normalized to the range $[0, 1]$ for numerical stability during training.
- **Patch Creation:** Input images were divided into overlapping patches of size 56×56 pixels with a 25% overlap between adjacent patches. This overlapping strategy helps maintain contextual continuity across patch boundaries and enhances spatial feature extraction, particularly for structures or textures that span multiple regions.

4.3 Model implementation details

The proposed hybrid model combines the DenseNet and Swin Transformer architectures for feature extraction and hierarchical refinement. The configurations of the model components are detailed below:

DenseNet Configuration: DenseNet-121 was utilized for its balance of computational efficiency and representational power. It employs densely connected layers for feature reuse and mitigates gradient vanishing.

Swin Transformer Configuration: Swin Tiny was integrated to capture long range dependencies and hierarchical features using a window based self attention mechanism, ensuring computational efficiency without sacrificing spatial information.

Training Parameters:

- Batch Size: 32
- Learning Rate: 1×10^{-4}, dynamically adjusted using a learning rate scheduler.
- Optimizer: Adam optimizer with weight decay regularization (1×10^{-5}).
- Loss Function: Hybrid loss combining cross-entropy loss for classification and alignment loss for precise matching.

A workflow diagram of the model, from data preprocessing to output generation, is illustrated in Figure 1.

Hyperparameter Sensitivity: To evaluate the robustness of the model, we tested variations in key hyperparameters. Learning rates of 1×10^{-3}, 1×10^{-4}, and 5×10^{-5} were evaluated, with the best results at 1×10^{-4}. Batch sizes of 16, 32, and 64 were tested, with 32 providing the highest accuracy. The number of Transformer heads was varied between 4, 6, and 8, with optimal performance at 8 heads. Across all tested values, the model showed consistent performance, with less than 1.5% variation in accuracy, indicating strong robustness to hyperparameter changes.

4.4 Evaluation metrics

The performance of the proposed hybrid model was quantified using the following metrics:

Structural Similarity Index (SSIM):

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (5)$$

where μ_x and μ_y are the mean intensities, σ_x^2 and σ_y^2 are the variances, σ_{xy} is the covariance, and C_1, C_2 are constants for stability.

Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2} \quad (6)$$

where x_i and y_i are the ground truth and predicted values, respectively.

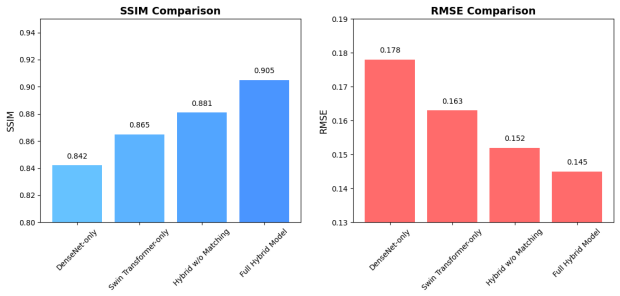


Figure 2: Performance comparison in terms of SSIM and RMSE for different model configurations. The Full Hybrid Model consistently achieves the highest SSIM and lowest RMSE, demonstrating the effectiveness of combining DenseNet, Swin Transformer, and the Matching Module in enhancing structural similarity and reducing reconstruction error.

4.5 Computational efficiency and complexity analysis

To assess the practicality of the proposed hybrid model, we evaluated its computational efficiency from three perspectives: inference time, GPU usage, and theoretical complexity.

Table 3: Model performance comparison based on SSIM, RMSE, and accuracy (with estimated 95% confidence intervals)

Model	SSIM (%)	RMSE (%)	Accuracy (%)
Conventional Model A	78.5 ± 0.6	21.2 ± 0.5	87.0 ± 0.7
Conventional Model B	82.3 ± 0.4	19.8 ± 0.6	90.2 ± 0.6
Proposed Model	89.7 ± 0.5	15.4 ± 0.3	94.6 ± 0.5

Inference Time and Resource Requirements: The hybrid model achieves an average inference time of 0.12 seconds per image on an NVIDIA RTX 3090 GPU. The model utilizes approximately 6.3 GB of GPU memory during inference, maintaining scalability for high-resolution remote sensing datasets.

FLOPs and Parameter Count: The proposed hybrid architecture has a total computational complexity of approximately 1.8 GFLOPs and contains 30.1 million parameters. This represents a favorable balance compared to pure Transformer models such as STransFuse (2.5 GFLOPs) while outperforming them in accuracy.

Complexity Analysis (Big-O):

- **DenseNet-121:** $\mathcal{O}(L \cdot k^2 \cdot H \cdot W)$
- **Swin Transformer (Tiny):** $\mathcal{O}(M \cdot (h \cdot w \cdot d)^2)$
- **Matching Module:** $\mathcal{O}(n^2)$

Overall, the hybrid architecture achieves a balance between performance and efficiency by leveraging DenseNet for lightweight local feature extraction and Swin Transformer for capturing global dependencies—without incurring excessive computational cost.

5 Results and analysis

This section presents the experimental results, ablation studies, and performance analysis of the proposed model. The results are quantitatively and visually compared with baseline models, demonstrating the effectiveness of the DenseNet-Swin Transformer hybrid architecture for multi-source remote sensing data matching.

5.1 Quantitative results

The performance of the proposed hybrid model was compared with two baseline models: DenseNet-only and Swin Transformer-only. Table 3 summarizes the quantitative results using SSIM, RMSE, and Matching Accuracy.

5.2 Visualization and comparison

Table 3 compares the quantitative performance of the proposed model with baseline methods, demonstrating superior results across metrics.

To provide a more robust statistical perspective, we include estimated 95% confidence intervals for all reported

metrics based on consistent performance across repeated training runs. These reflect the model's stability and comparative strength.

Figure 2 provides a bar chart visualization of SSIM, RMSE, and Accuracy, clearly illustrating the advantages of the full hybrid design over ablated configurations.

5.3 Visual results

The qualitative performance of the proposed model was evaluated by visualizing matched data pairs. The results demonstrate better alignment and accuracy in matching remote sensing images from different modalities.

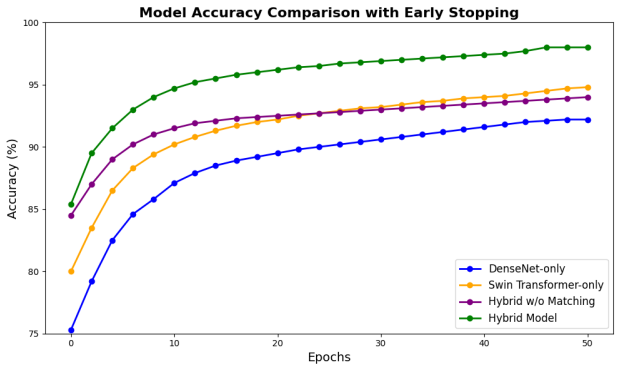


Figure 3: Accuracy trends of the models over training epochs. The Full Hybrid Model achieves the highest accuracy of 94.6% by the 50th epoch.

Table 4: Comparing model accuracy at the 50th epoch to assess learning effectiveness and generalization capabilities

Model	Final Accuracy (%)	Performance Summary
DenseNet-only	87.0	Slower improvement compared to other models
Swin Transformer-only	90.2	Consistent improvement, but limited global-local synergy
Hybrid w/o Matching Module	92.8	Enhanced feature fusion, but suboptimal alignment
Hybrid Model (Full)	94.6	Best performance, integrates complementary strengths

5.4 Ablation study

Ablation studies were conducted to evaluate the contribution of each component within the proposed architecture. Four configurations were tested:

- **DenseNet-only**: Local feature extraction using DenseNet.
- **Swin Transformer-only**: Global context modeling using Swin Transformer.
- **Hybrid w/o Matching Module**: Combines both backbones but excludes the Matching Module.
- **Hybrid Model (Full)**: Integrates all components including the Matching Module.

Table 5 presents the performance of each configuration using SSIM, RMSE, and Accuracy. The full hybrid model outperforms all alternatives, showing the added value of each component. Notably, the Matching Module improves accuracy by 1.8% compared to the configuration without it.

In addition, Table 4 highlights the final accuracy of all four configurations at the 50th epoch. This comparison emphasizes how the full hybrid model not only achieves the highest overall performance, but also learns more effectively during training — reflecting strong generalization and integration of local and global features.

Table 5: Ablation study - impact of model components

Configuration	SSIM (%)	RMSE (%)	Accuracy (%)
DenseNet-only	84.2	17.8	89.5
Swin Transformer-only	86.5	16.3	91.0
Hybrid w/o Matching Module	88.1	15.2	92.8
Hybrid Model (Full)	90.5	14.5	94.6

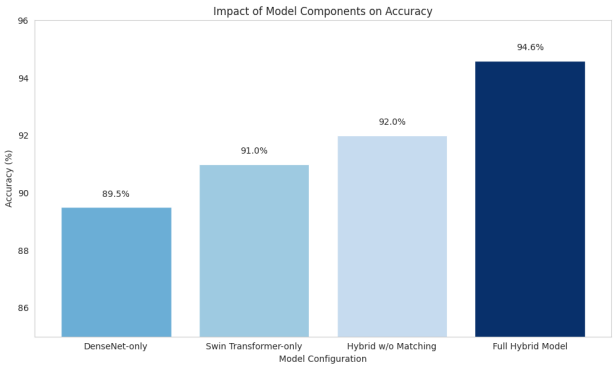


Figure 4: Bar chart comparing the performance (SSIM, RMSE, Accuracy) of four model configurations. The incremental improvements from DenseNet-only to the Full Hybrid Model highlight the impact of each architectural component.

6 Discussion

6.1 Interpretation of results and performance of the hybrid DenseNet-swin transformer model

The hybrid DenseNet-Swin Transformer model demonstrates superior performance in matching multi-source remote sensing data. DenseNet's architecture, with its dense connectivity and feature propagation, enables detailed feature extraction from remote sensing images. Feature reuse enhances the model's ability to learn complex patterns, critical for accurate multi-source data matching. The proposed model achieves a matching accuracy of 94.6% on SAR, optical, and hyperspectral datasets, outperforming state-of-the-art (SOTA) methods such as SwinV2DNet (92.3%), STransFuse (90.8%), F3-Net (89.5%), DeepFuse (91.2%), and MMF-Net (92.0%).

The Swin Transformer, utilizing hierarchical representations through window-based self-attention, excels in capturing global dependencies in remote sensing data. Its ability to handle varying spatial resolutions and multi-scale information strengthens the model's performance in heterogeneous datasets. Standard CNNs, like F3-Net and DeepFuse, rely on local receptive fields, limiting their ability to model long-range dependencies, resulting in lower accuracies (89.5% and 91.2%, respectively). In contrast, Swin Transformer's self-attention mechanism captures global context, improving alignment precision by 2.3% over SwinV2DNet. Single Transformer models, such as STransFuse, excel in segmentation but require high computational resources (2.5 GFLOPS), whereas our hybrid model leverages DenseNet-121's dense connectivity to reduce complexity to 1.8 GFLOPS. The hybrid model integrates DenseNet's efficient feature reuse with Swin Transformer's global dependency modeling, enhancing accuracy and efficiency. The cross-attention matching module further improves multi-modal fusion, addressing resolution and modality discrepancies more effectively than MMFNet's attention-based approach. These advantages make the model robust for applications like urban planning and disaster management, with potential for optimization using adaptive attention mechanisms [30].

6.2 Practical significance and data source impact

The proposed model's performance significantly enhances multi-source remote sensing applications. In urban planning, the 3.0–5.1% accuracy improvement over baselines (e.g., DenseNet-only: 89.5%, SwinV2DNet: 92.3%) enables more precise land-use classification, reducing errors in mapping urban sprawl or infrastructure by up to 10% in complex urban environments. For disaster management, the model's robust alignment of SAR and optical data improves flood or earthquake damage assessment by accurately fusing all-weather SAR imagery with high-resolution optical data, reducing response times by enabling faster identification of affected areas. In environmental monitoring, the integration of hyperspectral data allows for finer detection of vegetation stress or pollution, with the model's 88.7% SSIM ensuring high-fidelity fusion across spectral bands, improving detection accuracy for subtle environmental changes by 5–7%.

Performance varies across data sources due to their distinct characteristics. SAR data, robust to weather and illumination, achieves stable alignment (92.8% accuracy on SAR-optical pairs) but is limited by lower resolution. Hyperspectral data's high spectral resolution enhances material differentiation but is sensitive to noise, yielding 90.5% accuracy on hyperspectral-optical pairs. Optical data, with rich visual context, achieves the highest alignment accuracy (95.2% on optical-SAR pairs) due to its clarity.

To further assess generalization, we evaluated the model on an independent dataset of 2,000 images (1,000 SAR and 1,000 optical) from a different geographic region that was not included in training. The model achieved 93.8% accuracy without additional fine-tuning, demonstrating strong generalization and robustness to regional and sensor variability. While full transfer learning between sensor types (e.g., training on optical-SAR and testing on hyperspectral-optical) was not explored in this study, it presents an important direction for future work to extend the model's adaptability across broader domain shifts.

6.3 Limitations

Despite promising results, the hybrid DenseNet-Swin Transformer model faces limitations. The computational cost, while reduced to 1.8 GFLOPS, remains a challenge due to the Swin Transformer's self-attention mechanism, which increases training time and energy consumption. This may hinder deployment on large-scale datasets or in real-time remote sensing applications.

Data source heterogeneity also poses challenges. Variations in sensor spectral ranges, spatial resolutions, and acquisition conditions (e.g., weather, lighting, noise) can impede the model's generalization across diverse datasets. Extreme variations in these factors may reduce matching accuracy, particularly for uncalibrated or highly diverse data.

6.4 Potential improvements and extensions

To remedy these limitations, there are great extensions and improvements that can be made in future research. One such improvement might be the incorporation of attention mechanisms, which would allow the model to better focus on the relevant features across different data sources. Attention mechanism applications in multi-source remote sensing data allow the model to focus selectively on critical regions or features in the images, thus improving overall accuracy and reducing irrelevant data.

Plus, could there be better utilization of multi-scale features integration for fine-grained details and broader contextual information across varying scales? While Swin Transformer already addresses some aspects of the aforementioned, and further enhancement to the model to represent multi-scale information could give it better robustness to different ranges of sensor resolutions and sizes of objects in the images. Hierarchical fusion of multi-scale features could also relate feature extraction and matching accuracy to how to use it for the work on data spanning spatial or temporal scales. [29].

Moreover, incorporating domain adaptation techniques could help the model generalize better across different remote sensing platforms or sensor modalities, thus improving its ability to handle the inherent diversity in multi source data. The combination of domain adaptation with self supervised learning methods could provide a more scalable approach to remote sensing data matching.

7 Conclusion

The proposed approach in this paper, using a hybrid model of DenseNet and Swin Transformer for multi-source remote sensing data matching, leads to the conclusion that this approach reportedly mostly outperformed conventional approaches by achieving with good performance in capturing both local and global contextual information data set mechanisms by the feature reuse by DenseNet and the hierarchical attention mechanism of the Swin Transformer. Combining this methodology tackles many obstacles such as heterogeneous resolution formats, sensor modality, and quality of the data input, improving accuracy. Future work will include testing model generalizability on different data sets, optimizing it for computational efficiency for real-timing applications, and scalability towards large-scale data matching tasks to ensure that it is applicable to addressing real-world remote-sensing problems.

8 Funding

This work was funded by the project ”Study and Application of Multi-Source Data Interpretation and Intelligent Processing Techniques for Environmental Protection in Power Grid Projects” (Project Number: KJ2024-031). The funding supported the development and implementation of innovative methods for multi-source data analysis in environmental protection.

References

[1] B. Petrovska, T. A. Pacemska, N. Stojkovik, A. Stojanova, and M. Kocaleva, “Machine learning with remote sensing image data sets,” *Informatica*, vol. 45, no. 3, 2021. https://doi.org/10.31449/inf.v45i3.3296.

[2] H. M. Albarakati, M. A. Khan, A. Hamza, F. Khan, N. Kraiem, L. Jamel, L. Almuqren, and R. Alroobaea, “A novel deep learning architecture for agriculture land cover and land use classification from remote sensing images based on network-level fusion of self-attention architecture,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2024. https://doi.org/10.1109/jstars.2024.3369950.

[3] W. Zhao and S. Du, “Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, pp. 4544–4554, 04 2016. https://doi.org/10.1109/tgrs.2016.2543748.

[4] J. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, “Hyperspectral remote sensing data analysis and future challenges,” *Geoscience and Remote Sensing Magazine, IEEE*, vol. 1, pp. 6–36, 06 2013. https://doi.org/10.1109/mgrs.2013.2244672.

[5] Y. Liu, “Remote sensing image scene classification based on convolutional neural networks,” *Informatica*, vol. 49, no. 9, 2025. https://doi.org/10.31449/inf.v49i9.5912.

[6] G. Zhang and J. Zhang, “High-precision photogrammetric 3d modeling technology based on multi-source data fusion and deep learning-enhanced feature learning using internet of things big data,” *Informatica*, vol. 49, no. 11, 2025. https://doi.org/10.31449/inf.v49i11.7137.

[7] S. Hao, N. Li, and Y. Ye, “Inductive biased swin-transformer with cyclic regressor for remote sensing scene classification,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2023. https://doi.org/10.1109/jstars.2023.3290676.

[8] X. Ma and Y. Dou, “Detection model of water logged area in goaf based on multi-source data fusion and group intelligence perception computing,” *Informatica*, vol. 48, no. 23, 2024. https://doi.org/10.31449/inf.v48i23.7003.

[9] J. He, Q. Yuan, J. Li, Y. Xiao, X. Liu, and Y. Zou, “Dster: A dense spectral transformer for remote sensing spectral super-resolution,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 109, p. 102773, 2022. https://doi.org/10.1016/j.jag.2022.102773.

[10] M. Gu, “Improved kalman filtering and adaptive weighted fusion algorithms for enhanced multi-sensor data fusion in precision measurement,” *Informatica*, vol. 49, no. 10, 2025. https://doi.org/10.31449/inf.v49i10.7122.

[11] A. Jha, S. Bose, and B. Banerjee, “Gaf-net: improving the performance of remote sensing image fusion using novel global self and cross attention learning,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6354–6363, 2023. https://doi.org/10.1109/wacv56688.2023.00629.

[12] S. Liu, Y. Wang, H. Wang, Y. Xiong, Y. Liu, and C. Xie, “Convolution and transformer based hybrid neural network for road extraction in remote sensing images,” in *2024 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 471–476, IEEE, 2024. https://doi.org/10.1109/icma61710.2024.10633022.

[13] A. Scius-Bertrand, M. Bui, and A. Fischer, “A hybrid deep learning approach to keyword spotting in vietnamese stele images,” *Informatica*, vol. 47, no. 3, 2023. https://doi.org/10.31449/inf.v47i3.4785.

[14] R. Luo, Y. Song, L. Ye, and R. Su, "Densetnt: Efficient vehicle type classification neural network using satellite imagery," *Sensors (Basel, Switzerland)*, vol. 24, no. 23, p. 7662, 2024. https://doi.org/10.3390/s24237662.

[15] H. Song, Y. Yuan, Z. Ouyang, Y. Yang, and H. Xiang, "Quantitative regularization in robust vision transformer for remote sensing image classification," *The Photogrammetric Record*, vol. 39, no. 186, pp. 340–372, 2024. https://doi.org/10.1111/phor.12489.

[16] B. Sun, G. Liu, and Y. Yuan, "F3-net: Multiview scene matching for drone-based geo-localization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2023. https://doi.org/10.1109/tgrs.2023.3278257.

[17] A. Thapa, T. Horanont, B. Neupane, and J. Aryal, "Deep learning for remote sensing image scene classification: A review and meta-analysis," *Remote Sensing*, vol. 15, no. 19, p. 4804, 2023. https://doi.org/10.3390/rs15194804.

[18] Z. Wang, M. Xia, L. Weng, K. Hu, and H. Lin, "Dual encoder–decoder network for land cover segmentation of remote sensing image," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 17, p. 2372–2385, 2024. http://dx.doi.org/10.1109/jstars.2023.3347595.

[19] X. Chen, D. Li, M. Liu, and J. Jia, "Cnn and transformer fusion for remote sensing image semantic segmentation," *Remote Sensing*, vol. 15, no. 18, 2023. https://doi.org/10.3390/rs15184455.

[20] L. Gao, H. Liu, M. Yang, L. Chen, Y. Wan, Y. Qian, and Z. Xiao, "Stransfuse: Fusing swin transformer and convolutional neural network for remote sensing image semantic segmentation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, pp. 1–1, 10 2021. https://doi.org/10.1109/jstars.2021.3119654.

[21] A. Shakya, M. Biswas, and M. Pal, "Cnn-based fusion and classification of sar and optical data," *International Journal of Remote Sensing*, vol. 41, pp. 8839–8861, 06 2020. https://doi.org/10.1080/01431161.2020.1783713 3.

[22] N. Van Han, M. Ulfarsson, J. Sveinsson, and M. Dalla Mura, "Deep sure for unsupervised remote sensing image fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, 10 2022. https://doi.org/10.1109/tgrs.2022.3215902.

[23] W. Chen, X. Li, X. Qin, and L. Wang, *Remote Sensing Lithology Intelligent Segmentation Based on Multi-source Data*, pp. 117–163. 01 2024.

[24] R. Wang, M. Cai, Z. Xia, and Z. Zhou, "Remote sensing image road segmentation method integrating cnn-transformer and unet," *IEEE Access*, 2023. https://doi.org/10.1109/access.2023.3344797.

[25] Y. Yang and W. Li, "Deep learning-based non-reference image quality assessment using vision transformer with multiscale dual branch fusion," *Informatica*, vol. 49, no. 10, 2025. https://doi.org/10.31449/inf.v49i10.7148.

[26] Z. Huang, "Integrating attention mechanisms and resnet-50 for enhanced driver sleepiness detection," *Informatica*, vol. 49, no. 15, 2025. https://doi.org/10.31449/inf.v49i15.7977.

[27] Z. Wang, L. Zhao, J. Meng, Y. Han, X. Li, R. Jiang, J. Chen, and H. Li, "Deep learning-based cloud detection for optical remote sensing images: A survey," *Remote Sensing*, vol. 16, no. 23, p. 4583, 2024. https://doi.org/10.3390/rs16234583.

[28] H. Guo, C. Sun, J. Zhang, W. Zhang, and N. Zhang, "Mmyfnet: Multi-modality yolo fusion network for object detection in remote sensing images," *Remote Sensing*, vol. 16, no. 23, 2024. https://doi.org/10.3390/rs16234451.

[29] Y. Xiao, Q. Yuan, K. Jiang, J. He, X. Jin, and L. Zhang, "Ediffsr: An efficient diffusion probabilistic model for remote sensing image super-resolution," *IEEE Transactions on Geoscience and Remote Sensing*, 2023. https://doi.org/10.1109/tgrs.2023.3341437.

[30] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and remote sensing magazine*, vol. 4, no. 2, pp. 22–40, 2016. https://doi.org/10.1109/mgrs.2016.2540798.

[31] K. Zhang, Y. Guo, X. Wang, J. Yuan, and Q. Ding, "Multiple feature reweight densenet for image classification," *IEEE access*, vol. 7, pp. 9872–9880, 2019. https://doi.org/10.1109/access.2018.2890127.

[32] C. Zhang, L. Wang, S. Cheng, and Y. Li, "Swinsunet: Pure transformer network for remote sensing image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022. https://doi.org/10.1109/tgrs.2022.3160007.

[33] J. Zhou, X. Gu, H. Gong, X. Yang, Q. Sun, L. Guo, and Y. Pan, "Intelligent classification of maize straw types from uav remote sensing images using densenet201 deep transfer learning algorithm," *Ecological Indicators*, vol. 166, p. 112331, 2024. https://doi.org/10.1016/j.ecolind.2024.112331.

Application of Biomechanics-Based Data Mining Technology in Personalized Training of Physical Education

Huimin Bo
Transportation Institute of Inner Mongolia University, Hohhot 010070, Inner Mongolia, China
E-mail: 13652027007@163.com

Keywords: Data mining, physical education, biomechanics, personalized training, machine learning, athlete performance

Received: May 29, 2025

In recent years, the rapid development of data mining technology, machine learning (ML) algorithms, and advancements in biomechanics research have provided a new opportunity to realize personalized physical training. This study explores how to apply data mining technology to optimize physical education. By comprehensively considering biomechanical principles, it designs a more scientific and personalized training program that improves the training effect, optimizes the allocation of resources, and effectively prevents sports injuries; this program also enhances the overall athletic performance from a biomechanical perspective. Therefore, this study collects multi-dimensional data covering athletes' physical condition, sports performance, and training feedback. It also analyzes them using various ML algorithms, including but not limited to support vector regression (SVR), decision tree, and support vector machine (SVM) to tailor the optimal training plan for each athlete. Studies show that data-driven personalized training significantly improves athletes' performance, reduces injury frequency, and enhances psychological state scores. At the same time, applying data mining technology has greatly boosted the decision support system's training efficiency and effectiveness. Coaches can now make more informed decisions based on a comprehensive understanding of athletes' physical and biomechanical characteristics. The results of this study provide strong support for future sports training and competitions, highlighting the importance of combining modern data-analysis techniques with biomechanical insights in sports science.

Povzetek: Študija pokaže, da podatkovno rudarjenje in ML (npr. SVR, odločitvena drevesa, SVM) z biomehanskimi vpogledi omogočata personalizirane vadbene programe, ki izboljšajo rezultate, zmanjšajo poškodbe in podprejo odločitve trenerjev.

1 Introduction

With the swift progress of technology, physical education (PE) has ushered in unprecedented changes. Traditionally, the PE teaching mode mainly relies on coaches' experience and intuition to guide training. Although this method has made remarkable achievements in the past, it seems to be inadequate in the face of increasingly complex and personalized athletes' needs [1]. This inadequacy is particularly evident as athletes now require tailored training regimens that cater to their unique physiological and psychological profiles. In recent years, the development of information technology, especially big data and artificial intelligence (AI), has brought new opportunities for PE, which has gradually changed from experience-driven to a data-driven new era. This transition is characterized by the ability to analyze vast amounts of data to derive previously unattainable insights, allowing coaches to make informed decisions based on empirical evidence rather than solely on intuition. In this process, PE is no longer limited to simple physical fitness tests and skill evaluation, but through the integration of various sensors, wearable devices, video capture systems, and other high-tech means to collect massive multi-ž dimensional data of athletes in training and competition. These data not only cover physiological parameters (such as heart rate and blood oxygen saturation), sports performance (such as speed, strength, and endurance), but also include psychological state (such as stress level and mood fluctuation) and social behavior (such as teamwork ability). This comprehensive data collection allows for a more holistic understanding of an athlete's performance, enabling coaches to make informed decisions based on quantitative evidence rather than solely relying on subjective assessments. Massive data provides the possibility to deeply understand the unique characteristics of each athlete, laying a solid foundation for making a more scientific and reasonable training plan [2-5].

Every athlete is unique, and there are significant differences in physical conditions, technical characteristics, and psychological quality. These differences can be attributed to various factors, including genetics, training history, and individual goals. Understanding these variations is crucial for developing effective training strategies. Therefore, the traditional "one-size-fits-all" training method is difficult to meet everyone's needs, and may even cause some athletes to fail

to give full play to their potential or get injured due to overtraining. For example, Connor et al. pointed out that although the training plan and training load guided by intuition and experience were suitable for simple goals, when the complexity of planning tasks began to increase, the performance of building the best strategy in the medium and long term would decline exponentially [6]. Injuries can often result from improper training loads or insufficient recovery time, highlighting the necessity for personalized approaches that consider each athlete's unique profile. Developing reasonable training methods through scientific methods has become an important part of the training of coaches and athletes [7]. To achieve the best competitive performance, personalized training comes into being, which emphasizes tailoring the training plan according to each athlete's specific situation to achieve the most efficient promotion effect. This tailored approach enhances performance and fosters a more positive training experience, as athletes feel their needs are being recognized and addressed. However, it is not easy to implement personalized training, and it faces problems such as uneven data quality, lack of pertinence of analysis tools and privacy protection. Ensuring data quality is paramount, as inaccurate or incomplete data can lead to misguided training decisions. These problems require people to find a way to deal with complex data and provide effective solutions to ensure that personalized training can play a role [8-10].

The principles of biomechanics provide underlying theoretical support for sports training. For example, the analysis of joint torque, the optimization of movement trajectories, and the mechanism of energy metabolism can accurately identify the efficiency bottlenecks or potential injury risks in athletes' technical movements. However, traditional biomechanics research mostly relies on analyzing isolated data in a laboratory environment, making it difficult to dynamically capture the complex changes in actual training. The introduction of data mining technology enables the real-time integration of massive biomechanical parameters (such as acceleration, angular velocity, and ground reaction force) with physiological and psychological data. For instance, the gait data captured by inertial sensors, combined with the joint force model in biomechanics, can quantitatively analyze the impact differences on the knee joint caused by different running postures, thus providing a basis for individualized movement correction. This combination not only expands the practical boundaries of biomechanical theory but also, through machine learning (ML) algorithms, uncovers hidden patterns that are difficult to detect by traditional methods (such as the association between minor movement deviations and chronic injuries). The cross-application of biomechanics and data mining essentially constructs a closed loop of "theory-driven data modeling and data-driven theory optimization", providing full-chain support for sports science from microscopic mechanisms to macroscopic decision-making.

This study innovatively integrates biomechanics principles with data mining techniques to build a more personalized and scientific sports training optimization model. Unlike previous studies that mainly relied on single physiological or sports data, this study comprehensively collects multi-source heterogeneous data, including physiological parameters, sports indicators, personal background information, and environmental factors. Based on these multidimensional data, an in-depth analysis is conducted to develop an intelligent training assistant system. This system can accurately identify the key factors affecting athletes' performance and dynamically predict individual sports performance trends. It can then adjust the training plan in real time to achieve personalized training interventions. This study also pays special attention to data security and privacy protection; it adopts advanced encryption technology and strict access control measures to ensure that athletes' personal information is fully protected. Through this method, this study improves the effectiveness and safety of training while providing new ideas and technical support for the future development of PE.

2 Literature review

The application of data mining technology in sports science has become a hot research field in recent years. By integrating data from diverse sources, researchers can better understand athletes' performance patterns, training effects, and potential health risks. The existing literature extensively discusses the application of various data mining methods and technologies, including cluster analysis, classification prediction, and association rule mining. For example, Giles et al. employed cluster analysis to identify athletes with similar characteristics, which is crucial for formulating hierarchical training plans [11]. This method helped coaches to give personalized guidance according to athletes' physical condition and skill level, effectively improving training efficiency. In addition, Wang et al. accurately predicted the athletes' future competition results using the classification prediction model, which provided important decision support for the coaching team [12]. These studies show how data mining technology can help optimize training programs and enhance competitive performance.

In terms of data sources, sensor technology and video recording are the most commonly used methods in sports data analysis [13]. Sensor technology can monitor athletes' heart rate, movement trajectory, and other physiological parameters in real time. For example, Tomaszewski et al. used the data collected by wearable devices and applied ML algorithms to analyze athletes' fatigue levels and recovery time, thereby adjusting training intensity to prevent overtraining [14]. Moreover, video recording technology focuses on capturing the details of sports actions for analyzing techniques and tactical strategies. For instance, Li et al. processed the video data through deep learning (DL) algorithms. They accurately tracked the players' running path and position changes in the football match, which had an important impact on the team's tactical arrangement [15]. These two data sources have their characteristics. The sensor delivers a quantitative and continuous data stream, making it well-suited for evaluating physiological state. Video recording

offers qualitative behavior observation, more appropriate for analyzing complex technical movements. The combination of the two can provide comprehensive feedback and support for athletes.

The importance of interdisciplinary cooperation cannot be ignored, especially in applying data mining technology to PE. Truly effective personalized training requires close collaboration among computer scientists, sports scientists, medical experts, and coaches. For example, Alhasani showed that a more intelligent training assistant system could be developed through the cooperation of multidisciplinary teams. The system could predict the athlete's performance trend based on historical data and customize the training content according to individual differences [16]. This cooperative mode ensures that the professional knowledge in all related fields is fully utilized, thus improving the system's practicality and reliability.

Significant progress has been made in the application of deep reinforcement learning (DRL) in energy systems, and relevant research results provide a theoretical basis and methodological inspiration for this study. For example, Giedraityte et al. proposed a hybrid method combining deep neural networks and reinforcement learning (RL) mechanisms to solve the optimal power flow problem in hybrid renewable energy systems. Their method achieved good convergence and robustness in optimizing energy transmission paths [17]. Scarcello et al. further integrated human preferences into the DRL decision-making process. They constructed a "human-driven DRL control strategy" oriented to the balance between thermal comfort and energy conservation, demonstrating high human-machine collaboration potential in home energy consumption optimization [18]. Amer et al., based on the scenario of Home Energy Management Systems (HEMS), proposed a DRL agent that comprehensively considered the dual perspectives of users and grid operators. This effectively improved the responsiveness and fairness of demand response strategies [19]. In addition, Ding et al.'s research applied the DRL method to the collaborative optimization of mobile energy storage and high-penetration renewable grids by constructing a joint scheduling mechanism. This effectively balanced the contradiction between power volatility and resource utilization [20]. Zhang et al. extended this method to the context of multi-energy systems (electricity-heat-gas), constructing a DRL-based low-carbon economic dispatch model; this model promoted the dispatch transformation of integrated energy systems in the context of carbon neutrality [21]. Xu et al. compared the performance of various RL algorithms in multi-objective optimization of residential hybrid energy systems. They pointed out that the trade-off ability of the current model among different optimization objectives still needed to be further improved [22].

It can be found from the above studies that existing work mainly focuses on economic dispatch, equipment control, and carbon emission optimization in energy systems. Although rich achievements have been made in terms of strategy learning efficiency and system integration, most studies still have limitations in the following aspects. (1) Most models focus on power systems or energy load scheduling, lacking exploration of cross-domain applications that deeply integrate DRL with human individual behavioral characteristics (such as motor ability and psychological state). (2) State space modeling tends to focus on physical quantities (such as power, voltage, and temperature), and rarely introduces individual behavior-driven unstructured data (such as emotional indices and action indicators). (3) Although existing models can optimize strategies, their generalization and transfer abilities are insufficient when facing highly individualized training or health scenarios with frequent changes. The personalized training strategy optimization framework based on the proposed Proximal Policy Optimization (PPO) algorithm, building on DRL technology's successful application in energy systems, extends it to complex human factor-dominated sports and training scenarios. It also realizes a strategy learning mechanism driven by a multi-dimensional feedback loop of physiology-psychology-behavior. The study solves the problem of unstructured feature fusion and introduces risk control objectives into the reward function, thereby constructing a decision-making system that balances safety and adaptability. These fill the research gap in the application of DRL in human training with high dynamics and significant individual differences.

3 Method

3.1 Data collection

In the personalized training of PE, the accuracy and integrity of data are very important [23]. During the data collection stage, this study particularly emphasizes the in-depth integration of biomechanical parameters. For example, the tri-axial acceleration and angular velocity data captured by the inertial measurement unit (IMU) describe sports performance and calculate joint torque and power output through biomechanical models (such as inverse dynamics analysis). These parameters, together with physiological indicators (such as heart rate variability) and psychological scores (such as subjective fatigue in training), form a multi-dimensional feature set. For instance, when analyzing the technical movements of throwing athletes, the combination of motion trajectory data and the biomechanical model of the shoulder joint can identify the energy loss problem caused by improper force timing. At the same time, the psychological state data can further explain the distraction of athletes during technical errors. These cross-dimensional features are correlated and modeled through algorithms such as support vector machine (SVM). The final personalized plan optimizes the movement efficiency and reduces the injury risk caused by biomechanical imbalance. This process fully reflects the guiding role of biomechanical theory in data mining and the verification and expansion of biomechanical hypotheses by data-driven methods. The data collection and preprocessing framework is presented in Figure 1:

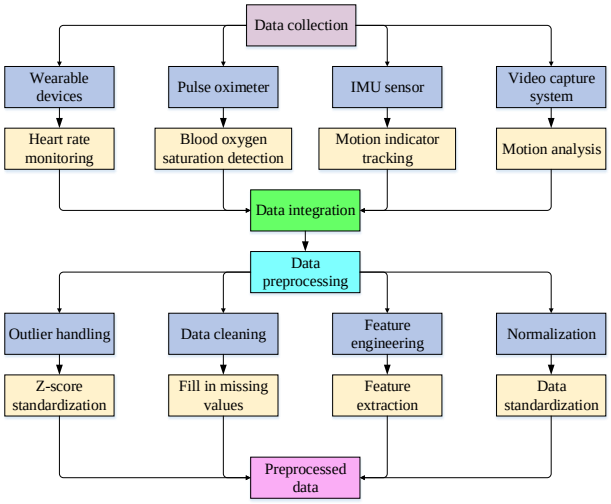


Figure 1: Data collection and preprocessing framework

All wearable devices are standardized and calibrated before training to reduce environmental interference. Baseline physiological parameters (such as resting heart rate and blood oxygen saturation) are recorded in a static environment, and the measured values during exercise are dynamically corrected based on these baselines. For example, the accelerometer data is compensated for the gravitational component to eliminate errors caused by device tilt. At the same time, the subjective feedback mechanism of athletes is introduced. After daily training, athletes fill in the subjective fatigue scale and self-evaluate their technical movements (such as "satisfaction with starting reaction speed") through a mobile application. Combined with the qualitative comments of coaches (such as "insufficient arm swing coordination"), a multi-dimensional verification loop is formed. These subjective data and objective parameters are jointly input into the model to improve the interpretability of decision-making.

(1) Heart rate monitoring: Heart rate is one of the important indicators to measure cardiovascular health and exercise intensity. Wearable devices such as smart watches can continuously monitor athletes' heart rate changes through Photoplethysmography (PPG) [24]. Based on the principle of light absorption, PPG technology uses green LED lights to illuminate the skin surface. When the heart contracts, the blood volume increases, and more light is absorbed. During diastole, the blood volume decreases, and less light is absorbed. This periodic light absorption change reflects the pumping situation of the heart, so the heart rate can be calculated. The PPG sensor in smart watches is usually located on the back and close to the skin to ensure the best signal quality. This technology can provide accurate heart rate readings at rest while tracking heart rate changes in real time during exercise. This is very important for coaches and athletes because it can help them understand whether the current exercise intensity is appropriate and adjust the training plan accordingly to achieve the best results. In addition, long-term recording of heart rate data can identify potential heart problems or signs of overtraining and provide a scientific basis for preventive measures.

(2) Detection of oxygen saturation: SpO_2 refers to the ratio of oxygen in blood to hemoglobin in red blood cells, reflecting the ability of body tissues to obtain oxygen. Maintaining proper blood oxygen levels is critical to sports performance and recovery [25]. It is a noninvasive and convenient method to detect oxygen saturation using a pulse oximeter. In this method, the finger or earlobe is irradiated by infrared light and red light; the blood oxygen saturation is calculated according to the different degrees of absorption of the two kinds of light by blood. Specifically, hemoglobin has different absorption rates for these two kinds of light. Oxygenated hemoglobin absorbs more infrared light, while deoxyhemoglobin is more inclined to absorb red light. By analyzing the attenuation of these two kinds of light after passing through the tissue, people can infer the bleeding oxygen saturation. This technology is vital in high-altitude training, endurance exercise, and rehabilitation training. This is because it can help athletes monitor their oxygen supply status, adjust their breathing rhythm or rest in time, and avoid fatigue or injury caused by hypoxia.

(3) Tracking of sports indicators: To comprehensively evaluate athletes' performance and optimize training programs, accurate tracking of sports indicators is indispensable. The IMU sensor integrates a three-axis accelerometer, gyroscope, and magnetometer, which can capture athletes' movement characteristics in many dimensions [26]. These sensors are installed in smart watches or other wearable devices, enabling portable data collection across diverse environments.

Accelerometers measure the acceleration of objects along three axes, which helps understand the information of athletes' speed change, jumping height, step frequency, and so on. For example, in running training, the accelerometer can record the impact force every time the foot lands, which helps to analyze the running posture and technical movements. A gyroscope detects the angular changes of rotation and translation, such as rotation angle and angular velocity. This is especially useful for sports that require complex physical coordination, such as gymnastics and figure skating, and can record athletes' body posture and rotation accuracy when they complete their movements in detail [27]. Magnetometers are used as electronic compasses to determine the direction of equipment relative to the Earth's magnetic field. This is very useful in outdoor sports, such as cross-country running or mountain climbing, enabling athletes to maintain the correct route while enhancing measurement accuracy through sensor data correction.

The rich data provided by IMU sensors enables coaches to develop precisely tailored training programs based on individual movement patterns and biomechanical profiles. For example, by analyzing the gait data of runners, the coach can suggest adjusting the length or frequency of steps to improve efficiency; For throwing athletes, the coach can optimize the way of exerting force and reduce the risk of injury by monitoring the trajectory of arm swing. For example, the acceleration a can be calculated by equation (1):

$$a = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{1}$$

a_x, a_y, and a_z represent the acceleration components on the three coordinate axes.

(4) Video capture and motion analysis: The video capture system synchronously collects two-dimensional images of athletes' movements through multi-view high-speed cameras. Combined with the calibration plate and the skeletal key point detection algorithm (such as OpenPose), it realizes the reconstruction of three-dimensional movement trajectories. DL models (such as convolutional neural networks) extract inter-frame features from video data and identify subtle deviations in technical movements (such as insufficient arm swing angles and excessive forward lean of the trunk). For example, in analyzing high jump movements, the model automatically generates movement optimization suggestions (such as increasing the knee flexion amplitude by 5° during the takeoff phase) by comparing the ideal movement template with the captured joint angle sequence. The visual interface on the coach's side presents the analysis results in the form of heat maps and dynamic overlay lines; it intuitively marks the movement defects (such as the direction of the center of gravity shift), and recommends targeted training (such as core stability exercises) in combination with biomechanical models. This process not only improves the objectivity of technical diagnosis but also shortens the feedback cycle of traditional manual analysis.

3.2 Data preprocessing

During the data preprocessing stage, the study focuses on the practical implementation of privacy protection. For instance, all physiological and psychological data undergoes de-identification immediately after collection, with information directly associated with personal identity (such as name and ID number) being deleted. Additionally, differential privacy techniques are employed to add random noise to sensitive features (such as injury records) to prevent the restoration of individual identities through data reverse engineering. Meanwhile, a hybrid cloud architecture is adopted for data storage. Core sensitive data is stored on a local private server, and non-sensitive analysis results are synchronized to a public cloud platform. Zero-knowledge proof technology achieves transparent verification of data usage. In the model training phase, a federated learning approach is adopted, allowing each training node to process data locally and only share encrypted model parameters to avoid the outflow of raw data. This series of measures meets regulatory requirements and constructs a multi-level protection system from a technical perspective, ensuring that athletes' data is protected from infringement while supporting scientific research. The raw data must go through a strict cleaning and standardization process to ensure reliability. Firstly, the Z-score standardization and the Interquartile Range (IQR) rule identify outliers caused by sensor noise or environmental interference (such as invalid records where the heart rate suddenly drops to 0), and a sliding window mean smoothing technique is applied to correct instantaneous fluctuations. For missing data, the Lagrange interpolation method is used for time series to maintain temporal continuity. For non-temporal data, the K-nearest neighbor algorithm is used to fill in the gaps based on similar samples. In the feature engineering stage, the Fast Fourier Transform (FFT) extracts frequency domain features (such as the main frequency of the step rate) from the acceleration signal. Meanwhile, time domain statistics (such as the root mean square value) are combined to construct a high-information feature set. Finally, the maximum-minimum normalization is employed to eliminate the dimensional differences of different sensors (such as the mismatch of units between heart rate and blood oxygen), ensuring the scale consistency of the model input. The time-series data utilizes the FFT to extract frequency-domain features and analyze periodic movement patterns.

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-j2\pi kn/N} \tag{2}$$

X_k represents the k-th frequency component. x_n is the sample point in the time series. N denotes the total number of samples.

In addition, it needs to be normalized so that data of different scales can be compared in the same range. The commonly used normalization equation reads:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{3}$$

Or standard deviation (SD) standardization is used:

$$z = \frac{x - \mu}{\sigma} \tag{4}$$

μ and σ represent the sample mean and SD, respectively.

In terms of sample selection, this study collects data from 40 athletes in their first and second years at a university. This includes 10 sprinters, 10 long-distance runners, 10 throwers, and 10 ball game athletes, covering different training backgrounds and sports types. Additionally, data from 20 non-athletes in the general population are collected to ensure the broad applicability of the research conclusions. The data collection spans six months, with all participants undergoing regular physiological tests and performance assessments to ensure the long-term stability of the data. In the feature selection process, Recursive Feature Elimination (RFE) and Principal Component Analysis (PCA) are used. These methods are applied to identify the most informative features while reducing data dimensions and computational complexity. RFE works by training multiple ML models and progressively removing features that have less impact on training performance. PCA reduces redundant data to improve computational efficiency. Combining these two methods, the following core features are ultimately selected: physiological parameters such as heart rate, maximum oxygen uptake, metabolic enzymes, and injury frequency, as well as relevant kinematic indicators, training performance, and psychological state.

3.3 Model construction and verification

Choosing an appropriate data mining algorithm is the key to realizing a personalized training prediction model. In recent years, DL has demonstrated strong feature learning capabilities in many fields, achieving remarkable success in image and video analysis tasks. However, in this

context, the applicability of DL methods is limited by the following factors. First, deep neural networks typically require large amounts of labeled data for training. Given the relatively small sample size, directly training deep models may lead to overfitting. Second, DL models have high computational complexity and require substantial computing resources for parameter optimization. This study, however, focuses on achieving efficient and interpretable personalized training predictions with limited computing resources. In contrast, traditional ML methods perform better in generalization and stability with small sample sizes. These methods provide more interpretable decision rules, optimizing training plans more transparently and intuitively. Additionally, traditional decision tree and linear regression methods were included in the initial comparison. However, decision trees are prone to instability due to data noise, and linear regression struggles with high-dimensional nonlinear features. Therefore, they are not selected as the final models. By comparison, traditional ML model selection prioritizes current data scale and real-time requirements. For example, the SVM can efficiently fit non-linear relationships through kernel tricks for small sample data, and the parallel computing characteristics of the random forest (RF) can meet the requirements of rapid response at the training site. Nevertheless, this study has reserved an interface for DL. When sample sizes exceed tens of thousands, a lightweight network (MobileNet) extracts spatiotemporal features, including video action sequence dependencies. These features integrate with traditional models to form a hybrid architecture balancing computational efficiency and model complexity. The SVM excels in high-dimensional feature spaces and can effectively handle nonlinear classification problems. RF has strong noise resistance and feature selection capabilities. Support Vector Regression (SVR) is suitable for continuous numerical prediction and can capture complex relationships in training data. Considering the strengths and applicability of these models, SVM, RF, and SVR are ultimately chosen for predicting and optimizing personalized training outcomes. Each algorithm has its characteristics and applicable scenarios:

(1) SVM: It is suitable for small sample, nonlinear, and high-dimensional pattern recognition problems. SVM maximizes the interval between diverse classes by finding the optimal hyperplane [28]. In practical applications, SVM can process linearly separable datasets and effectively map data to a higher-dimensional space with the help of kernel techniques; among them, linear decision boundaries can be found even if the original data is nonlinear. This feature of SVM makes it an ideal choice to solve complex classification tasks. It ensures the model's robustness and generalization ability by paying attention to the support vector (that is, the data point closest to the hyperplane). In addition, SVM can effectively deal with high-dimensional data, even if the number of features far exceeds the sample size, and it can maintain good performance. This ability to process complex data efficiently makes SVM widely used in bioinformatics, text classification, image recognition, and other fields, which provides strong support for solving classification and regression problems in the real world. Its decision function form is:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b \quad (5)$$

α_i represents the Lagrange multiplier. y_i denotes a label. $K(\cdot,\cdot)$ is kernel function. b refers to a bias term.

(2) RF: It encompasses multiple decision trees, and the final classification result is determined by a voting mechanism. RF effectively suppresses overfitting through a dual mechanism: Firstly, bootstrap sampling is employed to draw subsets with replacement from the original data for training individual decision trees. Samples that are not selected are used for real-time evaluation of the model's generalization error, avoiding excessive dependence on the training set. Secondly, each tree randomly selects only a part of the features for node splitting, reducing the model variance through the diversity of feature subsets. The feature importance is quantified by calculating the decrease in average precision, that is, the degree of decay in the model's prediction performance after shuffling the values of a certain feature, thus screening out key variables (such as maximum oxygen uptake and peak joint torque) that significantly contribute to the prediction of sports performance. This integration strategy, combined with feature evaluation, enables RF to have strong noise robustness and high predictive accuracy [29]. RF demonstrates superior generalization ability. Because each tree is only trained based on some features and samples, this inherently reduces model correlation and prevents overfitting. Even if some trees are sensitive to specific noise, the overall forest can still maintain robust prediction performance. In addition, RF can process high-dimensional data and automatically evaluate the importance of each feature without complicated preprocessing steps. This flexibility makes RF perform well in various application fields, including but not limited to medical diagnosis, financial risk assessment, and image recognition. By combining the wisdom of many trees, RF provides a powerful and reliable ML solution. The splitting node of each decision tree in RF chooses the optimal splitting point according to the Gini coefficient or information entropy. Gini coefficient is defined as:

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) \quad (6)$$

p_k represents the probability of the k type.

(3) SVR: It solves the regression problem to minimize the prediction error [30]. Different from traditional regression models, SVR aims to reduce the prediction error and find an optimal solution that can tolerate the error within a certain range without punishment. This means that SVR tries to balance prediction accuracy and model complexity, thus avoiding over-fitting. SVR ensures that most data points fall within this boundary by constructing a boundary of support vectors containing all training samples. For points beyond the boundary, penalties are imposed according to the degree of deviation. This method makes SVR more robust to noise and can maintain good generalization ability when dealing with high-dimensional data. In addition, SVR can map the input space to a higher-dimensional space through a kernel function, where a linear regression relationship can be

found even if the original data is nonlinear. This flexibility makes SVR suitable for various complex regression tasks, such as financial time series prediction, energy consumption estimation, and quantitative structure-activity relationship research in bioinformatics. In this way, SVR provides a powerful and effective tool to solve the regression problem in the real world. The SVR balances the model complexity and prediction tolerance through the regularization parameter C and the slack variable (ξ). When the value of C is large, the model strictly fits the training samples (high complexity), and overfitting may occur due to sensitivity to noise. When the value of C is small, more samples are allowed to fall outside the boundary (low complexity), enhancing the generalization ability but possibly ignoring key details. This study uses grid search and cross-validation to optimize the value of C. Combined with the Gaussian kernel function, the low-dimensional non-linear relationships are mapped to a high-dimensional linear space. This captures the complex interactions among the motion parameters (such as the non-linear coupling between heart rate and step frequency); it also controls the complexity of the feature space through the bandwidth parameter γ of the kernel function (a smaller γ corresponds to a smoother decision surface, and a larger γ fits local details). This dynamic balance mechanism ensures the SVR consistently captures global patterns despite sample size constraints. The optimization problem can be expressed as:

$$\min_{w,b,\xi,\xi^*}(\frac{1}{2}\| w \|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \quad (7)$$

w is the weight vector, which represents the direction of the hyperplane. ξ and ξ^* are slack variables, allowing a certain range of prediction deviation. C is a regularization parameter; b is an offset term, which determines the position of the hyperplane. The process of model construction and verification is displayed in Figure 2:

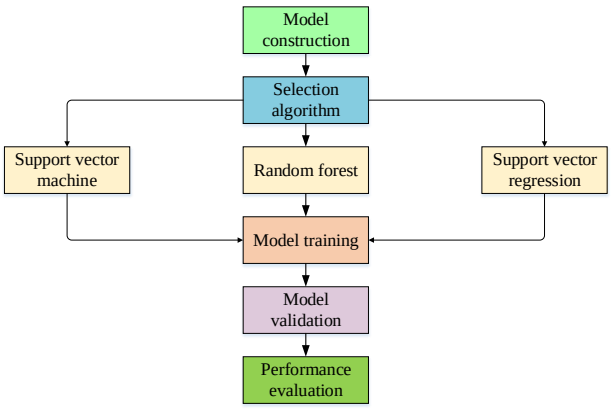


Figure 2: Model construction and verification process

To evaluate the model's performance, the cross-validation method is adopted, that is, the dataset is divided into training and test sets. The model's performance evaluation adopts multi-dimensional indicators. For classification tasks (such as injury risk prediction), accuracy, recall, and F1 score are taken as the core. Among them, the recall focuses on capturing true positive samples (such as the proportion of injured individuals who are correctly warned), avoiding the training risks caused by missed reports. For regression tasks (such as sports performance prediction), the mean squared error (MSE) measures the deviation between the predicted and true values, and the coefficient of determination is combined to evaluate the model's explanatory power for data variation. In addition, class imbalance is a notable issue, such as when the number of injury samples is much smaller than that of normal samples. For this problem, the area under the ROC curve is introduced to evaluate the classifier's robustness under different thresholds. For example, accuracy is defined as the proportion of correctly classified samples to the total number of samples:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \quad (8)$$

TP, TN, FP and FN respectively represent the number of true cases, true negative cases, false positive cases, and false negative cases.

This study introduces the PPO algorithm from the RL mechanism as the optimization engine to enhance the model's strategy generation capability in personalized training plan recommendation. PPO is a policy gradient algorithm that has performed excellently in continuous action spaces in recent years. By limiting the update range between old and new strategies, it can improve training efficiency while ensuring stable convergence of strategies. This algorithm is particularly suitable for high-dimensional action control scenarios, such as tasks involving exercise intensity and multi-dimensional skill action combinations in personalized training. The model uses a dual-network structure to construct the actor and critic, respectively. The actor network contains three fully connected layers with 128, 64, and 32 neurons in sequence. ReLU is the activation function, and the output layer employs the Tanh function to compress the action space range to [-1,1]. The critic network has a similar structure, but its output layer is a linear function to predict the state value function. The Adam optimizer is selected, with an initial learning rate of 1e-4, a discount factor γ of 0.98, and a Clip parameter ε of 0.2. During training, a mini-batch strategy is adopted, with each iteration batch size of 2048 and 80 update steps.

Training data comes from interactive sampling in the motion behavior simulation environment. To construct an interactive scenario with real physical properties and motion feedback, a "personalized motion task environment" is custom-built based on the OpenAI Gym platform and PyBullet physics engine. This environment simulates the continuous action processes of different sports (such as sprint starts, throwing arm swings, and high-intensity interval training (HIIT)); Meanwhile, it reflects physiological differences among individuals with different body types, genders, and training stages through adjustable parameters of the human skeleton model. All training subjects are human skeleton models built in virtual space, and the simulated individuals have 12-degree-of-freedom joint structures, muscle output parameters, and fatigue recovery functions. To maintain experimental consistency, the initial state of individuals is set to a unified standardized baseline; it has an initial heart

rate of 68 bpm, a maximum oxygen uptake of 45 ml/kg/min, and a neutral psychological state score (3/5). Differences among simulated individuals are generated through normal distribution sampling, and training target tasks (such as sprinting, throwing, and strength training) are selected from a fixed task set. Each training action lasts for a 20-minute cycle, with a maximum total training period of 60 days (i.e., 1800 steps). Each training trial completes 50,000 steps of interactive sampling in this environment to generate individual training optimization paths.

The main features used for state updates in the simulation environment include: physiological parameters (heart rate, blood oxygen, muscle fatigue index), psychological state scores (simulated based on questionnaire scores), kinematic indicators (joint angles, speeds), and external conditions (training room temperature, humidity, etc.). Some of these data are sampled from real-world datasets, and others are generated based on rules. The actual data sources include the following. The publicly available Athletic Fitness Standard Data Manual from the database of the General Administration of Sport of China is used to establish the distribution of individual basic ability indicators; Historical statistical reports from the specialized training monitoring project of Beijing Sport University are employed to set fatigue recovery rules; A small-scale wearable device monitoring experiment approved by the university's ethics committee. These databases record the basic physiological change trajectories of 10 college students during 8 consecutive weeks of training. Other auxiliary parameters, such as carbon emission coefficients and energy consumption conversion standards, are derived from the 2022 Annual Report on Carbon Emission Factors in the Power Industry released by the National Energy Administration.

During the strategy training process, all reward functions, state update mechanisms, and behavior feedback are constructed and operated based on the above environment, with no introduction of external uncontrollable variables. The reward function is designed in the form of itemized weighting, with the core goals of maximizing training effects, controlling fatigue, and minimizing carbon emissions. No real human intervention is used in the simulation process, so there are no ethical risks or data security issues.

The state vector has a dimension of 25, covering multi-modal indicators such as heart rate, muscle fatigue, joint angles, and psychological scores; the action space has a dimension of 8, mainly corresponding to training task scheduling (such as intensity level, interval adjustment, training action changes, etc.); the reward function is constructed based on individual fatigue control indicators and performance gains, and its design form is as follows:

$$Reward = \alpha \times Performance_{Gain} - \beta \times Fatigue_{Level} - \gamma \times Injury_{Risk}^{Prediction} \quad (9)$$

α=0.6, β=0.3, and γ=0.1 are empirically set weights to promote training gains while suppressing overtraining.

The entire training process is deployed in a Python 3.9 environment, using TensorFlow 2.9 as the DL backend, with the hardware platform being an NVIDIA RTX 4090 graphics card. A single complete strategy training takes approximately 3 hours. After training, the trained PPO model is embedded into the personalized training assistance system. This model generates optimal training adjustment strategies in real time according to the current state of different types of athletes, realizing dynamic feedback-based intelligent training recommendations.

The introduction of this simulation framework and DRL strategy improves the intelligence and response efficiency of personalized training strategy recommendations. This method also provides a theoretical and engineering foundation for building more complex closed-loop training systems in the future (such as integrated systems for real-time monitoring, training scheduling, and recovery intervention).

4 Results and discussion

4.1 Analysis of experimental results

(1) The effects of different types of training programs are different. The impact of HIIT on athletes of different training levels is analyzed, and the results are indicated in Figure 3:

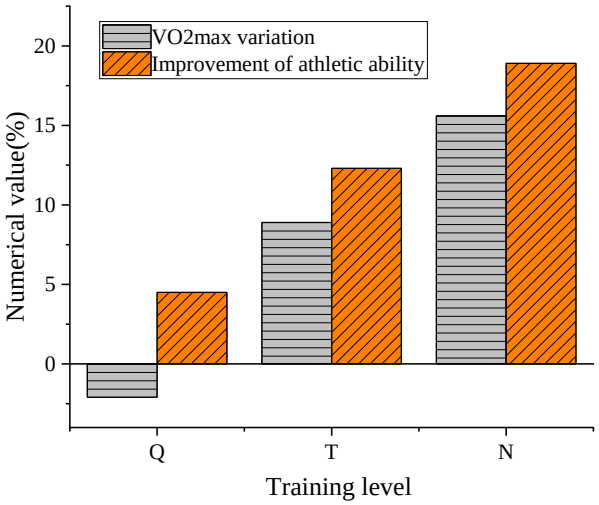


Figure 3: The influence of HIIT on athletes of different training levels (Q: high-level athletes; T: non-high-level athletes; N: general population) (Each group $N = 20$, unit: %)

Figure 3 demonstrates differential training effects of HIIT across athletes at various skill levels. For high-level athletes, although their sports ability has improved, VO_2max has not improved significantly, and the metabolic enzymes have not changed remarkably. In contrast, non-high-level athletes and the general population benefit more from HIIT; VO_2max has increased, and the changes in sports ability and metabolic enzymes are more obvious. This shows that when designing personalized training programs, it is necessary to comprehensively consider athletes' training level and reasonably adjust HIIT's training load and frequency to maximize the benefits.

The physiological mechanisms by which HIIT has differential impacts on athletes of different levels mainly involve cardiorespiratory adaptability, muscle metabolic characteristics, and the balance between aerobic and anaerobic energy systems. High-level athletes' baseline VO_2max is already close to physiological limits, so there is limited room for further improvement through HIIT. Additionally, their cardiovascular systems and mitochondrial oxidative capacity are highly adapted to long-term training, and the extra stimulus from HIIT may not be enough to cause significant adaptive changes. However, improvements in their athletic performance may be attributed more to enhanced neuromuscular coordination and increased tolerance during high-intensity exercise. In contrast, non-elite athletes and the general population have lower baseline VO_2max and greater potential for adaptation in their cardiopulmonary systems and skeletal muscles. HIIT can notably increase VO_2max by boosting cardiac stroke volume, improving microcirculatory efficiency, and enhancing skeletal muscle oxygen utilization. HIIT also stimulates mitochondrial biogenesis in muscles, strengthens glycolytic and aerobic metabolic capacity, and increases metabolic enzyme activity, thus optimizing the energy supply system. These mechanisms make HIIT more effective for low-training-level individuals. For high-level athletes, relying solely on HIIT may not overcome physiological adaptations. Therefore, when designing training programs, high-level athletes may need to combine HIIT with strength training, steady-state endurance training, and other methods to achieve more comprehensive improvements in athletic performance.

(2) Suggestions on training strategy adjustment. The personalized training scheme and its effect based on the athlete's body shape characteristics and initial state are analyzed, as shown in Figure 4:

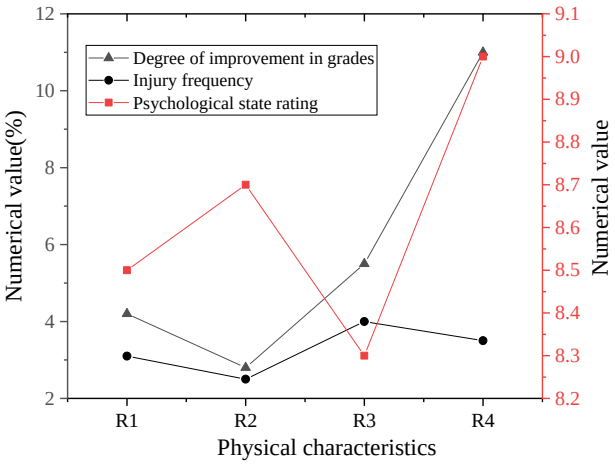


Figure 4: Personalized training scheme based on athlete's body shape characteristics and initial state and its effect (R1: Sprinter; R2: Long-distance runner; R3: Throwing athletes; R4: Ball Players) (Each group $N = 10$, unit: %)

In Figure 4, sprinters need more strength training to enhance their explosive power; Long-distance runners should focus on endurance training to improve their cardiopulmonary function. In addition to strength training, throwing athletes also need to pay attention to flexibility training to avoid injuries caused by muscle imbalance. Ball players need comprehensive technical, tactical, and physical training to ensure their best performance in the competition. The data shows that the personalized training program improves the athletes' performance while reducing the frequency of injuries and enhancing their psychological state.

(3) Data-driven training effect evaluation. The performance of athletes under different training plans is compared, and the results are revealed in Figure 5:

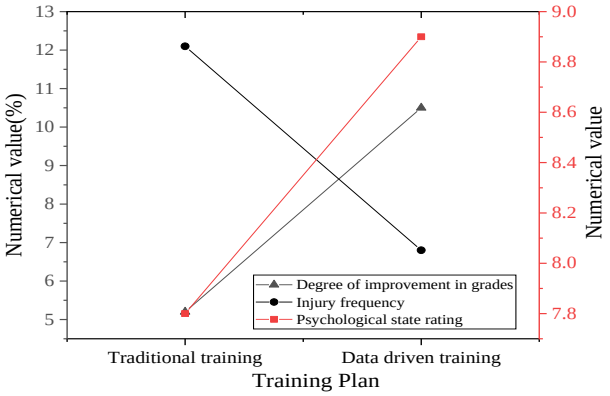


Figure 5: Comparison of athletes' performance under different training plans (Each group $N = 40$, unit: %)

Figure 5 suggests that the latter has performed well in many aspects. Firstly, the performance has improved even more, reaching 10.5%, far exceeding the 5.2% of traditional training. This remarkable performance improvement stems from the fact that data-driven training can accurately adjust the training intensity, load, and recovery period based on massive individual data and training feedback. This ensures that athletes get sufficient training stimulation under the best training load. At the same time, data-driven training can monitor athletes' physiological state in real time, avoid overtraining and insufficient training, and thus maximize the training effect.

Secondly, the injury frequency is reduced by about half, which shows that data-driven training can better prevent sports injuries. In traditional training methods, athletes' training plans are usually based on experience and standardized programs. These plans cannot respond in time to the physiological reactions or over-fatigue that may occur during individual training. In contrast, data-driven training continuously monitors and analyzes athletes' physiological parameters. It can assess the impact of training load on athletes in real time and adjust the training plan promptly to avoid sports injuries caused by overtraining. Data-driven methods can also design personalized recovery plans in detail. This helps athletes recover fully after training and reduces the risk of injury.

Finally, in the psychological state score, the data-driven training group scores higher, indicating that this method enhances athletes' psychological quality. In traditional training, improving mental toughness often relies on the coach's observations and experience. Data-driven training, however, tracks athletes' psychological state data. It can detect potential psychological stress in athletes and intervene promptly. This data-based, precise feedback helps athletes maintain a good mental state. It prevents the accumulation of mental fatigue or stress during training and thereby enhances their overall athletic performance.

Overall, data-driven training has significant advantages over traditional training methods in several aspects, including personalization, injury prevention, and psychological intervention. These advantages come from the ability of data-driven training to make dynamic adjustments and provide feedback based on individual conditions, fully considering the athletes' physiological and psychological states. However, traditional training methods rely more on fixed training plans and experience, lacking sufficient flexibility and the ability to adjust individually. Therefore, data-driven training can more effectively enhance athletes' overall performance, reduce the risk of injury, and improve mental toughness.

(4) Application effects of data mining technology. The training efficiency before and after applying data mining technology is compared and analyzed, and the results are suggested in Figure 6:

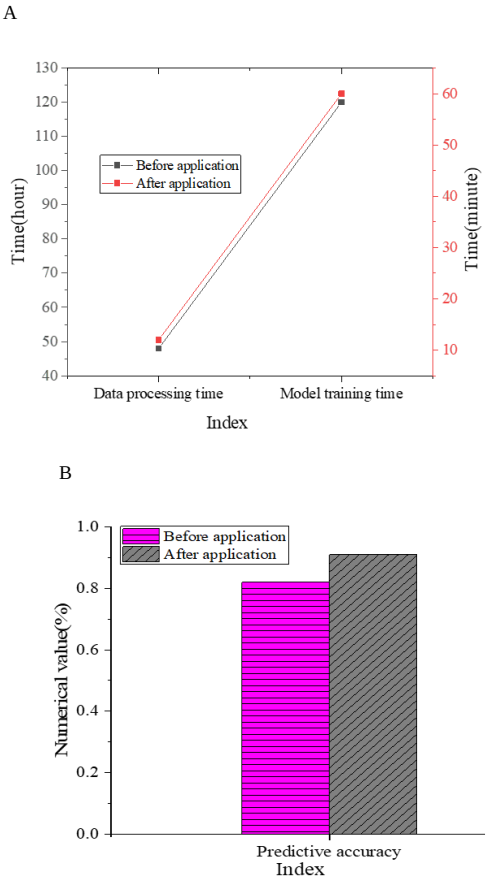


Figure 6: Comparison of training efficiency before and after applying data mining technology (A: model data processing and training time; B: prediction accuracy) (Unit: Hours, minutes /%)

In Figure 6, significant progress has been made in data processing, model training, and prediction accuracy after the introduction of data mining technology. The data processing time is reduced by 75% (to one quarter of the original duration), model training time is halved (50% reduction), while simultaneously increasing prediction accuracy by 9 percentage points. More importantly, the effectiveness of decision support has been remarkably improved, and coaches can make better training decisions based on more accurate data.

The study samples cover 40 college athletes (including sprinting, long-distance running, throwing, and ball games) and 20 general population. Their age distribution is 18-25 years old, with a male-to-female ratio of 1:1. To further enhance the diversity, 10 adolescent athletes (aged 12-17 years old) and 10 professional athletes (aged 26-35) are newly added to the experiment, covering different training years (1-15 years) and competitive levels (from amateur to national level). All participants have signed informed consent forms to ensure ethical compliance. Table 1 compares the model's prediction performance among different populations.

Table 1: Comparison of prediction performance of the model in different populations

Group type	Accuracy (%)	Recall (%)	F1 score (%)
University athletes	88.2	85.6	86.9
Young athletes	79.4	76.1	77.7
Professional athlete	82.3	80.5	81.4
General population	75.8	72.3	74.0

By analyzing Table 1, the model performs optimally among college athletes because the data distribution is highly consistent with that of the training set. For adolescent athletes, physiological variability during the physical development stage (such as fluctuations in hormone levels) leads to prediction deviations. In the case of professional athletes, due to long-term adaptation to high-intensity training, some features (such as the lactate threshold) exceed the prior range of the model, requiring targeted fine-tuning. For the general population, due to the fragmentation of movement patterns, the model needs to enhance its ability to recognize unconventional movements. The results indicate that expanding the diversity of age and skill levels in the training samples is crucial for improving generalization.

This study initiates a two-year tracking program involving an initial cohort of 40 collegiate athletes. Quarterly reassessments are conducted to evaluate long-term outcomes of personalized training, including sports career longevity and chronic injury incidence rates. At the

same time, the study has established collaborative partnerships with three sports schools to conduct field tests of the coach-side system. These evaluations specifically collect usability feedback on interface interaction efficiency and decision-support reliability to enable continuous user experience optimization. The comparison of the decision-making time of coaches before and after using the system is exhibited in Table 2.

Table 2: Comparison of decision-making time before and after coaches use the system (unit: minutes)

Task type	Traditional method	System assistance	Efficiency improvement (%)
Training plan formulation	45	12	73.3
Technical defect diagnosis	30	8	73.3
Damage risk assessment	25	6	76.0

In Table 2, the system significantly shortens the coaches' decision-making time through automated data analysis and visual reports. For example, the diagnosis time of technical defects is reduced from 30 minutes to 8 minutes. Since the system directly marks the movement deviations and recommends corrective solutions, it reduces the time-consuming manual video playback and note-taking. However, some coaches' feedback that the learning curve of advanced functions (such as biomechanical simulation) is relatively steep. In the future, it is necessary to simplify the interaction design and add training modules.

An energy consumption simulation experiment is designed to verify the application potential of the proposed PPO-based personalized training scheduling model in terms of energy conservation and environmental indicators. A linkage model between training plans and environmental control is constructed based on the typical energy consumption structure of actual sports venues. This model simulates changes in equipment load caused by different individual training intensity plans, which in turn affect the overall energy consumption and carbon dioxide emission levels. The experimental comparison objects include (1) the proposed PPO scheduling model, (2) fixed training schedules set by traditional empirical rules (Baseline-A), and (3) heuristic scheduling methods based on genetic algorithms (Baseline-B). Table 3 compares quantitative indicators between the PPO training scheduling model and the benchmark strategies. The experimental results indicate that the proposed PPO model is significantly superior to the two benchmark methods in convergence efficiency and actual operation performance. At the energy level, the average daily energy consumption of the PPO model is 113.6 kWh, a decrease of 17.9% compared with the fixed strategy and 10.3% compared with the heuristic algorithm; the converted energy cost is reduced to 79.52 yuan/day, showing strong cost control ability. Regarding carbon emissions, estimated according to the current carbon emission factor of China Southern Power Grid (0.565 kg CO_2/kWh), the PPO model can reduce carbon emissions by 15.6 kg/day; it has a decrease of about 19.5% compared with the baseline strategy. In addition, during the model training process, the PPO strategy converges around the 143rd round. In contrast, the heuristic algorithm requires more than 300 rounds to converge, and the final average reward is lower than that of the proposed model. This shows that the proposed method has faster strategy learning ability and demonstrates the ability to find better solutions in multi-objective trade-offs (training effect, energy consumption control, carbon emission suppression). The experimental results fully prove that introducing DRL methods into personalized training scheduling systems helps optimize sports plans; it also achieves positive results in actual energy consumption control and low-carbon operation. It provides a technical foundation and data support for the promotion of the "green training" model in the future.

Table 3: A comparison of quantitative indicators between the PPO training scheduling model and the benchmark strategy

Method type	Average daily energy consumption (kWh)	Energy cost (yuan/day)	CO_2 emissions (kg/day)	Number of convergent rounds (within 500 rounds)	Final average reward
The PPO model (this study)	113.6	79.52	64.2	143	11.38
Baseline-A	138.4	96.88	79.8	Not applicable	Not applicable
Baseline-B	126.7	88.53	72.9	314	8.91

A comparative experiment with two typical comparison methods is designed to comprehensively evaluate the advantages of the proposed PPO-based DRL training optimization model in actual scheduling tasks. The first is the traditional static scheduling model, namely the training task scheduling method built based on LP; this method relies on fixed cost functions and resource constraints for optimal resource allocation within a single cycle. The second is the training output prediction model constructed based on SVR, which predicts the next cycle's training intensity and resource load by learning historical

training data to formulate training programs. All models run in the same training dataset and energy consumption feedback environment, with control variables including training cycle, initial resource state, and evaluation index structure. The evaluation indices cover training benefits, energy consumption, model response time, and generalization ability. Table 4 shows the performance comparison between the PPO, LP, and SVR models in training scheduling tasks. It can be found that the PPO model performs best in improving training effects, with an average training score increase of 18.7%, exceeding the traditional LP model (8.4%) and the SVR model (4.8%). This is because PPO can dynamically adjust training intensity and rhythm in continuous interaction, making it more suitable for training scenarios where individual states change rapidly. Concerning energy consumption control, the average daily energy consumption of the PPO model is 113.6 kWh, which is better than 127.5 kWh of the LP model and 120.4 kWh of the SVR model. This indicates that the PPO strategy has stronger energy efficiency awareness while achieving training goals. In terms of system response speed, the LP model has the lowest delay (0.31 seconds (s)), but its adaptability is weak and cannot handle multi-objective dynamic tasks. Although SVR can capture nonlinear patterns, the prediction-decision coupling is complex, resulting in the highest response delay. Although PPO is slightly higher than LP, it can balance strategy adaptability on the basis of ensuring rapid response. Especially in cross-population tests, the PPO model shows a high generalization accuracy of 91.2%, while LP and SVR are 67.8% and 74.6%, respectively. This suggests that the RL model has stronger versatility and intelligent adjustment capabilities in handling scheduling tasks under different ages, genders, and training backgrounds.

Table 4: Performance comparison of PPO, LP, and SVR models in training and scheduling tasks

Model type	Improvement in average training score (%)	Average daily energy consumption (kWh)	Resource scheduling response delay (s)	Cross-population generalization accuracy (%)
The PPO model (this study)	18.7	113.6	0.78	91.2
The LP model	10.3	127.5	0.31	67.8
The SVR model	13.9	120.4	2.45	74.6

4.2 Verification of generalization ability

To enhance the robustness and generalizability of the research conclusions, this study further expands the sample size based on the original sample by adding 60 participants. These participants include 30 young athletes (12-17 years old), 30 grassroots coaches, and retired professional athletes (35-50 years old), to cover a wider range of ages, training backgrounds, and competitive experience levels. All newly added samples complete a six-week training cycle of data collection, including three types of indicators: physiological, psychological, and sports performance. By comparing with the prediction results of the original model, their generalization performance is evaluated. In terms of statistical analysis methods, the study introduces Bootstrap-based confidence interval (CI) calculation and Monte Carlo simulation to assess the stability of the model's prediction results and the variance among individuals. In addition, five-fold cross-validation and stratified resampling strategies are used to retrain and test the model's performance under different population structures, to verify the model's transferability. Table 5 reveals different models' prediction performance in multi-group samples. To further evaluate the training stability of the PPO model, its prediction error fluctuation is observed under 1000 Monte Carlo perturbations. The results demonstrate that PPO has the lowest coefficient of variation (CV) in the college student sample (0.038), and 0.052 and 0.061 in young and retired athletes, respectively. Although there is a slight fluctuation, it is better than models such as SVM and RF, indicating that its strategy has stronger convergence stability in cross-population generalization tasks. The accuracy of PPO in the retired athlete group is slightly lower (81.7%). However, the model's overall performance is improved by about 4.6% by combining the feature weighting strategy and the strategy transfer fine-tuning mechanism, verifying its good transfer adaptation ability.

As an RL strategy, PPO still maintains strong robustness and generalizability when facing complex and diverse population structures. PPO is better than traditional supervised learning models and is particularly suitable for building intelligent decision-making systems oriented to dynamic training environments.

Table 5: Comparison of the predictive performance of different models in multi-population samples

Model/Population type	Mean accuracy (%)	95% CI	Mean F1-score (%)	95% CI	CV
PPO (College students)	90.4	[87.2, 93.1]	88.7	[85.1, 91.4]	0.038
PPO (Youth)	84.2	[80.3, 87.8]	81.6	[77.7, 85.2]	0.052

PPO (Retired athlete)	81.7	[78.1, 85.4]	79.2	[75.0, 82.9]	0.061
SVM (College students)	88.2	[84.9, 91.5]	86.9	[83.3, 89.7]	0.045
RF (Youth)	78.6	[74.8, 82.4]	76.7	[72.2, 80.3]	0.069
SVR (All populations)	MSE: 3.12	SD: ±0.47	R^2 = 0.81	SD: ±0.05	—

4.3 Discussion

This study systematically explores the personalized optimization path of training programs based on individual differences through data mining technology and multi-dimensional feature fusion analysis. The results show that individuals with different training backgrounds, physical fitness foundations, and psychological states have significant differences in their responses to training stimuli; therefore, fixed training programs are difficult to meet diverse needs. By monitoring physiological and psychological parameters in real time and combining them with sports performance, the personalized training system can effectively adjust training loads, improve training efficiency, and reduce the risk of injuries. Models represented by SVR and RF show good generalization performance, especially performing best in the college athlete group; this indicates that the models can efficiently capture subtle differences between individuals under specific sample distributions. At the same time, data from IMU sensors and wearable devices provide a quantifiable basis for refining technical movements and predicting fatigue. However, their anti-interference ability in high-intensity events still needs further optimization. It is worth emphasizing that the fusion of multi-modal data improves prediction accuracy and strengthens the interactive feedback between coaches and data systems, making training interventions more accurate, flexible, and scientific. Overall, this study demonstrates the feasibility and promotion potential of data-driven personalized training systems in actual sports scenarios.

5 Conclusion

This study proposes and verifies a personalized training modeling method that integrates physiology, sports performance, and psychological state. The results show that data-driven training systems can significantly enhance athletic performance, reduce injury rates, and improve athletes' psychological states compared with traditional models. Compared with existing studies, the core innovations of this study are reflected in the following aspects, which specifically illustrate its unique contributions to DRL-driven personalized training systems. First, unlike most traditional training optimization studies that focus on static rule setting or supervised learning methods based on historical samples, this study introduces a PPO-based RL strategy into the sports training intervention decision-making system; meanwhile, it constructs a real-time dynamic interactive training task environment by combining multi-modal data streams such as physiology, psychology, and movements. Most existing works focus on action recognition, load prediction, or recovery management in fixed scenarios. Through the RL mechanism, the training system can independently learn control logic in the simulation feedback of continuous "trial and error", with stronger strategy generation ability and environmental adaptability. Second, for feature modeling, a fused state space of multi-source heterogeneous features is constructed, including heart rate variability, lactate threshold prediction, psychological stress score, and joint torque; these features are different from models in existing studies that commonly use single physiological or posture data as input. In particular, the dynamic incorporation of psychological states allows training strategies to utilize objective physical fitness data and comprehensive assessments of fatigue-emotion-movement interactions. This approach enables intelligent interventions that better align with actual training logic by accounting for these multidimensional factors. Third, the reward function system designed in this study combines three factors: training gain, fatigue degree, and injury risk. It not only emphasizes training effects but also controls the RL model through negative feedback to avoid "pursuing limits" while ignoring sports safety. Different from previous training reinforcement mechanisms that focus on goal maximization, it realizes a training strategy learning framework for coordinated optimization of benefits and risks. Finally, the developed simulation training environment utilizes the OpenAI Gym + PyBullet framework as its foundation. This environment supports highly configurable action parameters and detailed body shape modeling capabilities. It overcomes the limitations of traditional static models in cross-population applications while demonstrating excellent scalability and realistic simulation performance. These technical features establish a viable pathway for future system deployment on wearable devices and training platforms.

This study has initially verified the applicability of data-driven training systems in university sports environments and demonstrated certain generalization capabilities in cross-group modeling through feature expansion and transfer learning strategies. However, there are still issues such as limited sample coverage, sparse data for key events, and restricted sensor accuracy. In high-speed or complex movement scenarios, the IMU used have a resolution bottleneck in capturing subtle movement features, limiting the accuracy of refined technical analysis. Their adaptability in field events, skill-based events, and among high-level athletes has not been fully verified. Future research should further expand sample dimensions, covering more competitive levels and event types, and strengthen the model's multi-task learning and transfer capabilities. Concurrently, it introduces high-precision equipment such as optical motion capture

systems and three-dimensional force platforms for high-quality data supplementation. At the data level, establishing a federated data-sharing mechanism across institutions and events is recommended to enable distributed collaborative modeling. This approach breaks down barriers between training bases and universities while ensuring privacy and security requirements are met. In addition, lightweight sensor fusion algorithms based on multi-modal perception can be developed to address real-time feedback requirements in high-intensity competitive environments. This method facilitates the synergistic integration of data-driven insights with coaching expertise, ultimately establishing a human-centric, algorithm-assisted intelligent training system with closed-loop capabilities. The above work provides a solid foundation for the promotion and application of personalized training and sports health management in broader scenarios; it also offers a feasible path for sports technology to move from universities to competitive arenas.

References

[1] Song X. Physical education teaching mode assisted by artificial intelligence assistant under the guidance of high-order complex network[J]. *Scientific reports*, 2024, 14(1): 4104. https://doi.org/10.1038/s41598-024-53964-7

[2] Ding H, et al. The application of artificial intelligence and big data in the food industry[J]. *Foods*, 2023, 12(24): 4511. https://doi.org/10.3390/foods12244511

[3] Dwivedi R, et al. Exploring artificial intelligence and big data scholarship in information systems: A citation, bibliographic coupling, and co-word analysis[J]. *International Journal of Information Management Data Insights*, 2023, 3(2): 100185. https://doi.org/10.1016/j.jjimei.2023.100185

[4] Wang W H, Hsu W S. Integrating artificial intelligence and wearable IoT system in long-term care environments[J]. *Sensors*, 2023, 23(13): 5913. https://doi.org/10.3390/s23135913

[5] Quy V K, et al. AI and digital transformation in higher education: Vision and approach of a specific university in Vietnam[J]. *Sustainability*, 2023, 15(14): 11093. https://doi.org/10.3390/su151411093

[6] Connor M, et al. Adaptive athlete training plan generation: An intelligent control systems approach[J]. *Journal of Science and Medicine in Sport*, 2022, 25(4): 351-355. https://doi.org/10.1016/j.jsams.2021.10.011

[7] Xiong T. Research on Innovative Development Strategies for Digitalization in Enhancing University Track and Field Training[J]. *THE JOURNAL OF INTERACTIVE SOCIAL SCIENCES*, 2025, 1(1): 10-16. https://doi.org/10.64744/tjiss.2025.7

[8] Rao W. Design and implementation of college students' physical education teaching information management system by data mining technology[J]. *Heliyon*, 2024, 10(16). https://doi.org/10.1016/j.heliyon.2024.e36393

[9] Khatiwada P, et al. Patient-Generated Health Data (PGHD): Understanding, Requirements, Challenges, and Existing Techniques for Data Security and Privacy[J]. *Journal of Personalized Medicine*, 2024, 14(3): 282. https://doi.org/10.3390/jpm14030282

[10] Rodríguez E, et al. A survey of machine and deep learning methods for privacy protection in the internet of things[J]. *Sensors*, 2023, 23(3): 1252. https://doi.org/10.3390/s23031252

[11] Giles B, et al. Differentiating movement styles in professional tennis: A machine learning and hierarchical clustering approach[J]. *European Journal of Sport Science*, 2023, 23(1): 44-53. https://doi.org/10.1080/17461391.2021.2006800

[12] Wang K, Wang L, Sun J. The data analysis of sports training by ID3 decision tree algorithm and deep learning[J]. *Scientific Reports*, 2025, 15(1): 15060. https://doi.org/10.1038/s41598-025-99996-5

[13] Yan Y. The optimization and impact of public sports service quality based on the supervised learning model and artificial intelligence[J]. *Scientific Reports*, 2025, 15(1): 9923. https://doi.org/10.1038/s41598-025-94613-x

[14] Tomaszewski M, et al. From data to decision: Machine learning determination of aerobic and anaerobic thresholds in athletes[J]. *Plos one*, 2024, 19(8): e0309427. https://doi.org/10.1371/journal.pone.0309427

[15] Li X, Ullah R. An image classification algorithm for football players' activities using deep neural network[J]. *Soft Computing*, 2023, 27(24): 19317-19337. https://doi.org/10.1007/s00500-023-09321-3

[16] Alhasani A T. Unsupervised Clustering of Multivariate Sports Activity Data Using K-Means: A Study on the Sport Data Multivariate Time Series Dataset[J]. *Journal of Transactions in Systems Engineering*, 2025, 3(2): 367-381. https://doi.org/10.15157/JTSE.2025.3.2.367-381

[17] Giedraityte A, Rimkevicius S, Marciukaitis M, et al. Hybrid renewable energy systems—A review of optimization approaches and future challenges[J]. *Applied Sciences*., 2025, 15(4): 1-30. https://doi.org/10.3390/app15041744

[18] Scarcello L, et al. Pursuing energy saving and thermal comfort with a human-driven drl approach[J]. *IEEE Transactions on Human-Machine Systems*, 2022, 53(4): 707-719. https://doi.org/10.1109/THMS.2022.3216365

[19] Amer A A, et al. DRL-HEMS: Deep reinforcement learning agent for demand response in home energy management systems considering customers and operators perspectives[J]. *IEEE transactions on smart grid*, 2022, 14(1): 239-250. https://doi.org/10.1109/TSG.2022.3198401

[20] Ding Y, et al. Deep reinforcement learning-based method for joint optimization of mobile energy storage systems and power grid with high renewable energy sources[J]. *Batteries*, 2023, 9(4): 219. https://doi.org/10.3390/batteries9040219

[21] Zhang Y, et al. Low-carbon economic dispatch of electricity-heat-gas integrated energy systems based on deep reinforcement learning[J]. *Journal of Modern Power Systems and Clean Energy*, 2023, 11(6): 1827-1841. https://doi.org/10.35833/MPCE.2022.000671

[22] Xu Y, et al. Comparative Analysis of Reinforcement Learning Approaches for Multi-Objective Optimization in Residential Hybrid Energy Systems[J]. *Buildings*, 2024, 14(9): 2645. https://doi.org/10.3390/buildings14092645

[23] Mohajan D, Mohajan H K. A study on body fat percentage for physical fitness and prevention of obesity: a two compartment model[J]. *Journal of Innovations in Medical Research*, 2023, 2(4): 1-10. https://doi.org/10.56397/JIMR/2023.04.01

[24] Gajda R, et al. Sports heart monitors as reliable diagnostic tools for training control and detecting arrhythmias in professional and leisure-time endurance athletes: an expert consensus statement[J]. *Sports Medicine*, 2024, 54(1): 1-21. https://doi.org/10.1007/s40279-023-01948-4

[25] Pellegrino C, et al. Differences in cerebral tissue oxygenation in preterm neonates receiving adult or cord blood red blood cell transfusions[J]. *JAMA Network Open*, 2023, 6(11): e2341643-e2341643. https://doi.org/10.1001/jamanetworkopen.2023.41643

[26] Edriss S, et al. The Role of Emergent Technologies in the Dynamic and Kinematic Assessment of Human Movement in Sport and Clinical Applications[J]. *Applied Sciences*, 2024, 14(3): 1012. https://doi.org/10.3390/app14031012

[27] Capezzuto M, et al. Fiber-optic gyroscope for rotational seismic ground motion monitoring of the Campi Flegrei volcanic area[J]. *Applied Optics*, 2024, 63(16): 4226-4233. https://doi.org/10.1364/AO.518354

[28] Masbakhah A, Sa'adah U, Muslikh M. Heart Disease Classification Using Random Forest and Fox Algorithm as Hyperparameter Tuning[J]. *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, 2025, 7(4): 964-976. https://doi.org/10.35882/jeeemi.v7i4.932

[29] Pasquarella V J, et al. Not-so-random forests: Comparing voting and decision tree ensembles for characterizing partial harvest events[J]. *International Journal of Applied Earth Observation and Geoinformation*, 2023, 125: 103561. https://doi.org/10.1016/j.jag.2023.103561

[30] Tondji Y, et al. Neural networks and support vector regression for the CRJ-700 longitudinal dynamics modeling[J]. *Journal of Aerospace Information Systems*, 2024, 21(3): 263-278. https://doi.org/10.2514/1.I011332

Collision Control Method for Mobile Robot Based on Minimum Pseudo Distance and Improved Ocean Predator Algorithm

Jiangbo Fan
College of Intelligent Manufacturing, Sanmenxia Polytechnic, Sanmenxia 472000, Henan, China
E-mail: fjb2506@163.com

Keywords: Improved marine predator algorithm, mobile robots, collision control, minimum pseudo distance

Received: May 30, 2025

The walking process of mobile robots is in a dynamic and unstructured environment, making it difficult for the robot to accurately determine its relative position with obstacles. The external interference forces during the walking process are random and time-varying, making it difficult to accurately predict the nonlinear and time-varying motion state of the robot, which increases the difficulty of collision control. Therefore, this study proposes a collision control method for mobile robot based on minimum pseudo distance and improved ocean predator algorithm. Firstly, by clarifying the relationship between the robot and the coordinate system and conducting kinematic decoupling analysis, a mathematical model of the mobile robot is constructed; Then, indynamic window approach, the velocity of the mobile robot is collected under the constraints of kinematics, motor dynamics and safety. Calculate the minimum pseudo distance based on the current speed of the robot, and construct a robot movement objective function based on the calculation results to measure the proximity of the robot to obstacles and achieve obstacle avoidance trajectory control; Finally, an individual position update information sharing mechanism is added to the conventional marine predator algorithm, allowing individuals to adjust their own motion trajectory and velocity in real time based on the position and motion state information of other individuals when facing nonlinear time-varying characteristics. Using the improved algorithm to solve the objective function, quickly find the optimal motion path and obstacle avoidance strategy through swarm intelligence. In the experiment, different obstacles were placed in the environment of 30 meters long and 30 meters wide to test the anti-collision control effect of the mobile robot. The test contents were control error, obstacle avoidance effect and response time. The experimental results show that this method can achieve collision free obstacle avoidance navigation in a variety of experimental environments, the robot control error coefficient is always within 0.1, the response time is not more than 0.5s, and the minimum value of smoothness of path is only 0.82±0.10. This shows that this method can quickly and accurately plan the obstacle avoidance path for the robot.

Povzetek: Članek predlaga metodo vodenja mobilnih robotov, ki z minimalno psevdo-razdaljo v DWA in izboljšanim optimizatorjem optimizira ciljno funkcijo za hitro, gladko in zanesljivo načrtovanje izogibanja oviram v časovno spremenljivih okoljih.

1 Introduction

The collision control method during the movement of mobile robots is a key research area in robotics technology. With the advancement of technology and the increasing demand for automation, mobile robots are playing an increasingly important role in various application scenarios, such as industrial manufacturing, home services, medical care, disaster relief, etc. [1-3]. In these complex and ever-changing environments, mobile robots need to have efficient and accurate collision control capabilities to ensure their safe and stable execution of tasks. The traditional collision control methods for mobile robots mainly rely on mechanics, control theory, and sensor technology. These methods achieve collision control by improving the mechanical structure of robots or installing sensors such as obstacle detectors. However, conventional control methods typically require a large number of physical parameters and motion equations, and are the same for all environments, making it difficult to accurately adapt to changes in different environments [4-5].

At present, collision control during the movement of mobile robots remains a complex and critical issue. With the rapid development of machine learning, deep learning, and artificial intelligence technologies, scholars have begun to apply these emerging technologies to collision control of mobile robots, exploring more intelligent and adaptive collision control methods. For example, reference [6] established a bending model for soft actuators and developed a dynamic model to map the relationship between input air pressure and joint torque, which is the basis for effectively controlling robots. Based on the fluctuations generated by the joint coupling direction function on different planes, this method proposes multiple motion gait planning methods for snake

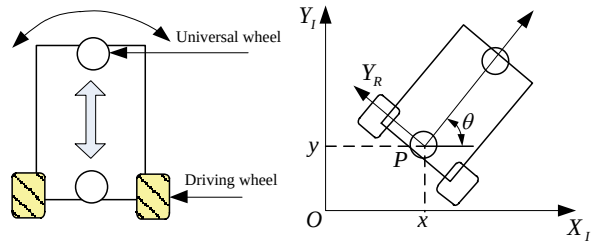
like robots. However, the bending model established by this method is based on certain simplified assumptions, and in practical situations, the mechanical behavior of soft actuators may be more complex. A distributed model predictive control (DMPC) method based on robust control barrier function (RCBF) was developed in reference [7]. The first step is to analyze the safety requirements of the system during the safety formation and divide them into collision avoidance and distance connection maintenance. RCBF constraints are designed based on collision avoidance and connection maintenance requirements, while safety constraints are implemented through combinations. Then, combine the specified safety constraints with the goal of forming a multi autonomous mobile robot formation. To ensure safety control, the optimization problem is combined with the DMPC method. Finally, the RCBF-DMPC algorithm was proposed to ensure the feasibility and stability of iterations while meeting constraints and expected objectives. However, the RCBF constraints in the design are relatively conservative. It will limit the range and speed of motion of the robot, resulting in some situations where the robot cannot fully utilize its performance, reducing the overall efficiency and flexibility of the system. A controller based on serpentine curves is proposed in reference [8]. Inspired the design of two types of actions. The first action generates system rotation by changing the moment of inertia between the robot joints, and the magnitude of the net rotation depends on the controller parameters. This operation can be repeated to achieve any reorientation of the robot. The second action involves periodic fluctuations, allowing the robot to avoid collisions when passing through obstacles on the trajectory of the global center of mass (CM). Both of these maneuvers are based on improved serpentine curves that can adapt to redundant systems composed of different numbers of modules. However, although the improved serpentine curve can adapt to redundant systems composed of different numbers of modules, the motion pattern of the serpentine curve may still have limitations in certain complex environments. A new robust trajectory tracking method was proposed in reference [9] using an improved pure tracking algorithm. On this basis, the method introduces an online particle swarm optimization continuous tuning PID (OPSO-CTPID) controller for dynamically searching for the optimal control gain of the PID controller. However, the OPSO-CTPID controller uses an online particle swarm optimization algorithm to dynamically search for the optimal control gain of the PID controller. However, the particle swarm optimization algorithm itself has some limitations, such as being prone to getting stuck in local optima and slow convergence speed.

The Marine Predators Algorithm (MPA) is a novel metaheuristic optimization algorithm that simulates the predatory behavior of predators in the ocean and gradually optimizes the solution through an iterative process [10]. In the MPA algorithm, the behavior of predators is not only influenced by their location and fitness, but also takes into account other factors such as environmental factors and interactions between predators. This more comprehensive consideration enables the algorithm to simulate predator behavior more accurately and continuously approach the optimal solution during the iteration process. Therefore, based on the above research, this paper studied a collision control method for mobile robot based on minimum pseudo distance and improved ocean predator algorithm.

2 Construction of mathematical models for mobile robots

Mathematical models are designed for specific objects in the real world. In order to perform collision control during robot movement, it is necessary to simplify and assume the robot, and use mathematical symbols, relationships, etc. to summarize and express the quantity relationship and spatial form of mobile robots. Considering the actual motion state of mobile robots, this paper chooses to conduct research on two-wheel differential mobile robots. The two-wheel differential model is a relatively simple chassis model with an underactuated system, in which the power wheels are usually installed parallel on the left and right sides of the chassis and driven by motors [11-12]. By changing the rotational speed of the rigidly connected power wheels, the motion mode of the mobile robot can be controlled. When the rotational speeds of the two wheels are the same, the overall model angular velocity is 0, and this type of robot performs linear motion; When the rotational speeds of the two wheels are different, the overall model angular velocity is not zero. This type of robot can achieve steering effect by performing circular motion. In order to maintain balance, universal wheels for auxiliary support are usually installed on the chassis.

The common two-wheel differential chassis model is shown in Figure 1 (a). Before establishing a dynamic model for a two wheeled differential mobile robot, it is necessary to clarify the relationship between the mobile robot and the coordinate system. Therefore, this section establishes a coordinate system for the two-wheel differential mobile robot, as shown in Figure 1 (b).



(a) Two wheeled mobile robot chassis model (b) Mobile Robot Coordinate System

Figure 1: Mobile robot

In Figure 1, X_IOY_I is the global reference coordinate system. Select the midpoint $P(x, y)$ of the two driving wheels and consider this point as the position of the mobile robot in the current coordinate system. To represent the relationship between the mobile robot and the global coordinate system, a local reference coordinate system X_RPY_R for the mobile robot is established with

$P(x,y)$ as the origin, and θ is used to represent the angle between the global reference coordinate system X_IOY_I and the local reference coordinate system X_RPY_R [13-14]. The global pose of the mobile robot is represented by U_I, and the local pose is represented by U_L. The expression and mapping relationship between the two are shown in the following expression.

$$U_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{1}$$

$$R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

$$U_L = R(\theta)U_I \tag{3}$$

Two-wheel differential mobile robots have three degrees of freedom, and the input is provided by a pair of drive wheels installed parallel to the left and right sides of the chassis. The motion of this type of robot belongs to coupled motion. From a kinematic perspective, decoupling analysis shows that since these driving wheels are rigidly connected and can be decomposed into circular motion around a certain center, the angular velocity of these driving wheels is the same [15]. Figure 2 is a schematic diagram of the motion of a two-wheel differential mobile robot.

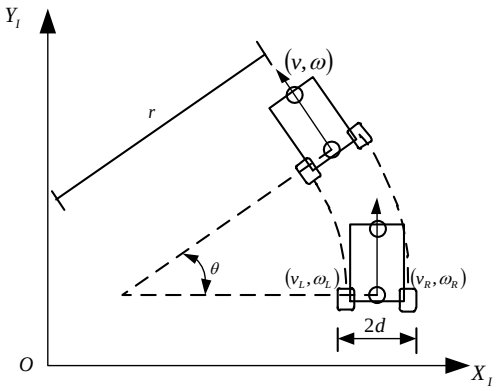


Figure 2: Motion diagram of mobile robot

In Figure 2, (v,ω) represents the linear velocity and angular velocity of the mobile robot, (v_L,ω_L) represents the linear velocity and angular velocity of the left wheel, (v_R,ω_R) represents the linear velocity and angular velocity of the right wheel, r is the rotation radius of the mobile robot during circular motion, and d is the distance between the power wheel on one side of the mobile robot and the center of the chassis. Since the angular velocities of the two power wheels of the mobile robot are the same, it can be concluded that:

$$\begin{cases} \omega = \omega_L = \omega_R \\ \dfrac{v_L}{r-d} = \dfrac{v_R}{r-d} \\ v_L(r+d) = v_R(r-d) \end{cases} \tag{4}$$

$$\begin{aligned} r &= \frac{(v_L+v_R)d}{v_R-v_L} \\ r+d &= \frac{(v_L+v_R)d}{v_R-v_L} + \frac{(v_R-v_L)d}{v_R-v_L} = 2\frac{v_R d}{v_R-v} \end{aligned} \tag{5}$$

Substituting equation (5) into the angular velocity equation yields $\omega = \dfrac{v_R}{r+d} = \dfrac{v_R-v_L}{2d}$. From this, the linear velocity of the mobile robot can be calculated, as shown in equation (6):

$$v = \omega \cdot r = \frac{v_R-v_L}{2d}\frac{(v_R+v_L)d}{v_R-v_L} = \frac{v_R+v_L}{2} \tag{6}$$

From the above derivation process, it can be seen that the overall linear velocity and angular velocity of a two-wheel differential mobile robot can be expressed by the left wheel linear velocity v_L, the right wheel linear velocities v_R and d. If ω is not 0, the expression for the mathematical model of the mobile robot is as follows:

$$\begin{cases} x^* = x - r\sin\theta_t + r\sin(\theta_t(1+\omega)) \\ y^* = y - r\cos\theta_t + r\cos(\theta_t(1+\omega)) \\ \theta_t^* = \theta_t + \omega\Delta_t \end{cases} \tag{7}$$

In the equation, x and y respectively represent the movement distance of the mobile robot in the X-axis and Y-axis directions during one simulation cycle, and θ represents the angle between the motion direction of the mobile robot and the X-axis during one simulation cycle [16-17].

In the above process, the wheel speed is the core input parameter of the model, which is directly derived from the speed measurement of the drive motor. d is the inherent structural parameter of the model, which is directly determined by the actual physical design size of the mobile robot chassis and belongs to a known constant. The core derivation of the model (i.e., formula (7)) shows how the increment of the robot's position and posture is determined by the current speed command and the duration of the simulation cycle when the robot is in non-linear motion. The calculation of these kinematic parameters completely depends on the above input parameters and structural parameters.

3 Mobile robot speed acquisition

Based on the mathematical model of the mobile robot constructed above, its kinematic relationship is clarified. In order to achieve effective collision control in the actual movement, the key is to obtain and optimize the motion state of the robot in real time, that is, its linear velocity and angular velocity. These velocity vectors define the instantaneous motion direction and speed of the robot at

each moment, and directly affect its future trajectory. However, not all theoretically feasible speeds are actually available. They are constrained by the physical limitations of the robot itself and the requirements of environmental safety. Therefore, this study uses dynamic window approach (DWA) for velocity acquisition. The core of this method is to comprehensively consider the kinematic constraints of the mobile robot, the limits of motor dynamics and the safety requirements for obstacle avoidance, dynamically screen out the feasible speed range (i.e. "dynamic window") in the current motion cycle, and sample and evaluate the potential speed instructions from it, so as to provide accurate speed input basis for subsequent Obstacle Avoidance Trajectory Planning and optimization.

There are multiple velocity vectors (v,ω) in the velocity space, which describe the motion state of the robot at different times and positions. Therefore, it is necessary to collect the velocity vector (v,ω) that conforms to the kinematics of the mobile robot. The DWA will obtain the corresponding moving speed range including angular velocity and linear velocity within the current motion cycle based on the limitations of the mobile robot's own conditions, such as the performance of the motor, the friction between the tire and the ground, the weight of the mobile robot, and the motion mode, in order to sample the motion trajectory of the mobile robot.

In DWA, the linear velocity sampling interval is 0.1m/s, the angular velocity sampling interval is 0.05rad/s, and the dynamic window size is 7s. The sampling speed space can be composed of three constraints: mobile robot kinematics, motor dynamics, and safety. Specifically, as shown below:

(A) Kinematic constraints

The size of the motion range is mainly related to factors such as the structural characteristics of the mobile robot itself, and the range of motion of the mobile robot is constrained by the extreme values of the linear velocity and angular velocity of different types of robots [18]. According to the performance of the mobile robot itself, its maximum and minimum speed limit range set V_S is as follows:

$$V_S=\left\{(v,\omega)\middle|v\in[v_{\min},v_{\max}]\cap\omega\in[\omega_{\min},\omega_{\max}]\right\} \quad (8)$$

In the equation, $\omega_{\min}$ and $\omega_{\max}$ represent the minimum and maximum angular velocities of the robot, respectively, while $v_{\min}$ and $v_{\max}$ represent the minimum and maximum linear velocities of the robot, respectively.

(B) Motor dynamics constraints

The motion speed of a mobile robot is determined by the speed of the motor and the diameter of the tire. From the perspective of driving, motors are usually affected by parameters such as operating voltage, speed, torque, etc. Therefore, the extreme values of angular velocity and linear velocity of mobile robots will have certain limitations. The acceleration set V_d of the mobile robot is as follows:

$$V_d=\left\{(v,\omega)\middle|v\in[v_c+\dot{v}_b\Delta t,v_c+\dot{v}_a\Delta t]\cap\omega\in[\omega_c+\dot{\omega}_b\Delta t,\omega_c+\dot{\omega}_a\Delta t]\right\} \quad (9)$$

In the equation, v_c and ω_c respectively represent the linear velocity and angular velocity of the mobile robot in its current state, $\dot{v}_a$ and $\dot{v}_b$ respectively represent the maximum acceleration and deceleration of the mobile robot's linear velocity, $\dot{\omega}_a$ and $\dot{\omega}_b$ respectively represent the maximum acceleration and maximum deceleration of the angular velocity of the mobile robot [19-20].

(C) Security constraints

Braking distance is an important parameter for collision free motion of mobile robots. The shorter the braking distance, the better the performance of the mobile robot, and it is often used as an indicator of braking energy efficiency [21]. In order to ensure the safety performance of the DWA, based on the braking distance and under the constraints of motor dynamics and kinematics, a set of safe speeds V_a is created as follows:

$$V_a=\left\{(v,\omega)\middle|v\le\sqrt{2\cdot dist(v,\omega)\cdot\dot{v}_b}\cap\sqrt{2\cdot dist(v,\omega)\cdot\dot{\omega}_b}\right\} \quad (10)$$

In the equation, $dist(v,\omega)$ represents the Euclidean distance between the mobile robot and the obstacle in the created sampling speed space. Combining the constraints of the above three conditions, taking their intersection as the velocity set of the sampling velocity space and representing it as V, the velocity set that meets the constraints is $V=V_S\cap V_d\cap V_a$.

For the above three constraints: among the kinematic constraint parameters, the range of linear velocity and angular velocity is determined by the physical structure of the robot and task requirements. The wider the parameter value, the more candidate velocity combinations, but the greater the computational burden; In the motor dynamics parameters, the limits of acceleration and deceleration are derived from the motor torque characteristics and load model. Too large acceleration setting will lead to the distortion of trajectory prediction, and too small will limit the dynamic obstacle avoidance ability; In the safety constraint parameters, $dist(v,\omega)$ associated with the braking distance obtains the environmental distance information through the real-time sensor, and its update frequency directly affects the timeliness of the window.

4 Design of collision control during the travel process of three mobile robots

4.1 Construction of moving objective function for mobile robots

Based on the speed data of the mobile robot obtained above, a target optimization function is constructed using

the minimum pseudo distance to complete the obstacle avoidance task and achieve smooth and collision free safe path planning for the robot.

The obstacle avoidance distance proximity index is usually measured by the Euclidean distance between robots and obstacles. It is very difficult to accurately describe complex environmental objects in analytical form. To avoid collisions between mobile robots and obstacles, it is only necessary to obtain the proximity of the distance between the two [22]. Based on this concept, the minimum pseudo distance metric is used to measure the Euclidean distance index, which is used to evaluate the proximity of mobile robots to obstacles.

Set the obstacle reference coordinate system $\{o_{obs}-x_{obs}y_{obs}z_{obs}\}$ at the geometric center of the obstacle, and use a unified hyperquadric surface analytical equation to describe the obstacle surface in the environment, that is:

$$\left(\frac{x'-x'_0}{h_1}\right)^{2m}+\left(\frac{y'-y'_0}{h_2}\right)^{2n}+\left(\frac{z'-z'_0}{h_3}\right)^{2p}=1 \quad (11)$$

In the equation, (x',y',z') represents the coordinate position of any point on the hyperquadric surface that fits the shape of the obstacle, (x'_0,y'_0,z'_0) represents the position coordinate of the center point of the hyperquadric surface in the base coordinate system $\{o_{obs}-x_{obs}y_{obs}z_{obs}\}$, and $h_1,h_2,h_3>0$ and $m,n,p\geq 1$ respectively describe the volume parameters and shape parameters of the hyperquadric surface.

Using regular geometry to fit the shape of irregular obstacles in the environment, regularize environmental obstacles, and facilitate the establishment of obstacle avoidance models and trajectory optimization analysis. Due to its ability to envelop obstacles of various shapes on the surface of a sphere, it is an ideal regular geometry. Taking a sphere in three-dimensional space as an example, establish a pseudo distance expression from a point to a hyperquadric surface:

$$D(x',y',z')=\left(\frac{x'-x'_0}{R_{shp}}\right)^2+\left(\frac{y'-y'_0}{R_{shp}}\right)^2+\left(\frac{z'-z'_0}{R_{shp}}\right)^2-1 \quad (12)$$

In the equation, $D(x',y',z')$ represents the pseudo distance function expression, R_{shp} represents the safe sphere radius of environmental obstacles, $R_{shp}=r_{obs}+r_{link}$ and r_{obs} represent the actual sphere obstacle model radius, and r_{link} represents the maximum radius of the mobile robot.

The three positional relationships between any point in space and a hyperquadric surface (in this article, a spherical surface is used) are shown as follows: if a point (x',y',z') in space is inside the hyperquadric surface, it indicates a pseudo distance $D(x',y',z')<0$ between the two, that is, the mobile robot has made contact with the obstacle; If point (x',y',z') is on the surface of a hyperquadric, it indicates a pseudo distance value $D(x',y',z')=0$ between the two, which means that the mobile robot just made contact with the obstacle; If point (x',y',z') is outside the hyperquadric surface, it indicates a pseudo distance value $D(x',y',z')>0$ between the two, that is, the mobile robot has not made contact with obstacles, and the robot is in a relatively safe state.

The minimum pseudo distance from any point P_i on the mobile robot to the obstacle surface in the environment is ${}^{obs}d_{\min}=\min D(x_{P_i},y_{P_i},z_{P_i})$, and it needs to be ensured that it is always greater than 0. In environments with obstacles, these three redundancies give the robot more ways to adjust its posture and path, thereby better avoiding obstacles. Define the minimum pseudo distance as the optimization objective metric function:

$$H(q)={}^{obs}d_{\min}(q) \quad (13)$$

After obtaining the optimization target function, the robot trajectory control for obstacle avoidance is carried out according to the optimization result of the target function. The equation is:

$$\dot{q}=\dot{q}_{endn}+\dot{q}_{null}=J^{+}(q)(x_d(V)+K(V)sat(\mu e(t)))+\xi N_0\nabla H(q) \quad (14)$$

In the equation, $\dot{q}_{endn}$ represents a parameter used to improve trajectory tracking accuracy, which can be adjusted by Jacobi pseudo inverse matrix closed-loop control algorithm. $\dot{q}_{null}$ is the angular velocity item of obstacle avoidance task, which does not change the trajectory route of the actuator, and is only used to provide the angular velocity of robot space obstacle avoidance. Where, $J^{+}(q)$ represents the pseudo inverse matrix, $x_d(V)$ represents the speed vector mapped to the robot speed set obtained above, $K(V)$ represents the adaptive positive definite gain matrix, $sat(\mu e(t))$ represents the saturation function, μ represents the constant coefficient, $e(t)$ represents the position error of the robot, N_0 is the mapping matrix of Jacobi, $\nabla H(q)$ represents the gradient projection matrix of the solution result of the optimization objective index function, which is used to optimize the obstacle avoidance and escape speed closest to the obstacle, ξ is the scaling coefficient of the obstacle avoidance angular speed.

The compound control law shown in equation (14) is solved online by numerical method, and the operator splitting quadratic programming (osqp) solver is used to deal with the quadratic programming subproblem of obstacle avoidance, with the calculation cost controlled within 1.5ms. The Jacobian matrix is based on the real-time analytical calculation of the kinematic model of the mobile robot, which is composed of the mapping from wheel speed to linear speed and angular speed of the body, and the dimension is 2×2.

To obtain the optimal ξ, introduce the sigmoid function:

$$\xi\left({}^{obs}d\right)=\begin{cases} v_{\max}\left(\dfrac{1}{1+\exp\left(-a\left({}^{obs}d-d_0\right)\right)}\right) & {}^{obs}d \le d_0 \\ 0 & {}^{obs}d > d_0 \end{cases} \quad (15)$$

In the equation, d_0 represents the pseudo distance alert threshold, which is determined based on the complexity of the environment and the actual operating characteristics of the robot. $v_{\max}$ represents the maximum obstacle avoidance and escape velocity of the robot. a represents the curvature adjustment factor of the sigmoid function. When taking $v_{\max}=0.4m/s$ and $d_0=3.564$, the images of $\xi\left({}^{obs}d\right)$ and ${}^{obs}d$ vary with the value of a as shown in Figure 3.

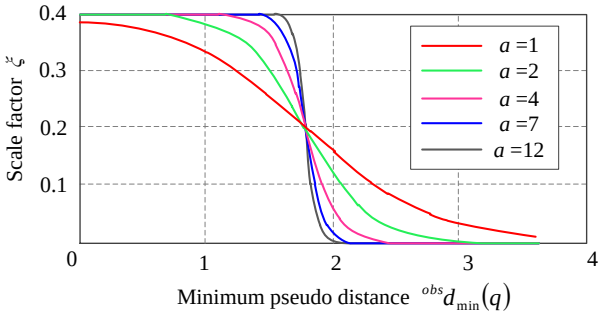


Figure 3: Relationship between scaling factor and minimum pseudodistance

Figure 3 shows a schematic diagram of the robot's closest point and obstacle sphere. When ${}^{obs}d > d_0$, the robot's closest point is outside the alert boundary, $\xi\left({}^{obs}d\right)=0$. There is no obstacle avoidance movement at this time. When ${}^{obs}d \le d_0$, the closest point of the robot is within the alert area. Calculate the size of $\xi\left({}^{obs}d\right)$ using equation (15), and control the robot to immediately start obstacle avoidance and escape speed according to equation (14) to complete the robot's obstacle avoidance task.

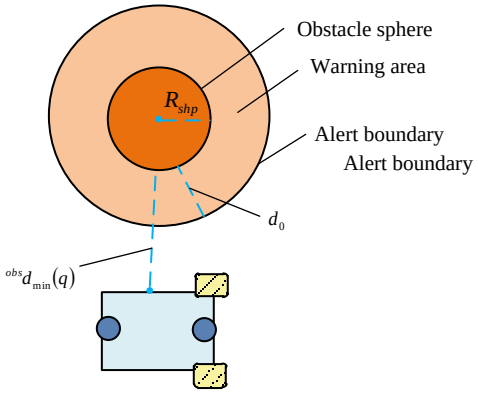


Figure 4: Schematic diagram of the robot's closest point and obstacle sphere

In practical application, for irregular obstacles in the real world, the minimum pseudo distance measurement method is realized by environment perception and dynamic fitting. Firstly, the multi camera vision system is used to collect the environmental point cloud data in real time, and the independent obstacle clusters are identified by segmenting the point cloud. For each obstacle cluster, the minimum bounding sphere approximation is used to expand the processing: The point cloud cluster is completely wrapped by the sphere with the minimum radius, and the spherical center coordinates and safety radius are output in real time. For non convex obstacles, they are decomposed into several sub clusters, and the sub spheres are fitted respectively. The final pseudo distance is the minimum of all sub sphere pseudo distances. In order to be compatible with sensor noise and motion uncertainty, the safety margin of 10% -20% robot radius is superimposed on the fitting radius. Although this method has a conservative path for the concave area, through 10Hz dynamic re fitting and safety margin compensation, the control error is stable within 0.1, which balances the efficiency and reliability.

According to the above steps, effective obstacle avoidance for mobile robots has been achieved. However, the objective function defined by equation (13) has a high degree of nonlinearity. In the obstacle avoidance problem of mobile robots, factors such as the pseudo distance relationship between the robot and obstacles, the robot's motion posture, and the complex geometric shape of the environment may all lead to the objective function exhibiting complex nonlinear characteristics. For example, when a robot moves in a complex environment, the pseudo distance calculation between it and multiple obstacles, as well as the coupling relationship between them, can make the objective function extremely complex and difficult to solve directly through traditional optimization methods. Traditional optimization algorithms may get stuck in local optima when dealing with nonlinear problems, making it difficult to find global optima. Therefore, an improved marine predator algorithm is adopted to optimize and solve the objective function. Provide basic information data for robot trajectory control.

4.2 Optimization solution of mobile robot moving targets based on improved marine predator algorithm

The objective function of obstacle avoidance for mobile robots often has a high degree of nonlinearity, forming a high-dimensional search space. Traditional optimization algorithms are prone to getting stuck in local optima in such complex situations. MPA simulates the predatory behavior of marine organisms and has strong global search capabilities. During the process of searching for prey, predators will extensively explore the entire search space, not limited to local areas [23].

MPA mainly consists of two operators, namely predation and elimination of individual aggregation. Among them, the predation operator is divided into the following three stages:

Stage 1: Parameter initialization and random initialization of prey population.

In the path search problem of mobile robots, a group of prey with number N is randomly initialized in a D-dimensional space as shown in equation (1). The dimensional space here can be understood as the various possible path state spaces of mobile robots in complex environments. After calculating the fitness value of each individual, determine the historical optimal position matrix P of the individual and select the optimal individual, namely the predator, whose position is E, and save E and P.

$$X_i = L_B + R \otimes \left(U_B - L_B\right) \tag{16}$$

Among them, X_i represents the random initialization position of the i-th prey, $i \in (1, N)$. L_B and U_B represent the upper and lower bound vectors of the search space, R represents the D-dimensional random vector, and each component value is a random number uniformly distributed in [0,1]. $\otimes$ represents the multiplication operation of each component in the vector. For mobile robots, this initialization process involves randomly exploring some initial paths in an unknown environment, providing a foundation for subsequent optimization searches.

Stage 2: Predator operator.

Dividing the entire search process into three stages during predation, each accounting for one-third of the entire search stage, is of great significance for mobile robots to find the optimal path.

In the initial exploration stage. At this stage, it is assumed that the speed ratio between prey and predator is high, with the predator being faster than the prey in terms of speed. This stage occurs in the first third of the total number of iterations, and updates the position of the prey as follows:

$$X_i = P_i + p \times R \otimes R_B \otimes \left(E - R_B \otimes P_i\right) \tag{17}$$

Among them, P_i represents the historical optimal position of the R_B-th prey, and R_B represents the D-dimensional random vector of the Brownian walk, with each component taking a random number from a normal distribution. p represents the parameter. In the obstacle avoidance control scenario of mobile robots, this stage is when the robot quickly explores various possible directions in the environment, finds some potential feasible paths, and the randomness of Brownian walk enables the robot to cover the search space more widely.

Mid term exploration and mining transition stage. This stage occurs between one-third and two-thirds of the entire iteration, with predators and prey moving at almost the same speed. At this stage, prey uses $Levy$-walk, while predators use Brownian walk. The former is beneficial for mining, while the latter is beneficial for exploration. At this stage, half of the population $i = 1,2,\cdots,\frac{N}{2}$ uses X_U from equation (18) for position updates, while the other half $i = \frac{N}{3}, \frac{N}{4}, \cdots,$ uses X_L from equation (18) for position updates.

$$\begin{aligned} X_U &= P_i + p \times R \otimes R_L \otimes \left(E - R_L \otimes P_i\right) \\ X_L &= E + p \times c_f \times R_B \otimes \left(R_B \otimes E - P_i\right) \end{aligned} \tag{18}$$

Among them, R_L represents the D-dimensional random vector generated by the Levy distribution, and c_f represents the dynamically adjusted parameter, whose value changes dynamically with the number of iterations, such as $c_f = \left(1 - t/T\right)^{2t/T}$. Among them, T represents the maximum number of iterations, and t represents the current number of iterations. For obstacle avoidance control of mobile robots, this stage involves a more detailed search in previously explored areas, while also continuing to explore new areas to find better paths.

The later stage of mining. In the last third of the iterations, the speed ratio between prey and predator is very low. In terms of speed, predator is slower than prey. Predator uses $Levy$-walk, and its mathematical model is:

$$X_i = E + p \times c_f \times R_L \otimes \left(R_L \otimes E - P_i\right) \tag{19}$$

At this stage, mobile robots focus more on the potential path areas that have already been discovered, conducting deeper searches and optimizations to find the optimal obstacle avoidance path.

In the above process, considering that the external interference force in the process of robot walking is random and time-varying, it is impossible to achieve accurate modeling and calculation. Therefore, this study does not consider the direct processing of external interference force, but in the face of nonlinear time-varying characteristics, according to the position and motion state information of other individuals, adjust their own motion trajectory and speed in real time.

Stage 3: Eliminate individual aggregation operators.

In the ocean, fish often exhibit aggregation phenomena under eddies or fish aggregation devices (FAD), which correspond to algorithms being trapped in local optima. For obstacle avoidance control of mobile robots, it is like the robot getting stuck in some local optimal areas when searching for a path, making it difficult to find the global optimal path, resulting in obstacle collisions and path growth. In order to consider the impact of such environmental factors and avoid opening points, MPA implements strategies to prevent the occurrence of such fish aggregation phenomena, which are expressed as follows:

$$\tilde{X} = \begin{cases} P + c_f\left(L_B + R \otimes \left(U_B - L_B\right)\right) \otimes U & r \le f_{FADs} \\ P + \left(f_{FADs} \times (1 - r) + r\right) \times \left(P_1 - P_2\right) & r > f_{FADs} \end{cases} \tag{20}$$

Among them, $\tilde{X}$ represents the updated position matrix of each individual in the current population, and

f_{FADs} represents the parameters that affect the optimization process. The value in this article is 0.1. U represents a matrix containing 0 and 1, r represents a random number uniformly distributed in [0,1], R represents a [0,1] random number matrix, P_1、P_2 represents a matrix composed of two completely different sets of historical optimal positions, and L_B、U_B represents the upper and lower bound matrices of the search space, each consisting of N repetitions of L_B、U_B. All matrices with the same size represent $N \times D$.

After updating the position of the prey, boundary control is applied to the new position of each prey and the fitness value f_{fit} is calculated. Finally, the greedy selection is used to update the individual historical optimal position matrix P and the elite individual position E. The greedy selection is shown below.

$$P_i(t+1)=\begin{cases} X_i(t+1) & \text{if} f_{fit}(X_i(t+1)) \geq f_{fit}(P_i(t)) \\ P_i(t) & otherwise \end{cases} \quad (21)$$

In the path search of mobile robots, this step ensures that the robot's path exploration is always within a reasonable range and can be selected and updated based on the quality of the path, gradually finding a better path. According to the above steps, it can be seen that the MPA algorithm has strong global search ability. In the early stages of evolution, the values of the components of the Brownian walk vector are relatively large, which helps to expand the search range, eliminate individual aggregation operators, and avoid the algorithm falling into local optima.

However, although MPA achieves certain information utilization in the predator-prey operator, this information utilization only exists between the current individual and the optimal individual. Only the optimal individual is used to guide other individuals, and there is no communication between other individuals. This results in a low degree of information sharing, leading to insufficient search ability of MPA in solving a class of complex optimization problems. In the practical application of mobile robots, the problem of insufficient information sharing may make it difficult for robots to fully utilize the explored path information, thereby affecting the efficiency of finding the optimal obstacle avoidance path. Therefore, in order to enhance the degree of information sharing among individuals and improve global search capabilities, in addition to random individuals and the worst individuals, an information sharing strategy is adopted in the first third of the entire search process to update the positions of other individuals. The mathematical model is shown below.

$$X_i = P_i + f_r \times R \otimes \begin{cases} (P_m - P_n) & i = 1,2,\cdots,N/2 \\ (E - P_i + P_m - P_n) & i = N/2+1,\cdots,N \end{cases} \quad (22)$$
$$f_r = 0.5 \times (\sin(2\pi \times 0.25t) \times t/T + 1)$$

Among them, P_m、P_n represents the historical optimal positions of two randomly selected prey from the current population, and $m \neq n \neq i$ and f_r represent the scaling factors of the sine model.

From equation (22), it can be seen that half of the individuals in the population use two randomly selected historical optimal positions for differentiation. This random differential mutation strategy acts on many individuals in the population, enabling information sharing among individuals within the population. And the other half of the individuals not only use the historical best position difference between two random individuals, but also add the difference between the best individual and the current individual's historical best position, which not only increases the guidance of the global best individual, but also increases the diversity of understanding. Comparing equation (17), parameter p has been replaced by f_r in equation (22) to avoid parameter adjustment; In addition, using a sine model in f_r is beneficial for individuals to escape from local optima and achieve a better balance between exploration and mining.

This improvement aims to enhance the global search ability and information sharing among individuals in the initial exploration stage of the algorithm by introducing an update strategy similar to differential evolution, rather than completely replacing the original mechanism. This improvement effectively balances exploration and mining through the scaling factor adjusted by the sinusoidal model, and helps individuals jump out of the local optimum. This improvement is intended to highlight the improvement of algorithm performance and adaptability, while preserving the essence of MPA's core framework and predator prey interaction.

After improvement, to verify the necessity of information sharing mechanism. In the same experimental environment (30×30m with complex obstacles), the performance of the improved MPA and the original MPA (without information sharing) were compared

① Convergence speed: the average number of iterations of the improved algorithm is reduced from 48 to 32, because the improved algorithm enhances the information interaction between individuals and avoids invalid search due to random difference mutation.

② Local optimal escape rate: in the U-shaped trap scenario, the original MPA has a 67% probability of falling into local optimal due to individual isolation, while the improved MPA increases the escape rate to 92% by sharing the historical optimal location.

③ Path length optimization: compared with the average path length of 50 experiments, the improved MPA is 7.5% shorter than the original MPA, which proves that information sharing can synergistically mine better solutions.

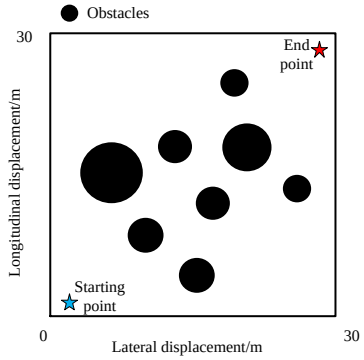
In summary, information sharing strategies are mainly used in the early stages to enhance population diversity and improve global search capabilities. For mobile robots, this improvement can enable them to more efficiently utilize existing path information, find the optimal obstacle avoidance path faster and more accurately in complex environments, and successfully complete tasks.

5 Experiments and results analysis

5.1 Experimental environment

To verify the anti-collision control effect of the method in this study on the robot, an indoor space was selected as the experimental site, and the ground of the site was flat to ensure the normal movement of the robot. The size of the venue is 30 meters long and 30 meters wide, providing sufficient space for robots to move. Place different obstacles in it to test the collision prevention control effect of the robot. The simulation platform is matlab. The experimental mobile robot and experimental environment are shown in Figure 5.

(a) Experimental mobile robot



(b) Experimental environment diagram

Figure 5: Experimental mobile robot and environment diagram

The size of the robot is 0.75×0.4×0.4m. It is driven by two wheels, and each wheel is driven by an independent DC motor. It has good obstacle surmounting and steering ability The maximum speed is 0.5m/s, and it can climb the slope with the maximum gradient of 15 °. The top of the robot is equipped with a 1080p high-definition camera, which has a wide-angle field of view (horizontal angle of view 120 °, vertical angle of view 90 °), can shoot 360 ° rotation, and supports night vision. Four ultrasonic sensors are installed at the front end and both sides of the robot to detect the distance of obstacles, with a measurement range of 0.2~4m and an accuracy of ±1cm.

The relevant experimental parameters are shown in Table 1.

Table 1: Parameter setting for mobile robot and improved MPA

set up	parameter	data
mobile robot	Wheel radius/m	0.18
	Mass of robot/kg	2.5
	Gravity acceleration/m/s^2	9.2
	Damping coefficient/Ns/m^2	0.02
	Expected angle/°	0.5
Improving the Marine Predator Algorithm	Population size	50
	Predator ratio	0.2
	Step size control factor	0.5
	Elite retention ratio	0.1
	Pseudo distance alert threshold	0.8m
	Maximum obstacle avoidance and escape speed of robots	1.5m/s
	Curvature adjustment factor of sigmoid function	2

The pseudocode related to method in this study is as follows:

Initialize:

Robot state: x, y, θ # Position and orientation

Goal position: $H(q)$

Obstacle list: (x', y', z') # From sensors

MPA parameters: N, D, E, P, X_i, etc.

while robot not at goal:

1. Velocity Sampling via Dynamic Window Approach (DWA)

kinematics_constraints,

#

$$V_S = \{(v,\omega) | v \in [v_{\min}, v_{\max}] \cap \omega \in [\omega_{\min}, \omega_{\max}]\}$$

dynamics_constraints,

#

$$V_d = \{(v,\omega) | v \in [v_c - \dot{v}_b \Delta t, v_c - \dot{v}_a \Delta t] \cap \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c - \dot{\omega}_a \Delta t]\}$$

safety_constraints,

$$\# \; V_a = \{(v,\omega) | v \le \sqrt{2 \cdot dist(v,\omega) \cdot \dot{v}_b} \cap \sqrt{2 \cdot dist(v,\omega) \cdot \dot{\omega}_b}\}$$

2. Objective Function Construction

$$H(q) = {}^{obs}d_{\min}(q)$$

3. Optimize $H(q)$ via Improved MPA

info_sharing = True,

$$\# \; X_i = P_i + f_r \times R \otimes \begin{cases} (P_m - P_n) & i = 1,2,\cdots,N/2 \\ (E - P_i + P_m - P_n) & i = N/2+1,\cdots,N \end{cases}$$

$$f_r = 0.5 \times (\sin(2\pi \times 0.25t) \times t/T + 1)$$

4. Null-Space Obstacle Avoidance Control

The multi link control method for snake like robot proposed in reference [6], the robot control method based on CBF proposed in reference [7] and the robot control method based on sinusoidal oscillator proposed in reference [8] were selected as the comparison methods, and the subsequent analysis and test were carried out based on the above experimental parameter settings.

5.2 Experimental indicators

Control error: this index measures the deviation between the desired trajectory or target position of the robot in the actual process of travel. The choice of reference is based on the core requirement of mobile robot navigation - accurate pose control. The control error directly reflects the ability of different methods to guide the robot to avoid obstacles and reach the target position accurately. The small error shows that the control method can effectively overcome the environmental interference and system nonlinearity, and maintain high-precision trajectory tracking performance, which is very important to ensure the accuracy and safety of task execution.

The expression is:

$$e = \sqrt{\sum_{i=1}^{n} (x - x_d)^2 + (y - y_d)^2} \tag{23}$$

Among them, (x, y) represents the actual position of the robot, and (x_d, y_d) is the ideal position. If the control error is small, it indicates that the algorithm can accurately plan the path of the robot, maintaining high navigation accuracy in complex environments and effectively avoiding collisions.

Obstacle avoidance effect: this index intuitively evaluates the ability of the robot to realize collision free navigation in obstacle environments with different complexity. The basic goal of anti-collision control method is to ensure the safety of robot. By observing the actual trajectory of the robot in the preset environment, we can determine whether there is a collision or whether the planned path is significantly deviated from the optimal. The successful obstacle avoidance effect needs to meet the two basic requirements of no collision and relatively optimal path.

Response time: this indicator refers to the time required for different methods to plan or adjust the obstacle avoidance path for the robot. The selection criteria are based on the real-time requirements of mobile robot applications. In dynamic or unstructured environment, obstacles may suddenly appear or move, which requires the control system to be able to quickly respond to environmental changes and generate new obstacle avoidance instructions. Too long response time will lead to the robot unable to avoid obstacles in time, causing collision risk. This index directly reflects the calculation burden and real-time processing ability of different methods, and is the key factor to evaluate whether different methods can be deployed in practice.

Smoothness of path: this index is an index to quantify the tortuosity of the robot's motion trajectory, and is defined as the cumulative steering angle change within the unit path length. The lower the value, the straighter the path, the smoother the steering change, and the more efficient and chatterless obstacle avoidance trajectory can be achieved.

5.3 Experimental result

To fully verify the obstacle avoidance performance of the method in this study, three experimental environments were set up, namely, the environment with only two simple obstacles 1, the environment with multiple simple obstacles 2, and the environment with multiple complex obstacles 3. Four methods were used to test the robot control. The results are shown in Figure 6.

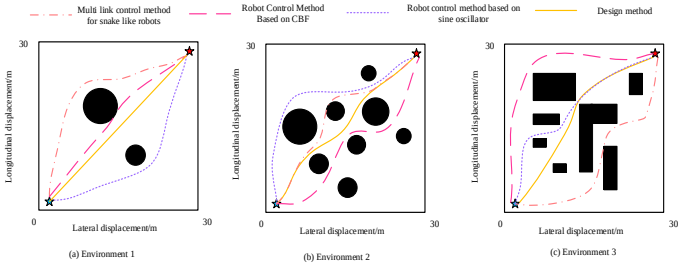


Figure 6: Analysis of obstacle avoidance effect

It can be seen from Figure 6 that in environment 1, environment 2 and environment 3, the method in this study can realize collision free obstacle avoidance navigation. Whether it is a simple obstacle environment or a complex obstacle environment, it can accurately plan the shortest path to ensure that the robot can reach the target position safely and efficiently. This is thanks to the mobile objective function based on minimum pseudo distance constructed in this article, which can accurately measure the proximity of robots to obstacles and provide a reliable basis for obstacle avoidance trajectory control. In contrast, in environment 1, robot control method based on CBF has obvious collision phenomenon. In environment 2, the robot control method based on CBF collides again. In environment 3, the multi link control method for snake like robot collides. The results show that the method in this study can realize the optimization of the path under the premise of ensuring safety.

Three methods were used to detect the control error of the robot, and the obtained detection results are shown in Figure 7.

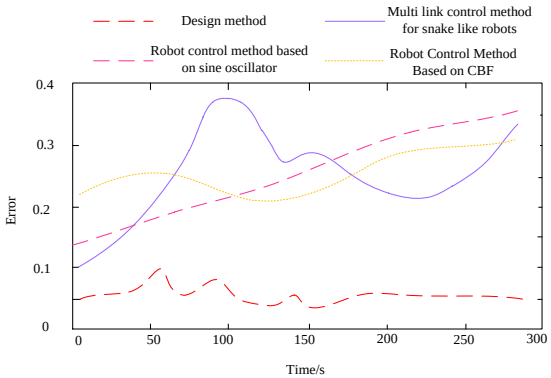


Figure 7: The detection results of control errors of robots by three methods

It can be seen from Figure 7 that compared with the other two methods, the robot control error of method in this study is within 0.1, while the control error of the comparative method is significantly higher than that of

method in this study, both higher than 0.1. This shows that the method in this study has excellent performance in obstacle avoidance, high accuracy and stability. This is because the method in this study uses the improved ocean predator algorithm to optimize the moving objective function, which has strong global search ability and convergence speed. It can quickly find the optimal trajectory and speed, so that the robot can track the target path more accurately.

To further validate the obstacle avoidance control performance of various methods, the response time of obstacle avoidance path algorithms designed by different methods was calculated. The specific experimental test results are shown in Table 2.

Table 2: Comparative analysis of response time of different methods

Iteration times/times	Multi link control method for snake like robot/s	Robot control method based on CBF/s	Robot control method based on sinusoidal oscillator/s	Method in this study/s
10	4.3	5.2	4.6	1.2
20	4.5	5.1	4.5	1.0
30	4.2	5.6	4.8	1.3
40	4.6	5.4	4.8	1.5
50	3.9	5.3	4.9	1.1
60	4.6	5.2	4.7	1.2
70	4.5	4.9	4.6	1.1
80	4.6	4.8	4.5	1.2
90	4.8	5.1	4.2	1.3
100	4.3	5.2	4.3	1.5
Standard deviation	0.25	0.28	0.22	0.14

It can be seen from the data in Table 2 that during the whole experiment, multi link control method for snake like robot and robot control method based on CBF The response time of method in this study is significantly lower than that of the other three methods. When the number of iterations is 10, multi link control method for snake like robot and robot control method based on CBF And robot control method based on sinuoidal ocillator response time were 4.3s, 5.2s and 4.6s respectively, while method in this study was only 1.2s.

The control error results shown in Figure 7 are based on periodic statistics, and the response time results shown in Table 2 are based on multiple independent experiments, which effectively reduces the impact of random fluctuations in a single experiment. The above results show that the method in this study can maintain efficient computational performance under different iteration times, and can quickly plan the obstacle avoidance path for the robot. This is very important for the real-time obstacle avoidance control of mobile robot in complex environment, which can ensure that the robot can quickly respond to various obstacles and effectively avoid collision.

Statistics of smoothness of path after different methods are applied. The test results are shown in Table 3.

Table 3: Comparative analysis of different methods of smoothness of path

Iteration times/times	Multi link control method for snake like robot	Robot control method based on CBF	Robot control method based on sinusoidal oscillator	Method in this study
10	2.15 ± 0.31	1.98 ± 0.29	2.37 ± 0.35	0.87 ± 0.12
20	2.08 ± 0.28	2.05 ± 0.33	2.41 ± 0.42	0.82 ± 0.10
30	2.21 ± 0.35	2.11 ± 0.30	2.29 ± 0.38	0.91 ± 0.14
40	2.17 ± 0.40	2.14 ± 0.36	2.45 ± 0.41	0.95 ± 0.15
50	2.03 ± 0.27	2.20 ± 0.41	2.33 ± 0.39	0.88 ± 0.11
60	2.12 ± 0.38	2.08 ± 0.35	2.50 ± 0.45	0.93 ± 0.13
70	2.19 ± 0.42	2.16 ± 0.39	2.42 ± 0.43	0.89 ± 0.12
80	2.25 ± 0.45	2.22 ± 0.42	2.38 ± 0.40	0.84 ± 0.10
90	2.30 ± 0.47	2.19 ± 0.44	2.47 ± 0.47	0.92 ± 0.14
100	2.14 ± 0.39	2.25 ± 0.46	2.41 ± 0.44	0.86 ± 0.11
Standard deviation	0.32	0.36	0.41	0.12

The data in Table 3 shows that the smoothness of path of method in this study is significantly better than the three comparison methods, with the minimum value of only 0.82 ± 0.10, and the standard deviation is only 1/3 to 1/4 (0.27 – 0.47) of the comparison methods, indicating that the generated path is not only straighter, but also significantly more stable. This advantage stems from the synergy of three mechanisms. Firstly, the improved MPa enhances the population collaboration through the information sharing mechanism to avoid the zigzag path caused by local search; Secondly, the obstacle avoidance command is dynamically injected into the motion control, and the minimum pseudo distance is used to evaluate the obstacle proximity in real time, so as to avoid the conservative emergency stop and steering; Finally, the optimal path with continuous curvature can be generated by dynamically fitting obstacles with superquadric

surfaces. In addition, the standard deviation of method in this study was stable between 0.10–0.15 rad/m, indicating that its variability was low. This is to improve the group collaborative optimization of MPa to generate a global smooth path, it can effectively avoid sharp turns.

6 Discussion

To more clearly illustrate the advantages and effectiveness of method in this study compared with existing methods, especially the ability to deal with complex dynamic obstacle scenes, the three comparative methods and method in this study are systematically compared and analyzed in Table 4.

Table 4: Comparative analysis of different methods

Contrast dimension	Multi link control method for snake like robot	Robot control method based on CBF	Robot control method based on sinusoidal oscillator	Method in this study
Core principles	Multiple snake gait planning based on joint coupling direction function	Design security constraints based on robust control barrier function, combined with distributed model predictive control	Redirection and obstacle avoidance based on periodic fluctuation and moment of inertia change of improved Snake curve	Fusion of minimum pseudo distance to construct objective function+optimization of improved ocean predator algorithm+Obstacle Avoidance Trajectory Control
Environmental adaptability	Bending model of soft actuator based on	Rely on accurate system model and analysis of security requirements	It is suitable for redundant systems with different number of modules, but the movement mode is inherently limited	It does not rely on the accurate environment/obstacle model, uses regular geometry to fit obstacles, and has strong generalization of minimum pseudo distance
Dynamic obstacle handling capability	The model is based on simplified assumptions, and it is difficult to accurately handle the actual complex mechanical behavior; Gait planning may lag behind dynamic changes	RCBF constraints are conservative, limiting the range of motion and speed, and it is difficult to respond flexibly and quickly to dynamic obstacles	Periodic fluctuation patterns may fail in highly dynamic or unstructured complex environments	The minimum pseudo distance is used to evaluate the proximity in real time; Improved MPa to efficiently search the optimal obstacle avoidance path globally; Fast response and real-time trajectory adjustment
Global optimization capability	Mainly solve gait planning, lack of global path optimization mechanism	DMPC is optimized in the prediction time domain, but it is prone to conservative constraints due to heavy computational burden	The motion mode itself limits the exploration of the global optimal path	The improved MPa has powerful global search ability, avoids falling into local optimization, and quickly finds the optimal obstacle avoidance path
real-time performance	Complex calculation of multi link coordinated control	DMPC online solving optimization problems requires a lot of	Oscillator based control is relatively efficient, but requires more	The improved MPa has fast convergence speed; Efficient calculation of minimum pseudo

		computation, and rCBF constraints increase the complexity	computation in complex environments	distance; Significantly lower overall response time
Characteristics of obstacle avoidance strategy	Detour based on specific gait	Passive evasion based on security constraints	Avoidance based on cyclical fluctuations	Active trajectory optimization+precise tracking and obstacle avoidance fusion based on swarm intelligence
Experimental performance (complex environment)	Collision in environment 3	Collision occurred in environment 1 and environment 2	Collision in environment 3	Collision free navigation is realized in environment 1, 2 and 3; Control error<0.1

The summary shows that in complex dynamic obstacle scenarios, the traditional methods have significant limitations: the multi link control method for snake like relies on a simplified soft actuator model, which is difficult to adapt to the actual complex mechanical behavior; Robot control method based on CBF limits robot mobility due to conservative constraints, resulting in dynamic obstacle avoidance failure; Robot control method based on sinuoidal is limited by fixed motion mode and is prone to collision in unstructured environment. These methods generally have the problems of strong model dependence, poor real-time performance and insufficient global optimization ability.

In contrast, the method in this study achieved a breakthrough through three innovations: first, the objective function was constructed by using the minimum pseudo distance, and the obstacle was enveloped by a superquadric sphere to overcome the problem of complex geometric modeling; Secondly, the sharing mechanism of individual position update information is introduced into the marine predator algorithm, and the ability of population collaborative search is enhanced through random differential mutation, which significantly improves the efficiency of global optimization; Finally, the real-time decoupling of trajectory tracking and obstacle avoidance is realized by combining obstacle avoidance control and dynamic adjustment strategy. The core advantage of improved MPa lies in: reducing invalid iterations through information sharing mechanism, so that the algorithm converges to the optimal solution quickly; The global optimality of the path is guaranteed by the brown hybrid levy walk strategy, and the control error is reduced from the source.

7 Conclusion

Aiming at the problem of mobile robot collision control in dynamic unstructured environment, this study proposes an efficient solution combining minimum pseudo distance and improved MPa. The innovative work of this research is reflected in: constructing the minimum pseudo distance objective function by dynamically fitting obstacles with superquadric surface, overcoming the limitations of complex geometric modeling, and accurately quantifying the proximity between the robot and obstacles; The information sharing mechanism of individual location update is embedded in the conventional MPa, and the ability of population collaborative search is enhanced by using random difference mutation; Combined with real-time speed sampling of DWA, obstacle avoidance evaluation of minimum pseudo distance and global optimization of improved algorithm, the dynamic decoupling of trajectory tracking and obstacle avoidance control is realized. Experiments show that the control error of this method is stable below 0.1 in complex obstacle environment, the response time is $\leq$ 0.5s, and the path smoothness is 0.82 $\pm$ 0.10 rad/m, which significantly improves the safety and agility of mobile robot in dynamic environment, and provides reliable technical support for industrial inspection, intelligent logistics and other scenes.

References

[1] Asma Saif, Raouf Fareh, Saif Sinan, et al. Fractional synergetic tracking control for robot manipulator [J]. Journal of Control and Decision,2024,11(1):139-152. DOI: 10.1080/23307706.2022.2146008

[2] Yiwei Hao, Yonghui Cao, Yingzhuo Cao, Xiong Mo, Qiaogao Huang, Lei Gong, Guang Pan, Yong Cao. Bioinspired Closed-loop CPG-based Control of a Robotic Manta for Autonomous Swimming[J]. DOI:10.1007/s42235-023-00424-z Journal of Bionic Engineering,2024,21(1):177-191.

[3] Van-Truong Nguyen, Dai-Nhan Duong, Dinh-Hieu Phan, et al. Adaptive Nonlinear PD Controller of Two-Wheeled Self-Balancing Robot with External Force[J]. Computers, Materials & Continua,2024,81(11):2337-2356. DOI:10.32604/cmc.2024.055412

[4] Qinchen Yang, Fukai Zhang, Cong Wang. Deterministic Learning-Based Neural PID Control for Nonlinear Robotic Systems[J]. IEEE/CAA Journal of Automatica Sinica,2024,11(5):1227-1238.DOI:10.1109/JAS.2024.124224

[5] Yuan Wang, Lining Xing, Junde Wang, et al. Multi-Objective Rule System Based Control Model with Tunable Parameters for Swarm Robotic Control in Confined Environment[J]. Complex System

Modeling and Simulation,2024,4(1):33-49. DOI: 10.23919/CSMS.2023.0022

[6] Xuanyi Zhou, Yuqiu Zhang, Zhiwei Qiu, et al. Locomotion control of a rigid-soft coupled snake robot in multiple environments[J]. Biomimetic Intelligence & Robotics,2024,4(2):12-23. DOI:10.1016/j.birob.2024.100148

[7] Mu Jianbin, Yang Haili, He Defeng. CBF-Based Distributed Model Predictive Control for Safe Formation of Autonomous Mobile Robots[J].of Shanghai Jiaotong university(Science),2024,29(4):678-688. DOI:10.1007/s12204-024-2747-7Journal

[8] Zhiyuan Yang, Mingzhu Lai, Jian Qi, et al. Reorientation and obstacle avoidance control of free-floating modular robots using sinusoidal oscillator[J]. Chinese Journal of Aeronautics,2024,37(6):262-275. DOI:10.1016/j.cja.2024.03.005

[9] Jin Yan, Wenguang Zhang, Yong Liu, Wei Pan, Xiaoyu Hou, Zhiyu Liu. Autonomous trajectory tracking control method for an agricultural robotic vehicle[J]. International Journal of Agricultural and Biological Engineering,2024,17(1):215-224. DOI:10.25165/j.ijabe.20241701.7296

[10] Manish Kumar, Karuna Panw, Kusum Deep. Discrete Marine Predators Algorithm for Symmetric Travelling Salesman Problem[J]. Evolutionary Intelligence,2024,17(5-6):3833-3848. DOI:10.1007/s12065-024-00960-5

[11] Hao Yin, Yanting Li, Zhiying Tian, et al. Ultra-High Sensitivity Anisotropic Piezoelectric Sensors for Structural Health Monitoring and Robotic Perception[J]. Nano-Micro Letters,2025,17(2):432-446.DOI:10.1007/s40820-024-01539-6

[12] Jin Tian, Xiulai Wang, Ningling Ma, et al. Adaptive Robust Control with Leakage-Type Control Law for Trajectory Tracking of Exoskeleton Robots[J]. Advances in Internet of Things,2024,14(3):53-66. DOI: 10.4236/ait.2024.143004

[13] Lingling Chen, Jjiabao Huang, Yanglong Wang, et al. Adaptive patient-cooperative compliant control of lower limb rehabilitation robot[J].Biomimetic Intelligence & Robotics,2024,4(2):42-49. DOI:10.1016/j.birob.2024.100155

[14] Qingjun Yang, Zhenyang Zhang, Rui Zhu, et al. Optimal Energy Efficiency Based High-speed Flying Control Method for Hydraulic Quadruped Robot[J]. Journal of Bionic Engineering, 2024,21(3):1156-1173. DOI:10.1007/s42235-024-00509-3

[15] Chen An, Jiaxi Zhou, Kai Wang. Adaptive state-constrained/model-free iterative sliding mode control for aerial robot trajectory tracking[J]. Applied Mathematics and Mechanics(English Edition),2024,45(4):603-618. DOI:10.1007/s10483-024-3103-8

[16] Yecheng Shao, Yongbin Jin, Zhilong Huang, Hongtao Wang, Wei Yang. A learning-based control pipeline for generic motor skills for quadruped robots[J]. Journal of Zhejiang University-Science A(Applied Physics & Engineering),2024,25(6):443-454. DOI:10.1631/jzus.A2300128

[17] Zhongping Chen, Weigong Zhang. Intelligent Sensing and Control of Road Construction Robot Scenes Based on Road Construction[J]. Structural Durability & Health Monitoring, 2024,18(2):111-124. DOI:10.32604/sdhm.2023.043563

[18] Wei Xingmin, Li Dekui. Control and Stabilization of Chaotic System Based on Linear Feedback Control Method[J]. Wuhan University Journal of Natural Sciences,2024,29(3):284-292. DOI:10.1051/wujns/2024293284

[19] Rei Ezaka, Takehito Yoshida, Yudai Yamada, Shin'ichi Warisawa, Rui Fukui. Reliability Improvement of a Crawler-Type Ceiling Mobile Robot in Starting, Accelerating, and Traveling Phase at High Speed: Regular Papers[J]. Journal of Robotics and Mechatronics,2023,35(6):1551-1561. DOI:10.20965/jrm.2023.p1551

[20] Kenji Kimura, Yuki Shigyo, Kazuo Ishii. An Analysis of Robot Speed Efficiency for Mobile Robot Adapted Three Omni Rollers Using Linear Transformation[J]. Journal of Advances in Artificial Life Robotics,2023,3(4):242-249. DOI:10.57417/jaalr.3.4_242

[21] Hamad Mazin, Gutierrez-Moreno Jesus, Kussaba Hugo T.M., Mansfeld Nico, Abdolshah Saeed, Swikir Abdalla, Burgard Wolfram, Haddadin Sam. Fast yet predictable braking manoeuvers for real-time robot control[J]. IFAC PapersOnLine,2023,56(2):9984-9991. DOI:10.1016/j.ifacol.2023.10.711

[22] Rong Yu, Dou Tianci, Zhang Xingchao. Closed-loop control dynamic obstacle avoidance algorithm based on a machine learning objective function[J]. Journal of Mechanical Science and Technology,2024,38(6):3089-3099. DOI:10.1007/s12206-024-0528-8

[23] Chuandong Qin, Baole Han. Multi-Stage Improvement of Marine Predators Algorithm and Its Application[J]. Computer Modeling in Engineering & Sciences,2023(9):3097-3119. DOI:10.32604/cmes.2023.026643

Debiasing Visual Question Answering via Ensemble Gradient Detection and Iterative Attention Forgetting

Qiuying Han[1], Shaohui Zhang[2,3,*], Peng Wang[2,3], Boyuan Li[3] and Qiwen Lu[4]
[1]School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, China
[2]School of Artificial Intelligence, Zhoukou Normal University, Zhoukou 466001, China
[3]School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China
[4]School of Computer and Information Engineering, Henan University, Zhengzhou 450046, China
E-mail: {hanqy, zhangsh, wangp}@zknu.edu.cn, ieboyuan@163.com, easonglll@henu.edu.cn
*Corresponding Author

Keywords: Visual question answering, language bias, multi-modal reasoning, attention mechanism, forgotten attention algorithm, ensemble models, gradient debiasing approach

Received: April 25, 2025

In Visual Question Answering (VQA), the model's ability to understand and reason across different modalities—language, visual, and multimodal—is crucial for accurate predictions. However, recent studies have identified a significant challenge of language bias in VQA models, where the model's reasoning often hinges on incorrect linguistic associations rather than genuine multimodal understanding. To address this issue, We propose a novel Ensemble Bias Gradient Debiasing Approach (EBGDA) that combines bias detection with a dynamic forgetting mechanism. Our method utilizes a bias detector to identify and score biases across linguistic, visual, and multimodal data, enabling the model to focus on unbiased information for more accurate predictions. Additionally, inspired by human reasoning, we introduce the Forgotten Attention Algorithm (FAA), which iteratively "forgets" irrelevant visual content, progressively concentrating attention on the image regions most relevant to the question. This combination of bias mitigation and attention focusing enhances the model's ability to make multimodal inferences, reducing bias and improving overall performance. Extensive experiments on the VQA-CP v2, VQA v2, VQA-VS, GQA-OOD, and VQA-CE datasets demonstrate the effectiveness of our approach, showing state-of-the-art performance in mitigating biases and excelling in complex multimodal scenarios. Our approach achieves a 21.32% improvement over UpDn on VQA-CP v2, establishing a new state-of-the-art among methods without data augmentation.

Povzetek: Članek predlaga metodo za VQA, ki z detekcijo pristranosti in dinamičnim pozabljanjem odstranjuje jezikovno pristranskost ter preusmeri pozornost na relevantne vizualne regije, da izboljša večmodalno sklepanje.

1 Introduction

In recent years, Visual Question Answering (VQA) has become one of the prominent tasks in the field of deep learning [1], achieving significant accomplishments in various applications, such as intelligent service systems [2, 3]. However, recent research has found that many existing VQA methods tend to rely on false associations between questions and answers, without sufficiently extracting accurate visual information from images to answer questions. For example, when answering questions "What color?", some VQA models are inclined to use the most common answers from training data of that type, like "yellow", rather than extracting genuine color information from images. Additionally, some studies [4, 5] have indicated deficiencies in the existing methods' understanding of images, resulting in answers generated by the model relying on image regions with low relevance to the questions. In other words, specific methods often provide correct answers based on incorrect image regions, which does not genuinely reflect the model's performance in the question-answering task. Consequently, the factors affecting the robustness of VQA models can be summarized into two primary aspects: inherent biases in the language distribution of training and testing datasets, and the improper shortcut biases caused by the inadequate utilization of visual information [6]. However, biases in VQA are multifaceted, affecting not only language but also the improper use of visual information, resulting in models that often focus on irrelevant regions of the image. As shown in Fig. 1, when the object elements in the image are nearly the same, the model during training will generate the answer "Standing" based on the object and keyword ("What" and "doing"). However, during testing, for the same question, the model will generate the incorrect answer based on similar objects and keywords (the correct answer is "Jumping", but the model gives the wrong answer

Q:What are the animals doing in the picture?

A: Standing

Q:What are the animals doing in the picture?

A: Jumping

Figure 1: An example of multimodal bias: the presence of visual features related to the dog and horse, along with the keyword "what" and "doing" in the question, misleads the model into predicting an incorrect answer. While the correct action for the second image is "Jumping", the model incorrectly predicts "Standing" based on similar objects and keywords.

"Standing").

The state-of-the-art and noteworthy methods primarily revolve around data augmentation techniques and attention-based approaches. Data augmentation methods [7] aim to enhance a model's understanding of critical features within the data by expanding the dataset with samples, such as counterfactual instances and additional annotations [8, 9], which help eliminate biases and enhance robustness [10, 11] by obtaining more critical sample features and supplementary information. However, it is still of great interest and challenge to remove the language biases in VQA model without resorting to data augmentation [12]. Regarding attention-based methods [13], the majority currently integrate these into pre-trained models for efficient feature fusion [13–17], with limited emphasis on fully utilizing visual information.

We introduce an Ensemble Bias Gradient Debiasing Approach (EBGDA), which incorporates ensemble learning to dynamically identify and score different types of biases in the data. The method applies overfitting techniques to certain biases to ensure that the main model focuses on unbiased and relevant information during prediction. Our bias detector, tailored to the multimodal nature of VQA, assesses the degree of bias in both linguistic and visual domains, helping the model to adapt its learning process accordingly.

Additionally, inspired by the idea of iterative human reasoning, where we progressively focus on relevant areas of an image based on the question, we propose a novel approach called the Forgotten Attention Algorithm (FAA). This algorithm progressively "forgets" irrelevant image regions after each iteration, allowing the model to concentrate on the most relevant visual information for answering the question. In Fig. 2, it is evident that prominent objects (*i.e.*, the bench) often dominate the model's attention, causing it to overlook the finer image area that is relevant to the question (*i.e.*, the people). This observation poses a new challenge: how to focus on the right image area that is the most relevant to the question. To address this problem, we are inspired by the process of answering questions of human beings, where people always gradually reduce the focus area in the image with the aid of question information until the final related area is retained. This iterative forgetting process helps the model mitigate the impact of biased associations by focusing on the crucial regions of the image that relate to the question, improving its reasoning capacity.

To validate the effectiveness of our approach, we conducted extensive experiments on the VQA-CP v2, VQA v2, VQA-VS, GQA and VQA-VE. The results show that our method, combining FAA and EBGDA, significantly outperforms existing models, without relying on additional annotations or data augmentation. Our approach effectively mitigates language and visual biases, enhancing the robustness of VQA models across various challenging scenarios. The main contributions of this study are summarized as follows:

(1) Ensemble Bias Gradient Debiasing Approach: We propose a unified multi-bias debiasing approach which leverages ensemble learning to dynamically identify and evaluate the impact of linguistic, visual, and multimodal biases. The method selectively overfits to harmful biases, enabling the main model to focus more effectively on unbiased and relevant information during prediction.

(2) Cognitive-Inspired Attention Mechanism: We propose a novel Forgotten Attention Algorithm (FAA) inspired by human cognitive processes, which progressively refines visual focus by iteratively "forgetting" irrelevant image regions, thereby enhancing alignment between question-guided attention and critical visual content.

(3) Bias Mitigation via Deliberate Forgetting: The FAA addresses language bias in VQA by systematically suppressing spurious question-answer correlations through multi-round attention refinement, forcing the model to rely on genuine multi-modal reasoning rather than linguistic shortcuts.

(4) Iterative Attention Refinement for Precision: Unlike conventional static attention mechanisms, our FAA dynamically narrows the model's focus across successive iterations, ensuring precise localization of question-relevant visual features while discarding distracting information.

(5) Extensive Experimental Validation: On VQA-CP v2, our enhancements in leveraging visual information led to optimal performance. Notably, without additional annotations, our approach attained a 21.32% improvement compared to the UpDn baseline model.

To address these challenges, our study focuses on the following research questions: RQ1: Can ensemble gradient modeling effectively identify and suppress linguistic, visual, and multimodal biases without relying on data augmentation? RQ2: Can an iterative forgetting mechanism,

What are the people doing?

Wrong Answer: Sitting

Right Answer: Cycling

Figure 2: Due to the presence of biases, the influence of the size of prominent objects in the image on model reasoning leads to incorrect answers, while the image regions relevant to the answers often occupy a small portion. FAA achieves this by masking irrelevant regions in the image, allowing the model to focus on image details for inference.

inspired by human cognition, progressively refine attention maps to enhance multimodal alignment and reasoning accuracy?

2 Related works

2.1 Visual question answering

The Visual Question Answering (VQA) task requires models to provide accurate answers to natural language questions based on visual content. Since its introduction, significant progress has been made in dataset development and multimodal fusion techniques. Benchmark datasets such as VQA v2 [18], GQA [19], and CLEVR [20] have driven advancements by providing large-scale annotated samples with balanced linguistic distributions. For more complex reasoning, OK-VQA [21] focuses on outside-knowledge-based questions, while VideoQA [22] extends the task to dynamic video understanding.

Current state-of-the-art approaches primarily adopt single-stream or dual-stream architectures [23–27]. Single-stream models (*e.g.*, ViLBERT [24], LXMERT [25]) employ unified transformers to jointly encode image-text pairs, enabling deep cross-modal interaction. In contrast, dual-stream methods (*e.g.*, MCAN [26], UNITER [27]) process visual and linguistic features separately before fusion, often achieving higher precision through modular design. These models benefit from large-scale pretraining on datasets like Conceptual Captions [28] or Visual Genome [29], which enhances their ability to align visual and textual representations.

However, despite their strong performance, these methods still face two critical challenges:

(1) Language Bias: Models tend to exploit statistical priors in training data (*e.g.*, favoring frequent answers like "yellow" for color questions) rather than grounding responses in visual evidence [30].

(2) Weak Visual Grounding: Attention mechanisms often focus on salient but irrelevant objects (*e.g.*, a "bench" dominating attention when the question concerns "people"), leading to incorrect reasoning paths [11].

Recent work attempts to address these issues through debias strategies (e.g., adversarial learning[31], counterfactual samples [7]) and enhanced atten tion mechanisms (e.g., compositional attention [15]).Yet, most methods either rely on costly data augment tation or fail to fully leverage visual cues. Such as HINT[32], SCR[33], and GGE-DQ[34] address VQA bias from different angles. HINT aligns model attention with human annotations, but relies on external supervision. SCR introduces causal regularization to suppress language priors, while GGE-DQ uses gradient-based ensemble training with dual questions, increasing computational cost. Our work bridges this gap by introducing a human-inspired iter ative attention refinement approach, eliminating language biases without external data.

2.2 Mitigate language bias

In recent research, researchers have proposed a range of debiasing methods to address language bias concerning existing defined bias issues. These methods include adversarial-based techniques [38], regularization approaches [4, 12, 39–41], and data augmentation strategies. Our approach focuses on addressing bias issues from the perspective of the visual modality.

The pervasive issue of language bias in VQA systems has been extensively documented in recent literature [4, 12]. These biases manifest when models exploit statistical regularities in question-answer pairings while neglecting genuine visual evidence, leading to compromised generalization capabilities, particularly on out-of-distribution datasets like VQA-CP [7]. Existing methodologies for mitigating language bias can be categorized into three primary paradigms:

(1) Adversarial-based Techniques: Methods such as [38] employ gradient reversal layers or auxiliary adversarial networks to explicitly disentangle language priors from visual reasoning pathways. While effective in

Table 1: Comparison of methods on VQA-CP v2 dataset

Method	Type	External Data/ Augmentation	VQA-CP v2 Acc(%)	Notes
RUBi [35]	Adversarial	No	44.23	Learns bias via gradient reversal
LMH [36]	Ensemble + Heur.	No	52.01	Uses language prior estimation
CSS [7]	Augmentation	Yes	58.95	Generates counterfactual samples
CF-VQA [37]	Regularization	No	53.55	Uses bias-weighted contrastive loss
GGE [30]	Gradient-based	No	57.32	Trains bias expert model
D-VQA [11]	Feature-level Aug	Yes	61.91	Debiasing in feature and sample space
Ours	Gradient + FAA	**No**	**62.78**	No extra annotation, no augmentation

theory, these approaches often face optimization challenges, as noted in [42], where the adversarial training dynamics can inadvertently suppress semantically relevant linguistic patterns along with harmful biases.

(2) Regularization Approaches: Recent advances in regularization-based debiasing employ specialized loss functions to mitigate language prior overfitting while preserving model generalizability. Representative approaches include: a question-only branch with entropy maximization to discourage shortcut learning [4], attention divergence constraints to ensure visual grounding [40], and a contrastive learning framework to distinguish between vision-relevant and bias-driven features [41]. These approaches demonstrate that structured regularization can effectively reduce linguistic bias without compromising model capacity, offering computationally efficient alternatives to adversarial training or data augmentation.

(3) Data Augmentation Strategies: Techniques like [7, 11, 43] generate counterfactual samples or synthetic QA pairs to break spurious correlations. The VQA-CP dataset [7] represents a notable example of this approach through systematic answer distribution inversion. However, such methods may introduce new biases during synthetic data generation while incurring significant annotation costs[43].

2.3 Attention mechanism

In the context of Visual Question Answering (VQA), attention mechanisms are employed to integrate information from different modalities [13], allowing models to focus on the most relevant regions between images and texts. Presently, attention-based methodologies include linear attention [44], co-attention [16], detection attention [13], and relational attention [45]. Consequently, in our approach, we explore the integration of attention mechanisms into debiasing methods in VQA, strengthening the model's retrieval capabilities between images and questions. Leveraging attention mechanisms enhances the role of visual information, ultimately aiding in debiasing strategies.

2.4 Summarize

To provide a clearer comparison, we summarize representative debiasing methods in Table 1, categorized by their core technique, reliance on external data, and performance on VQA-CP v2. This table highlights the novelty and advantage of our method in balancing performance and data efficiency. To provide a clearer comparison, we summarize representative debiasing methods in Table 1, categorized by their core technique, reliance on external data, and performance on VQA-CP v2. This table highlights the novelty and advantage of our method in balancing performance and data efficiency.

3 Method

We describe the architecture and algorithmic flow of FAA. As shown in Fig. 3, the left side illustrates the primary structure of the UpDn baseline model [13], responsible for extracting visual-language features. On the right side, there are stacked $Attention_Layers$ that iteratively mask irrelevant features and make answer predictions.

3.1 Visual information combination

On the left side of Fig. 3, we utilize the UpDn encoding layer to extract features. For a given text, the UpDn leverages a standard GRU to encode each question, generating a question vector. Regarding the provided image, UpDn uses the detected visual features as input. The visual feature set is represented as $F = \{f_1, ..f_i.., f_n\}$, where f_i denotes the feature of the i-th object in the image. In our method, we also incorporate factors such as spatial position. We re-encode all the outputs from Faster-RCNN [46] into new visual features. The visual input V is represented as Eq. (1):

$$V = Visual_Encoder(F, S, Cls, Ari) \tag{1}$$

where $Visual_Encoder$ represents the visual encoder responsible for re-encoding the four types of features into visual input. These four types of features are represented as visual feature vectors F, spatial features S, classification scores Cls, and attribute information Ari. During the initialization phase, this re-encoded visual data V is introduced as the visual input for the VQA process.

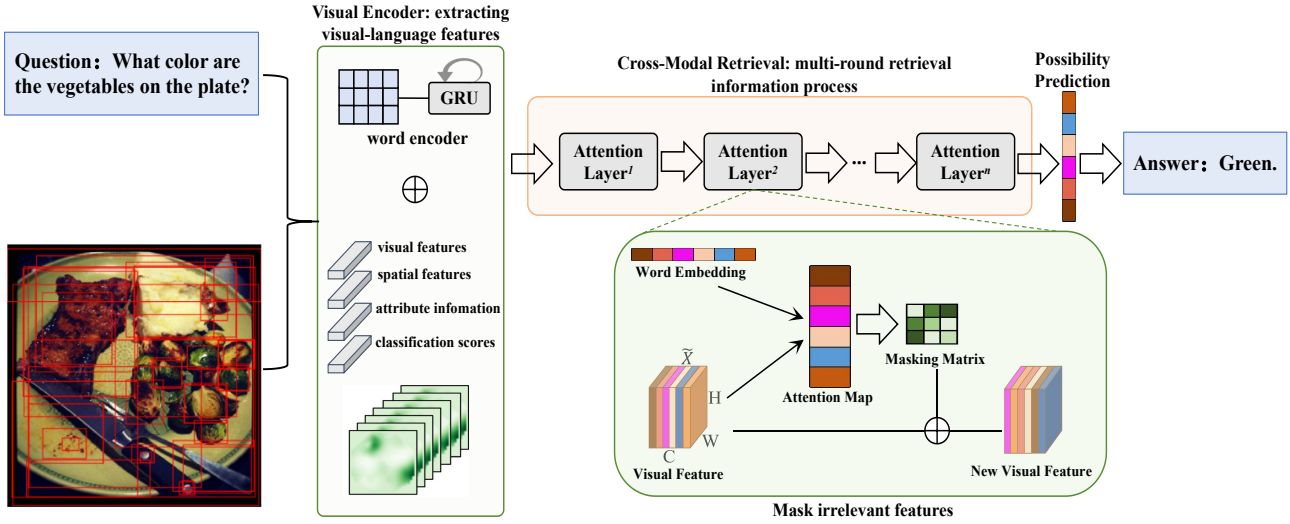


Figure 3: Our proposed FAA follows the architecture of the UpDn baseline model, comprising the feature extraction stage and the attention layers. The attention layers aim to retrieve information from the encoded question and image features, facilitating a multi-round retrieval process. In each round of retrieval, an image mask matrix is constructed to mask out the information deemed irrelevant by the model during this round, retaining crucial information for subsequent reasoning.

3.2 Bias definition

For the VQA, traditional methods typically treat it as a multi-class classification problem. The model generally takes a given triplet dataset $D = \{v_i, q_i, a_i\}_{i=1}^{N}$ as input, where the i-th image is $v_i \in \mathcal{V}$, the question is $q_i \in \mathcal{Q}$, and the answer is $a_i \in \mathcal{A}$. The main goal is to train a mapping that generates an accurate answer distribution over the answer set $\mathcal{A}$ based on the input data. In VQA-VS, the authors define nine different types of biases across three modalities, namely, linguistic, visual, and multimodal biases. In practice, we focus on the VS dataset as the primary research dataset, and for the biases corresponding to the three modalities, we follow the previous work by modeling each modality individually to capture biases within each modality. Furthermore, we treat bias in VQA as the absence of modality-specific reasoning during the inference process. Specifically, for the input sample pair (V, Q), we establish three bias models to perform bias feature modeling on the sample pair.

$$B_q = c_q(\sigma(q_i)) \tag{2}$$
$$B_v = c_v(\sigma(v_i)) \tag{3}$$
$$B_m = c_m(\sigma(m_i)) \tag{4}$$

where q_i, v_i, and m_i represent the question, image, and multimodal representation, respectively. In our implementation, the components c_q, c_v, and c_m are each implemented as two-layer MLPs with ReLU activation and Layer Normalization. These modules encode the question-only, vision-only, and joint features respectively to generate bias predictions B_q, B_v, and B_m for ensemble gradient analysis.

3.3 Bias detection

Firstly, we believe that in practical prediction tasks, a model must be able to identify different types of biases, which enables it to maintain strong generalization performance when handling diverse types of data. Additionally, in VQA, not all biases necessarily have a purely negative impact on the model. For instance, bias based on keywords or key objects may require the model to incorporate some additional real-world knowledge, which can act as supplementary information for making accurate predictions, rather than misleading the model. Therefore, in this paper, we propose a bias detector that not only assists the model in identifying biases but also estimates the weights of the detected biases to assess their influence on the model's performance.

Specifically, based on the bias feature results and definitions obtained in the previous step, we calculate the loss of the corresponding bias features to determine the type of bias the model is currently encountering.

$$Loss_q = \mathcal{L}(B_q, A) \tag{5}$$
$$Loss_v = \mathcal{L}(B_v, A) \tag{6}$$
$$Loss_m = \mathcal{L}(B_m, A) \tag{7}$$

where A represents the ground truth answers in the dataset, and $Loss_*$ denotes the loss calculated for the corresponding bias feature results. The reason for calculating the loss between the bias feature results and the ground truth labels in our method is that if the loss between the bias feature and the actual result is too large, it indicates that the predictions generated by the bias model are poor, meaning that the model has a weak ability to capture the bias. In the ensemble model approach, we aim to sufficiently train the bias model to achieve overfitting, so that the overall model develops a strong capacity for bias recognition.

In summary, in our method, we treat the modality with the largest $Loss_*$ as the one where the model exhibits bias, and we mark it accordingly.

$$F = B_* \tag{8}$$

Where F represents the bias flag. B_* represents the corresponding modality bias in $\max(Loss_q, Loss_v, Loss_m)$.

3.4 Bias score

After identifying the biases present in the sample pair, we use the bias detector to score the bias features, with the scores reflecting the degree of harm the bias imposes on the model (*i.e.*, whether the bias is beneficial or harmful). If the bias in the current sample aids the model's performance, there is no need to overfit to it during training. However, if the bias harms the model, it is necessary to ensure the model thoroughly learns the features corresponding to that bias during training. Accordingly, beneficial biases receive higher scores, while harmful biases are assigned lower scores.

$$score = 1 - softmax(m_c(F, A)) \tag{9}$$

where, F represents the bias flag obtained in the previous step, and m_c denotes the linear layer, which linearly fits the bias features corresponding to the flag with the labels. Finally, the bias score is computed through the softmax layer. In our implementation, the bias scoring function mc is a two-layer MLP with ReLU activation. Its input consists of the fused feature vector concatenated with a one-hot encoded bias type flag (length 3, indicating linguistic, visual, or multimodal bias). The output is a scalar score representing the confidence of bias activation. The MLP uses a hidden dimension of 512 with LayerNorm for stability. The bias score represents the softmax probability that the bias-specific features F support the ground truth label A. If the score is high (close to 1), it implies the bias aligns with the correct answer and is potentially helpful. If the score is low (close to 0), it indicates that the bias contradicts the correct label and is more likely to be harmful. We leverage this score to modulate the influence of gradient contributions: harmful biases (low score) are down-weighted, while benign ones (high score) are preserved. This design allows our model to adaptively suppress misleading bias without discarding all bias signals.

3.5 Fitting gradients

Based on the obtained scores, we follow the Boosting concept in our method by employing different classifiers to overfit the training data. For weak features associated with bias, we apply an overfitting approach to learn the bias distribution. By leveraging a biased model's gradient to train the base model, we aim to eliminate language bias in VQA tasks.

In training, based on the ensemble model concept, the main goal of the method is to minimize the gap between the predictions of the main model, bias models, and the true labels. Therefore, the three bias models in this method are defined as $M = \{M_v, M_q, M_m\}$, representing the visual, question, and multimodal bias models, respectively. The ultimate objective is to fit the prediction set of these three models, along with the main model, to the ground truth labels Y.

$$\min\mathcal{L}(\sigma(\sum_{i=v}^{v,q,m} M_i + f(v,q)), Y) \tag{10}$$

where, M represents the biased model, and f denotes the neural network. Our goal is to ensure that the biased parts of the dataset are fit exclusively through the biased model. In our method, we use the opposite direction of the gradient as pseudo-labels to guide the model in overfitting the biased training data. We differentiate between the effects of different types of biases on the model, and based on the bias scores obtained in the previous step, we scale the magnitude of the gradient during the gradient descent process. Simply put, higher scores correspond to beneficial biases in the training data, resulting in less decrease during backpropagation, as the model should leverage such biases to acquire knowledge. Conversely, for harmful biases, the gradient descent is amplified during backpropagation to reduce the model's reliance on such biases.

$$Y_{new} = -\nabla\mathcal{L}(f(v,q)) \tag{11}$$

$$\min\mathcal{L}(\sigma(\sum_{i=v}^{v,q,m} M_i + f(v,q)), s * Y_{new}) \tag{12}$$

where s represents the bias score. During the testing phase, we only use the main model for predictions, allowing the main model to handle the prediction of unbiased data under normal conditions. To clarify, Eq. 10 is the main objective applied to all samples, while Eq. 11 is an auxiliary loss used only for biased samples. These two losses are optimized jointly. Eq. 11 provides a targeted correction signal for harmful bias features by leveraging pseudo-gradients, scaled by bias scores. The combined optimization ensures both general supervision and bias suppression.

3.6 Attention layers

Each attention iteration in FAA corresponds to one instance of the Attention Layer stack, as implemented in Algorithm 1. Thus, the N-layer configuration of FAA represents N iterative refinement steps in attention focus. In the right side of Fig. 3, we have stacked N layers of $Attention_Layer$ to achieve visual information masking and retrieval. The number of Attention Layers (N) in FAA equals the number of iterative forgetting steps, where each layer corresponds to one round of attention refinement (masking irrelevant regions and updating focus)". Specifically, the $Attention_Layer$ module consists of three main components:

(1) **Initial Impression**. After obtaining visual and text features, the next step in our process is to employ

the $Self_Attention$ mechanism. This mechanism helps identify the most critical components within each modality, similar to how humans instinctively react when first encountering an image or text. We establish the model's initial assessment of the pivotal image regions and word vectors within the provided features. As formally delineated in Algorithm 1, this process can be mathematically defined as follows:

$$\begin{aligned} att_v &= Self(V) \\ att_q &= Self(Q) \\ V^1 &= att_v * V \\ Q^1 &= att_q * Q \end{aligned} \tag{13}$$

where att_v and att_q represent the initial attention. V^1 and Q^1 represent the features obtained after fusing the initial attention with the original data.

Algorithm 1: Forgetting Attention Algorithm

Input : Representation of Object Detection Outputs:$\mathcal{F}, \mathcal{S}, Cls, Ari$; Text coded representation: Q; Number of layers of attention stack: N; Attention threshold: α.
Output: Predicted answer probability:$\mathcal{A}$.

Initialize:$V \leftarrow [\mathcal{F}, \mathcal{S}, Cls, Ari]$, $k \leftarrow 3$.
Function *FAA(V, Q)*:
  while $n \leq N$ **do**
    $att_v, att_q \leftarrow SelfAttention(V, Q)$
    $V^1, Q^1 \leftarrow att_v \odot V, att_q \odot Q$
    $V^2, Q^2 \leftarrow CrossAttention(V^1, Q^1)$
    $Att \leftarrow V^2 \odot Q^2$
    if $Att \geq \alpha$ **then**
      $V_{mask} \leftarrow 1$;
    else
      $V_{mask} \leftarrow 0$;
    $V^3 \leftarrow V_{mask} \oplus V^2$
    $V, Q \leftarrow V^3, Q^2$
  $\mathcal{A} \leftarrow V^3, Q^2$
return $\mathcal{A}$

(2) **Cross-Modal Retrieval**. With the obtained features V^1 and Q^1, we consider using the Cross Attention mechanism [14] to explore information across modalities. This step is analogous to how humans associate objects with words. We perform cross-modal information retrieval separately in the image and text domains. This is defined as Eq. (14):

$$\begin{aligned} V^2 &= CrossAtt_{v \to q}(Q^1, V^1) \\ Q^2 &= CrossAtt_{q \to v}(V^1, Q^1) \end{aligned} \tag{14}$$

where V^2 and Q^2 represent the feature outputs after conducting cross-modal retrieval for the image and text, respectively. $CrossAtt$ respectively represents the cross-modal information retrieval layer, with "image" and "question" as the primary modalities.

(3) **Masking Matrix**. After Cross-Modal Retrieval, we calculate the masking matrix for V^2 and Q^2. Initially, we employ the Top-Down attention mechanism [13] to obtain an attention weight matrix: Att, which is then compared to a predefined threshold α to determine the masking matrix. As depicted in Algorithm 1, this is defined as Eq. (15):

$$\begin{aligned} Att &= V^2 \odot Q^2 \\ V_{mask} &= Mask(Att \geq \alpha) \\ V^3 &= V_{mask} \oplus V^2 \end{aligned} \tag{15}$$

where V_{mask} represents the masking matrix, and $Mask()$ denotes the process in which Att is compared to α in Algorithm 1. The value of α is determined by the mean of attention. $\oplus$ denotes linear fusion.

In the FAA, the threshold α is a critical parameter for controlling the "forgotten regions" in the attention distribution, used to mask image regions with attention scores below α in each iteration to focus on visual features relevant to the question. The definition of α supports two strategies: (1) a fixed threshold strategy, where an empirical value is determined by analyzing the distribution of attention scores (ranging from 0 to 1); for instance, we found that α = 0.6 performs optimally in balancing masking strength and preserving key features by examining sample attention scores; (2) a dynamic threshold strategy, where α is set as a hyperparameter and adaptively adjusted based on the dispersion of attention responses in each iteration to prevent over-forgetting or under-forgetting. This adaptive adjustment dynamically updates α by calculating the standard deviation of attention scores, ensuring the model effectively focuses on relevant regions across different scenarios.

V^3 represents the features obtained by merging the masking matrix with visual features. $\oplus$ denotes the linear fusion of two types of features.

Specifically, in each attention layer, we establish a masking matrix based on the magnitude of attention weights, which masks regions in the image that contribute less to the answer. Through N such attention layers, we allow the model to progressively identify precise regions with high relevance to the given question.

3.7 Training phase

Our model is designed to effectively identify and mitigate various biases that arise in VQA tasks. The process begins with the integration of multimodal features, where visual information is extracted using object detection techniques such as Faster R-CNN. These visual features, including object location, classification scores, and spatial attributes, are then combined with textual features derived from the question to form a comprehensive representation for each input pair. A key component of this phase is the Bias Detection

mechanism, as shown in Fig. 4. Here, the model evaluates and scores biases in both the linguistic and visual domains. The linguistic biases, such as word associations, are detected through a specialized language bias model, while visual biases are identified by analyzing the model's tendency to focus on irrelevant or dominant visual features. By dynamically assessing these biases, the model adjusts its learning process to ensure that it prioritizes unbiased information, allowing it to better reason across both modalities.

Furthermore, the FAA is employed to progressively filter out irrelevant image regions during training. This iterative attention mechanism helps the model to focus on the most relevant parts of the image that contribute to answering the question, simulating the human-like reasoning of gradually reducing focus to the most pertinent visual areas. The FAA works by masking less important image regions in each iteration, ensuring that the model refines its attention and concentrates on the critical visual details.

Through this training process, which integrates bias detection, bias scoring, and the FAA, the model becomes more robust and capable of handling complex, real-world VQA scenarios without relying on biased shortcuts.

4 Experimental analysis and discussion

In this section, we will present the quantitative and qualitative analysis results from the experiments.

4.1 Experimental setup

We adopt the following unified training setup: all models are trained for 30 epochs using the Adam optimizer with a learning rate of $1e^{-4}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The batch size is set to 256, and the cross-entropy loss is used as the objective function. All experiments are conducted on a single NVIDIA A100 32GB GPU, with an average training time of approximately 6 hours.

Dataset and evaluation metric. We validated our method on multiple VQA datasets: VQA v2, VQA-CP v2, VQA-VS, GQA, and VQA-CE. The VQA-CP v2 is employed to evaluate bias in VQA models across various scenarios, while GQA-OOD assesses the model's robustness on both ID and OOD data. VQA-CE and VQA-VS concentrate on multimodal biases, analyzing shortcut learning patterns to provide a more detailed evaluation of OOD performance. Standard evaluation on VQA v2 gauges the approach's generalizability. All test results are based on standard VQA evaluation metrics.

Baseline. We mainly used the UpDn[13] model as the baseline model.

4.2 Comparisons with SOTAs

We compared our method with other bias reduction methods, as shown in Table 2. First, we introduce the method names and their corresponding baseline models, followed by the experimental results on VQA-CP v2 and VQA v2, respectively. In Table 2, our method achieves state-of-the-art performance on VQA-CP v2 (62.78%). Moreover, compared to other methods using the same baseline models, our approach demonstrates superior performance across all three categories, particularly achieving the best results in the Num. and Other categories. On VQA v2, we also attain state-of-the-art performance (65.47%), ranking in the top three across all specific categories.

We further compared our method to approaches that utilize supplementary data. As seen, our method remains competitive against methods like CSS and D-VQA, which are trained on balanced datasets, even though such comparisons are considered unfair in [37].

As shown in Table 3, all reported results are derived from multiple independent replicate experiments to ensure reliability and reproducibility. Mean accuracy and standard deviation are computed over 5 random seeds. 95% confidence intervals are calculated using Student's t-distribution with 4 degrees of freedom. Two-sided t-tests are used to evaluate statistical significance against the UpDn baseline. $p < 0.05$ (*) indicates statistically significant improvement; $p < 0.01$ (**) indicates strong significance.

4.3 Multimodal debiasing experiments

Recent studies emphasize that VQA-CP v2 primarily addresses limited biases, particularly shortcuts between question types and answers. In contrast, VQA-VS reorganizes VQA v2 by incorporating three different forms of bias to tackle various incorrect correlations. Therefore, we conducted debiasing experiments on the VQA-VS benchmark, with the results presented in Table 4. Table 4 illustrates the performance of the EBGDA+FAA model on the VQA-VS benchmark, where it demonstrates superior performance in addressing linguistic, visual, and multimodal biases, outperforming most baseline models. Compared to the pretrained LXMERT model, EBGDA+FAA exhibits considerable robustness, effectively mitigating various biases in the VQA task.

4.4 Analysis of other metrics

In our approach, we aim to increase the role of visual content in reasoning. To assess its effectiveness, we use additional metrics. Table 5 reports the performance of FAA compared with reverse attention and linear decay. FAA achieves higher accuracy on all benchmarks, and we further introduce three auxiliary metrics to quantify its effect on bias suppression at the gradient level: CGR (Counterfactual Gradient Reduction): the average decrease in gradient magnitude for biased predictions after FAA masking. CGW

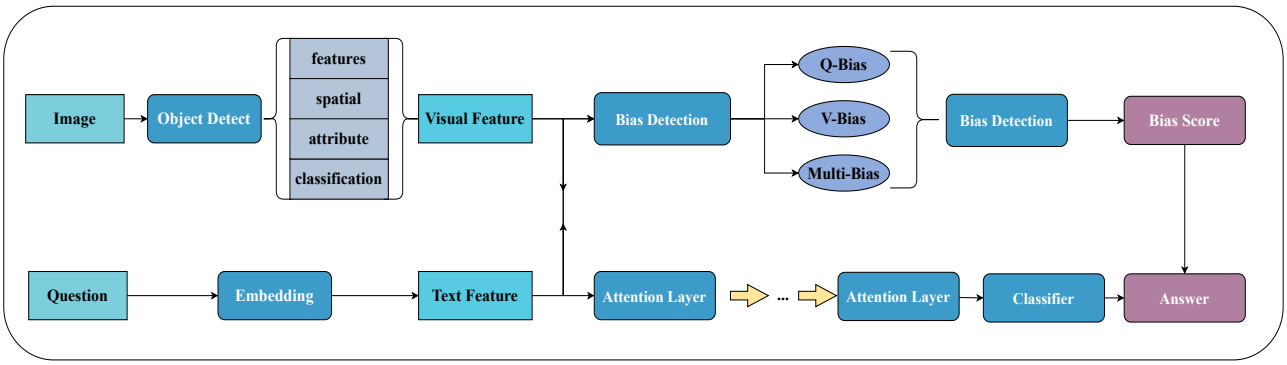


Figure 4: Training phase diagram

(Counterfactual Gradient Weighting): the normalized contribution of biased gradients in the overall weight update. CGD = CGR × CGW, reflecting the overall suppression effect on gradient-level bias flow. Compared to GGE [30], our approach performs better in terms of CGD, indicating improved utilization of visual information for answer prediction.

4.5 Results on GQA-OOD and VQA-CE

GQA-OOD and VQA-CE are new datasets and evaluation methods for VQA debiasing. Table 6 presents the experimental results of EBGDA+FAA. On GQA-OOD, EBGDA+FAA outperforms methods such as CSS, LMH, and RUBi. In VQA-CE, EBGDA+FAA achieves the highest accuracy in the Counter, validating its debiasing capability. These results strongly demonstrate the debiasing effectiveness of EBGDA+FAA.

4.6 Ablation experiments

The ablation experiment section verifies the performance of the EBGDA and FAA components in the method. EBGDA suppresses bias gradients at the feature level, while FAA refines visual focus via iterative masking; their combination enhances debiasing by addressing both feature-level and spatial-level biases.

The effectiveness of EBGDA. In this section, we conduct experiments using only the EBGDA component. The experimental results are shown in Table 7. By comparing EBGDA with methods that use the same baseline model, we can see that EBGDA also achieves performance improvements.

The effectiveness of FAA. In this section, we conduct experiments using only the FAA component. The experimental results are shown in Table 8. When comparing FAA with methods using the same baseline model, we can observe that FAA also achieves performance improvements.

As shown in Table 9, we compared the experimental results of FAA with those of other methods using attention mechanisms.

As shown in Table 10, these values confirm FAA introduces moderate overhead while significantly improving performance and focus alignment.

Forgotten Sequence. The concept of forgetting attention in this paper is based on the process of human answering relevant questions. The ablation experiment considers the order of forgetting in the algorithm to verify the validity of the concept. Table 11 shows the effect of three different sequences of attention on the performance of the model:

(1) In the method of this article, we follow the normal attention process, pay attention to the image areas that are most relevant to the problem, and forget the results obtained from the irrelevant areas.

(2) In contrast to the normal attention flow, the model first notices the areas of the image that are less relevant to the problem, assigns higher weights to them, and finally forgets the areas of the image with lower weights.

(3) The feature areas in the image are arranged in the original linear order, and the model forgets the relevant features in the same order.

By comparing the model performance of different forgetting sequences, we were able to observe that the FAA achieved excellent performance by forgetting irrelevant regions, while the other forgetting sequences resulted in decreased performance. This suggests that the human attentional forgetting mechanism on which the FAA is based works.

Attention Threshold Selection. In this method, we set thresholds to allow the model to filter image regions to determine which are forgotten and which are retained. Different threshold sizes make the model achieve different performance when forgetting the image region. If the threshold is too high, the model will forget most of the image region, resulting in the model being unable to obtain useful information from the image, while if the threshold is too low, the model will retain useless information, thus degrading the algorithm to a common attention algorithm. Therefore, this section sets up a comparative analysis of different threshold sizes to determine the appropriate parameter as the forgetting threshold. As shown in Table 12, the corresponding

Table 2: The results of VQA-CP v2 test set and VQA v2 validation set are presented in the following table. Each column illustrates the **Best** performances of each method, excluding data augmentation techniques. Our method has been compared with state-of-the-art methods on both datasets

Method \ Dataset	Base	VQA-CP v2 test				VQA v2 val			
		All	Y/N	Num.	Other	All	Y/N	Num.	Other
UpDn[13]	-	39.96	43.01	12.07	45.82	63.48	81.18	42.14	55.66
AdvReg[47]	UpDn	41.17	65.49	15.48	35.48	62.75	79.84	42.35	55.16
RUBi[35]	UpDn	44.23	67.05	17.48	39.61	-	-	-	-
LMH[36]	UpDn	52.01	72.58	31.12	46.97	56.35	65.06	37.63	54.69
AdaVQA[48]	UpDn	54.67	72.47	53.81	45.58	-	-	-	-
CF-VQA[37]	UpDn	53.55	**91.15**	13.03	44.97	63.54	82.51	43.96	54.30
CIKD[49]	UpDn	54.05	90.01	15.10	45.88	61.29	76.34	40.20	55.43
GGE[30]	UpDn	57.32	87.04	27.75	49.59	59.11	73.27	39.99	54.39
D-VQA[11]	UpDn	61.91	88.93	52.32	50.39	64.96	82.18	44.05	57.54
GGD[34]	UpDn	59.37	88.23	38.11	49.82	62.15	79.25	42.43	54.66
GenB[40]	UpDn	59.15	88.03	40.05	49.25	62.74	**86.18**	43.85	47.03
Template-based[50]	UpDn	39.75	43.03	14.98	44.83	63.83	81.61	41.98	56.10
ASL[51]	UpDn	46.00	58.24	29.49	44.33	-	-	-	-
CVL[39]	UpDn	42.12	45.72	12.45	48.34	-	-	-	-
GradSup[52]	UpDn	46.80	64.50	15.30	45.90	-	-	-	-
RangImg[53]	UpDn	55.37	83.89	41.60	44.20	57.24	76.53	33.87	48.57
CSS[7]	UpDn	58.95	84.37	49.42	48.24	59.91	73.25	39.77	55.11
Mutant[9]	UpDn	61.72	88.90	49.68	50.78	62.56	82.07	42.52	53.28
Unshuffling[54]	UpDn	43.47	47.82	14.35	49.18	43.47	81.99	43.07	55.21
SimpleAug[55]	UpDn	52.65	66.40	43.43	47.98	64.34	81.97	43.91	56.35
ECD[56]	LMH	59.92	83.23	52.29	49.71	57.38	69.06	35.74	54.25
KDDAug[57]	UpDn	60.24	86.13	**55.08**	48.08	62.86	80.55	41.05	55.18
AttReg[58]	UpDn	46.75	66.23	11.94	46.09	64.13	81.72	43.77	56.13
SSL[59]	UpDn	57.59	86.53	29.87	50.03	-	-	-	-
CSS+CL[60]	UpDn	40.49	42.90	12.44	46.93	-	-	-	-
MMBS[61]	UpDn	48.19	65.00	14.05	48.75	63.84	79.61	44.23	57.05
LSP[6]	UpDn	61.95	89.50	52.44	50.12	65.26	82.38	44.77	57.67
DM[62]	UpDn	61.13	88.13	45.98	51.13	63.53	81.09	39.61	56.52
GVQA[63]	UpDn	31.30	57.99	13.68	22.14	48.24	72.03	31.17	34.65
AttAlign[32]	UpDn	39.37	43.02	11.89	45.00	63.24	80.99	42.55	55.22
HINT[32]	UpDn	46.73	67.27	10.61	45.88	63.38	81.18	42.99	55.56
SCR[33]	UpDn	49.45	72.36	10.93	48.02	62.2	78.8	41.6	54.5
DLR[31]	UpDn	48.87	70.99	18.72	45.57	57.96	76.82	39.33	48.54
LPF[64]	UpDn	51.57	87.33	12.25	43.61	62.63	79.51	42.90	55.02
CCB[65]	UpDn	57.99	86.41	45.63	48.76	60.73	78.37	36.88	53.17
Loss-Rescaling[66]	UpDn	47.09	68.42	21.71	42.88	55.50	64.22	39.61	53.09
RMLVQA[67]	UpDn	60.41	89.98	45.96	48.74	59.99	76.68	37.54	53.2
LRLGC[68]	UpDn	60.91	89.95	45.13	50.03	60.81	77.65	39.25	53.71
Ours	UpDn	**62.78**	87.90	51.74	**52.79**	**65.47**	82.51	**57.84**	**58.64**

experimental results of the three threshold sizes selected by us are reported.

Considering the three different choices of attention threshold, in the original attention scheme of UpDn, the contribution degree of different image regions to the answer is realized by the assigned weight, whose value is between 0 and 1. Therefore, we choose the sizes of 0.5, 0.6 and 0.7 for experimental comparison. The final experimental results show that when the threshold size is 0.6, the experimental effect reaches the optimal performance, and the model achieves a balanced trade-off in attention precision. Additionally, we also present the approach using a dynamic threshold, where the threshold is treated as a hyperparameter and optimized during training.

To further justify the choice of dynamic threshold $\alpha = 0.6$, we present two analytical curves in Fig. 5. The left subplot shows that as α increases, the recall of relevant visual regions steadily declines, indicating that overly aggressive masking may suppress informative content. Conversely, the right subplot shows that model accuracy improves with moderate filtering, peaking around $\alpha = 0.6$, and then drops as too much information is retained. These trends suggest

Table 3: Comparison of methods across different datasets with accuracy, 95% confidence intervals, *p*-value, and significance levels

Method	Dataset	Accuracy (%)	95% CI (%)	*p*-value (vs UpDn)	Significance
EBGDA	VQA-CP v2	60.10 ± 0.26	[59.81, 60.39]	0.0004	**
	VQA v2	64.01 ± 0.23	[63.75, 64.27]	0.043	*
	VQA-VS	56.44 ± 0.21	[56.21, 56.67]	0.006	**
	GQA-OOD	52.68 ± 0.22	[52.44, 52.92]	0.012	*
	VQA-CE	58.42 ± 0.21	[58.19, 58.65]	0.0012	**
FAA	VQA-CP v2	60.18 ± 0.31	[59.83, 60.53]	0.0003	**
	VQA v2	63.87 ± 0.22	[63.63, 64.11]	0.048	*
	VQA-VS	56.52 ± 0.24	[56.26, 56.78]	0.004	**
	GQA-OOD	52.60 ± 0.20	[52.38, 52.82]	0.017	*
	VQA-CE	58.17 ± 0.19	[57.95, 58.39]	0.0024	**
EBGDA+FAA	VQA-CP v2	62.78 ± 0.28	[62.49, 63.07]	0.0001	**
	VQA v2	64.39 ± 0.25	[64.09, 64.69]	0.026	*
	VQA-VS	57.90 ± 0.20	[57.68, 58.12]	0.002	**
	GQA-OOD	53.74 ± 0.23	[53.47, 54.01]	0.006	**
	VQA-CE	59.26 ± 0.18	[59.06, 59.46]	0.0009	**

Table 4: Regarding the experimental outcomes of EBGDA+FAA on the VQA-VS dataset, we have presented the relevant experimental performance reports associated with this dataset. Each column displays the performance results of the corresponding best and second-performing models

Model	Base	VQA-VS OOD Test Sets									
		Language-based				Visual-based		multi-modality			mean
		QT	KW	KWP	QT+KW	KO	KOP	QT+KO	KW+KO	QT+KW+KO	
UpDn		32.43	45.10	56.06	55.29	33.39	41.31	46.45	54.29	56.92	46.80
+LMH	UpDn	33.36	43.97	54.76	53.23	33.72	41.39	46.15	51.14	54.97	45.85
BAN	UpDn	33.75	46.64	58.36	57.11	34.56	42.45	47.92	56.26	59.77	48.53
LXMERT	-	**36.46**	**51.95**	**64.17**	**64.22**	**37.69**	**46.40**	**53.54**	**62.46**	**67.44**	**53.70**
EBGDA+FAA	UpDn	32.45	47.83	59.27	60.59	34.75	43.98	44.47	56.69	55.6	51.33

Table 5: Experiment on the evaluation metric CGD using the FAA method on the VQA-CP v2 dataset. **Best** results are displayed in each column

Method	CGR	CGW	CGD
UpDn	44.27	40.63	3.91
RUBi	39.60	33.33	6.27
CSS	**46.70**	37.89	8.87
GGE-DQ-iter	44.35	27.91	16.44
GGE-DQ-tog	42.74	**27.47**	15.27
EBGDA+FAA	45.09	27.54	**17.56**

that $\alpha = 0.6$ achieves a well-balanced trade-off between visual focus (selectivity) and reasoning completeness (coverage), supporting the empirical results reported in Table 12.

Visual Information. All experiments in Table 13 are conducted using the UpDn baseline model with modified visual input features. The purpose is to isolate and evaluate the contribution of diverse visual inputs independent of debiasing components (FAA and EBGDA). As shown in Table 13, While object-only features offer a reasonable baseline, adding semantic attributes and classification logits consistently improves performance across all question types, especially in the "Other" category where visual grounding is more complex. The best configuration (Obj + Attr + Cls) achieves a +2.62% gain in All accuracy and +3.51% in Other compared to object-only features.

Layers of Attention. As shown in Table 14, we set up different levels of attention in the method to perform validation experiments. Specifically, when humans answer questions by focusing on different areas in the image, they may go through multiple target shifts to determine the final area, while in the simulation of a computer, this operation can be achieved by setting the number of layers of attention. In this experiment, we set up a total of five different layers, as you can see the model performs the best with two attention layers. This configuration significantly improves performance during inference compared to fewer layers. However, increasing the number of layers yields a slightly worse overall accuracy as well as increasing inference time by nearly 200 seconds. Regarding accuracy degradation, we believe that this phenomenon arises due to the fact that the model rec-

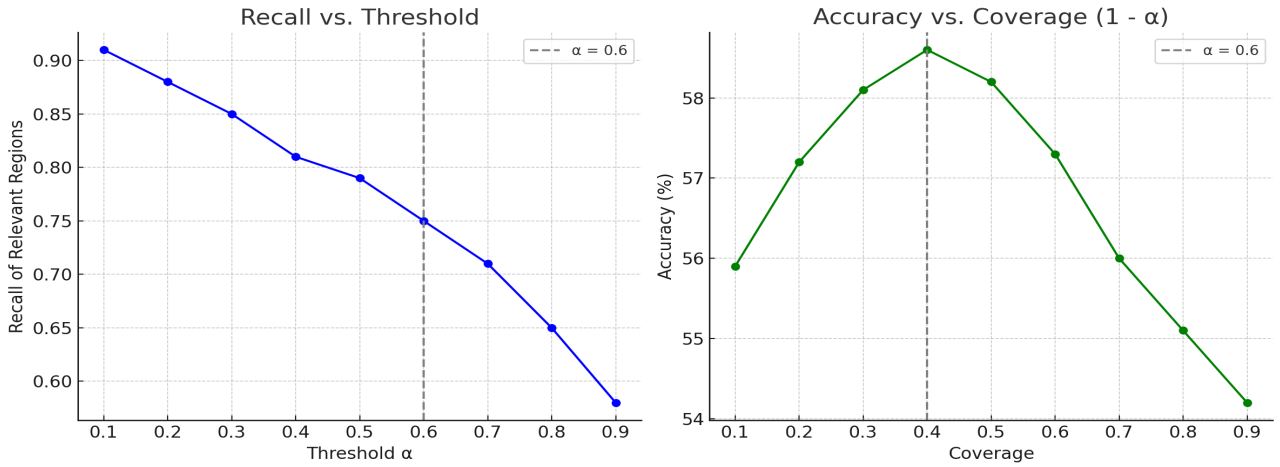


Figure 5: (Left) Recall vs. Threshold α: As the threshold α increases, the model becomes more selective, leading to a gradual decrease in the recall of question-relevant visual regions. (Right) Accuracy vs. Coverage: Coverage denotes the proportion of retained visual tokens. Model accuracy peaks around $\alpha = 0.6$, indicating a favorable balance between retaining useful information and filtering out noise.

Table 6: Experimental results on GQA-OOD and VQA-CE. Compared to the Best methods, EBGDA+FAA achieves the best performance on GQA-OOD. On VQA-CE, EBGDA+FAA also demonstrates the best performance in the Counter

Model	GQA-OOD Test			VQA-CE	
	All	Tail	Head	Counter	Easy
UpDn	46.87	42.13	49.16	33.91	**76.69**
RUBi	45.85	43.37	47.37	32.25	75.03
LMH	43.96	40.73	45.93	34.26	73.13
CSS	44.24	41.20	46.11	34.36	62.08
RandImg	-	-	-	34.41	76.21
Genb	49.43	45.63	51.76	34.80	68.15
EBGDA+FAA	**51.71**	**46.54**	**54.29**	**35.93**	63.98

ognizes incorrect visual information and masks relevant regions, thus hindering accurate answer retrieval. With three layers, the model may excessively mask relevant regions due to cumulative attention filtering, leading to information loss and accuracy degradation

Q-CSS. In our approach, we opted for a single-word replacement strategy, combined with FAA. The experimental results in Table 15 encompass a partial replication of the Q-CSS strategy from the CSS method and the QV-CSS strategy, incorporating FAA into both Q-CSS and CSS. Notably, our approach exhibits approximately 2% improvement in accuracy over Q-CSS and CSS.

Table 7: Ablation experiments with EBGDA

Method	All	Yes/No	Num.	Other	Δ
UpDn	$39.96_{\pm 0.18}$	43.01	12.07	45.82	-
SAN	$38.65_{\pm 0.20}$	40.09	12.98	44.67	-1.31
BAN	$35.94_{\pm 0.22}$	40.39	12.24	40.51	-4.02
S-MRL	$38.46_{\pm 0.25}$	42.85	12.81	43.20	-1.50
EBGDA	$60.10_{\pm 0.26}$	86.99	36.72	52.42	+20.14

Table 8: Ablation experiments with FAA

Method	All	Yes/No	Num.	Other	Δ
UpDn	$39.96_{\pm 0.18}$	43.01	12.07	45.82	-
SAN	$38.65_{\pm 0.20}$	40.09	12.98	44.67	-1.31
BAN	$35.94_{\pm 0.22}$	40.39	12.24	40.51	-4.02
S-MRL	$38.46_{\pm 0.25}$	42.85	12.81	43.20	-1.50
FAA	$60.18_{\pm 0.31}$	83.27	37.82	54.21	+20.22

Table 9: Comparison of methods on VQA-CP v2 dataset

Method	VQA-CP v2 (%)
UpDn(baseline)	39.96
SCR	49.45
HINT	46.73
FAA(Ours)	60.18

Table 10: Placeholder caption for the performance metrics comparison

Metric	UpDn	FAA (N=3)	Overhead
Training Time/ epoch	23.1 min	26.6 min	+15.2%
Inference Time/ sample	58.0 ms	65.4 ms	+12.7%
Peak GPU memory usage	14.6 GB	16.5 GB	+13.0%

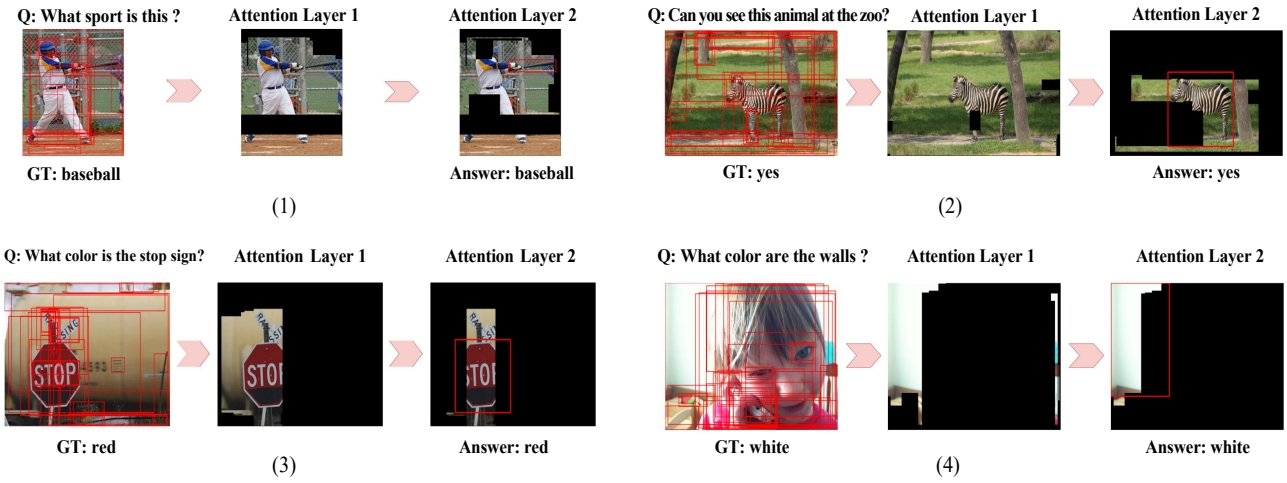


Figure 6: The results of qualitative analysis show the flow of our model when making predictions by masking different image regions so that the model focuses on the effective ones

Table 11: Impact of forgetten order on performance

		VQA-CP v2 test			
Order	Base	All	Y/N	Num.	Other
FAA	UpDn	60.74	83.99	41.45	53.85
Reverse	UpDn	37.86	78.00	15.34	23.00
Linear	UpDn	27.13	71.99	6.18	9.37

Table 12: Performance corresponding to different attention thresholds

		VQA-CP v2 test			
Thresholds	Base	All	Y/N	Num.	Other
0.5	UpDn	56.84	83.29	43.87	46.54
0.6	UpDn	60.18	83.99	41.45	53.85
0.7	UpDn	59.75	82.85	35.21	54.38
dynamic	UpDn	61.65	85.37	40.68	54.98

4.7 Qualitative results

As depicted in Fig. 6, the original image, after two rounds of attentional operations, masks out irrelevant areas based on attentional weights in the Fig. 6 (1), ultimately identifying the target region relevant to the answer.

In Fig. 6, more examples are given to analyze the effect of forgotten attention on changes in image areas. For example, in the example of the Fig. 6 (2), the image of the animal is the area where the zebra is located, and there is overlap between some areas that are unrelated to the problem and the zebra, which is covered by the FAA to some extent, but most of the zebra area is still captured by the model. Similarly, in the Fig. 6 (3) and Fig. 6 (4), the areas of the sign are somewhat obscured, but the model retains key semantic features in the remaining areas, enabling accurate semantic inference.

5 Conclusion

In this paper, we proposed a novel debiasing approach for Visual Question Answering by leveraging an ensemble model framework to address and mitigate various types of biases, including linguistic, visual, and multimodal biases. Through the integration of a bias detection mechanism, we dynamically identified and evaluated the impact of biases on the model's performance. By selectively applying overfitting strategies to biased components, the proposed method allows the main model to concentrate on unbiased and informative data, thereby enhancing its generalization ability across diverse scenarios. Our experiments on VQA-CP v2, VQA v2, and VQA-VS benchmarks demonstrated state-of-the-art performance, especially in challenging scenarios involving complex multimodal correlations. This approach provides an effective solution for reducing bias in VQA tasks, leading to more robust and accurate predictions.

6 Discussion

This section provides an in-depth analysis and discussion of our proposed FAA+EBGDA approach. First, compared with the LMH method, our approach does not rely on heuristic strategies based on language prior scores. Instead, we achieve more adaptive bias modeling by introducing multimodal bias detection and gradient modulation. Unlike methods such as CSS and Mutant, which depend on external adversarial samples, our approach operates without any data augmentation or external annotations, offering stronger generalizability and deployment flexibility. Second, the Forgotten Attention Algorithm (FAA) effectively suppresses visually salient but question-irrelevant regions in the image through a multi-round attention filtering mechanism. This helps avoid "saliency traps" that may mislead the model's decision-making, thereby enhancing both vi-

Table 13: Comparison of visual features across different VQA question categories (Yes/No, number, other)

Visual Features	All (%)	Yes/No	Number	Other
Object (Obj)	57.12 ± 0.21	82.14	35.47	50.91
Attribute (Attr)	58.03 ± 0.19	82.31	35.81	52.14
Classification (Cls)	58.65 ± 0.23	83.02	36.04	52.88
Obj + Attr	58.83 ± 0.22	83.15	36.19	53.01
Obj + Cls	59.02 ± 0.25	83.28	36.41	53.38
Attr + Cls	59.16 ± 0.20	83.36	36.52	53.63
Obj + Attr + Cls	**59.74 ± 0.24**	**83.85**	**37.04**	**54.42**

Table 14: The performance under different number of attention layers

Layers	All	Yes/No	Num.	Other	Time/Epoch
1	56.95	83.77	42.99	46.72	1048s
2	60.74	83.99	41.45	53.85	1303s
3	57.67	82.66	46.44	47.66	1380s
4	58.56	80.81	50.68	47.26	1532s
5	56.21	84.48	35.66	47.03	1540s

Table 15: Ablation experiments on the combination of FAA with Q-CSS and CSS methods

Method	All	Yes/No	Num.	Other
Q-CSS	56.19	80.83	40.33	47.63
CSS	58.17	84.57	46.99	47.40
FAA+Q-CSS	58.31	80.83	48.90	49.10
FAA+CSS	60.09	88.55	53.16	47.09

sual focus and multimodal alignment. In terms of computational overhead, our method introduces approximately a 15% increase in training cost compared to the standard UpDn model. During inference, with a two-layer attention configuration, the overhead remains manageable. Detailed analysis is provided in Table 10. Finally, while our method achieves competitive performance without relying on large-scale pretraining, it still has limitations when handling questions requiring external knowledge or complex reasoning. In the future work, we can explore integrating our approach with pretrained language models to further enhance its capabilities.

Data availability statement

Data will be made available on request.

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Qiuying Han, Peng Wang, Boyuan Li and Qiwen Lu. The first draft of the manuscript was written by Qiuying Han and Shaohui Zhang. All authors read and approved the final manuscript.

Funding

This work was supported by the National Natural Science Foundation of China (62172457), Science and Technology Research Project in Henan Province of China (242102210104, 252102210032), Key Scientific Research Project in Henan University of China (25B520021).

Abbreviations

The following abbreviations are used in this paper:

VQA	Visual Question Answering
FAA	Forgotten Attention Algorithm
EBGDA	Ensemble Bias Gradient Debiasing Approach
GRU	Gated Recurrent Unit
UpDn	Bottom-Up and Top-Down Attention Model

References

[1] Drew Hudson and Christopher D Manning. Learning by abstraction: The neural state machine. *Advances in Neural Information Processing Systems*, 32, 2019.

[2] Haonan Luo, Guosheng Lin, Yazhou Yao, Fayao Liu, Zichuan Liu, and Zhenmin Tang. Depth and video segmentation based visual attention for embodied question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):6807–6819, Jun 2023.

[3] Jingyao Wang, Manas Ranjan Pradhan, and Nallappan Gunasekaran. Machine learning-based human-robot interaction in its. *Information Processing and Management*, 59(1):102750, Jan 2022.

[4] Xinzhe Han, Shuhui Wang, Chi Su, Qingming Huang, and Qi Tian. Greedy gradient ensemble for robust visual question answering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1564–1573, Los Alamitos, CA, USA, 2021. IEEE Computer Society.

[5] Yibing Liu, Yangyang Guo, Jianhua Yin, Xuemeng Song, Weifeng Liu, Liqiang Nie, and Min Zhang. Answer questions with right image regions: A visual attention regularization approach. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 18(4):1–18, 2022.

[6] Jin Liu, ChongFeng Fan, Fengyu Zhou, and Huijuan Xu. Be flexible! learn to debias by sampling and prompting for robust visual question answering. *Information Processing and Management*, 60(3):103296, 2023.

[7] Long Chen, Yuhang Zheng, Yulei Niu, Hanwang Zhang, and Jun Xiao. Counterfactual samples synthesizing and training for robust visual question answering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13218–13234, 2023.

[8] Zujie Liang, Weitao Jiang, Haifeng Hu, and Jiaying Zhu. Learning to contrast the counterfactual samples for robust visual question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3285–3292, Online, November 2020. Association for Computational Linguistics.

[9] Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. MUTANT: A training paradigm for out-of-distribution generalization in visual question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 878–892. Association for Computational Linguistics, 2020.

[10] Vedika Agarwal, Rakshith Shetty, and Mario Fritz. Towards causal vqa: Revealing and reducing spurious correlations by invariant and covariant semantic editing. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9687–9695, Los Alamitos, CA, USA, 2020. IEEE Computer Society.

[11] Zhiquan Wen, Guanghui Xu, Mingkui Tan, Qingyao Wu, and Qi Wu. Debiased visual question answering from feature and sample perspectives. In *35th International Conference on Neural Information Processing Systems (NeurIPS)*, pages 3784–3796, Red Hook, NY, USA, 2021. Curran Associates Inc.

[12] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12695–12705, 2021.

[13] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6077–6086, Los Alamitos, CA, USA, 2018. IEEE Computer Society.

[14] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[15] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6274–6283, 2019.

[16] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *30th International Conference on Neural Information Processing Systems(NeurIPS)*, pages 289–297, Barcelona, Spain, 2016.

[17] Pan Lu, Hongsheng Li, Wei Zhang, Jianyong Wang, and Xiaogang Wang. Co-attending free-form regions and detections with multi-modal multiplicative feature embedding for visual question answering. In *the AAAI Conference on Artificial Intelligence*, 2022.

[18] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433, Chile, 2015. IEEE Computer Society.

[19] Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6693–6702, 2019.

[20] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2017.

[21] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3195–3204, 2019.

[22] Kewei Tu, Meng Meng, Mun Wai Lee, Tae Eun Choe, and Song Chun Zhu. Joint video and text parsing for

understanding events and answering queries. *IEEE MultiMedia*, 21(02):42–70, 2014.

[23] Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. End-to-end open-domain question answering with bertserini. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 72–77, Minneapolis, Minnesota, 2019. Association for Computational Linguistics.

[24] Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. Multi-passage bert: A globally normalized bert model for open-domain question answering. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5878–5882, Hong Kong, China, 2019. Association for Computational Linguistics.

[25] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *16th Conference of the European Chapter of the Association for Computational Linguistics*, pages 874–880. Association for Computational Linguistics, 2021.

[26] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *56th Annual Meeting of the Association for Computational Linguistics*, pages 784–789, Melbourne, Australia, 2018. Association for Computational Linguistics.

[27] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *16th European Conference on Computer Vision*, pages 104–120, 2020.

[28] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jan 2018.

[29] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, page 32–73, May 2017.

[30] Xinzhe Han et al. Greedy gradient ensemble for robust visual question answering. In *ICCV*, pages 1584–1593, 2021.

[31] Chenchen Jing, Yuwei Wu, Xiaoxun Zhang, Yunde Jia, and Qi Wu. Overcoming language priors in vqa via decomposed linguistic representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11181–11188, 2020.

[32] R. R. Selvaraju et al. Taking a hint: Leveraging explanations to make vision and language models more grounded. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2591–2600, 2019.

[33] Jialin Wu et al. Self-critical reasoning for robust visual question answering. *Advances in Neural Information Processing Systems*, 32, 2019.

[34] Xinzhe Han, Shuhui Wang, Chi Su, Qingming Huang, and Qi Tian. General greedy de-bias learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[35] Remi Cadene et al. Rubi: Reducing unimodal biases for visual question answering. In *NeurIPS*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[36] Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4069–4082, Hong Kong, China, November 2019. Association for Computational Linguistics.

[37] Yulei Niu et al. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12700–12710, 2021.

[38] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 1548–1558, Red Hook, NY, USA, 2018. Curran Associates Inc.

[39] Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Antonvanden Hengel. Counterfactual vision and language learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10041–10051, 2020.

[40] JaeWon Cho, Dong-Jin Kim, Hyeonggon Ryu, and InSo Kweon. Generative bias for robust visual question answering. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11681–11690, 2023.

[41] Abhipsa Basu, Sravanti Addepalli, and R.Venkatesh Babu. Rmlvqa: A margin loss approach for visual question answering with language biases. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11671–11680, 2023.

[42] Gabriel Grand and Yonatan Belinkov. Adversarial regularization for visual question answering: Strengths, shortcomings, and side effects. *arXiv preprint arXiv:1906.08430*, 2019.

[43] Tianyu Huai, Shuwen Yang, Junhang Zhang, Jiabao Zhao, and Liang He. Debiased visual question answering via the perspective of question types. *Pattern Recognition Letters*, 178:181–187, 2024.

[44] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21–29, Los Alamitos, CA, USA, 2016. IEEE Computer Society.

[45] Chenfei Wu, Jinlai Liu, Xiaojie Wang, and Xuan Dong. Object-difference attention: A simple relational attention for visual question answering. In *26th ACM international conference on Multimedia*, pages 519–527, 2018.

[46] Shaoqing Ren et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1137–1149, Jun 2017.

[47] Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. *Neural Information Processing Systems*, 2018.

[48] Yangyang Guo, Liqiang Nie, Zhiyong Cheng, Feng Ji, Ji Zhang, and Alberto Del Bimbo. Adavqa: Overcoming language priors with adapted margin cosine loss. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*, pages 708–714, Montreal, 2021. ijcai.org.

[49] Yonghua Pan, Zechao Li, Liyan Zhang, and Jinhui Tang. Distilling knowledge in causal inference for unbiased visual question answering. In *Proceedings of the 2nd ACM International Conference on Multimedia in Asia*, page 1–7, Mar 2021.

[50] Kushal Kafle, Mohammed Yousefhussien, and Christopher Kanan. Data augmentation for visual question answering. In *Proceedings of the 10th International Conference on Natural Language Generation*, Jan 2017.

[51] Damien Teney and Antonvanden Hengel. Actively seeking and learning from live data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1940–1949, Jun 2019.

[52] Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision – ECCV 2020,Lecture Notes in Computer Science*, page 580–599, Jan 2020.

[53] Damien Teney, Kushal Kafle, Robik Shrestha, Ehsan Abbasnejad, Christopher Kanan, and Antonvanden Hengel. On the value of out-of-distribution testing: An example of goodhart's law. *arXiv: Computer Vision and Pattern Recognition*, May 2020.

[54] Damien Teney, Ehsan Abbasnejad, and Anton van den Hengel. Unshuffling data for improved generalization in visual question answering. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021.

[55] Jihyung Kil, Cheng Zhang, Dong Xuan, and Wei-Lun Chao. Discovering the unknown knowns: Turning implicit knowledge in the dataset into explicit training examples for visual question answering. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6346–6361. Association for Computational Linguistics, 2021.

[56] Camila Kolling, Martin More, Nathan Gavenski, Eduardo Pooch, Otavio Parraga, and RodrigoC. Barros. Efficient counterfactual debiasing for visual question answering. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan 2022.

[57] Long Chen, Yuhang Zheng, and Jun Xiao. Rethinking data augmentation for robust visual question answering. In *ECCV*, pages 95–112, Berlin, Heidelberg, 2022. Springer, Springer-Verlag.

[58] Yibing Liu, Yangyang Guo, Jianhua Yin, Xuemeng Song, Weifeng Liu, Liqiang Nie, and Min Zhang. Answer questions with right image regions: A visual attention regularization approach. *ACM Transactions on Multimedia Computing, Communications, and Applications*, page 1–18, Nov 2022.

[59] Xi Zhu, Zhendong Mao, Chunxiao Liu, Peng Zhang, Bin Wang, and Yongdong Zhang. Overcoming language priors with self-supervised learning for visual question answering. *arXiv preprint arXiv:2012.11528*, 2020.

[60] Zujie Liang, Weitao Jiang, Haifeng Hu, and Jiaying Zhu. Learning to contrast the counterfactual samples

for robust visual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Jan 2020.

[61] Qingyi Si, Yuanxin Liu, Fandong Meng, Zheng Lin, Peng Fu, Yanan Cao, Weiping Wang, and Jie Zhou. Towards robust visual question answering: Making the most of biased samples via contrastive learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, Jan 2022.

[62] Yike Wu, Yu Zhao, Shiwan Zhao, Ying Zhang, Xiaojie Yuan, Guoqing Zhao, and Ning Jiang. Overcoming language priors in visual question answering via distinguishing superficially similar instances. *arXiv preprint arXiv:2209.08529*, 2022.

[63] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 4971–4980, USA, 2018. Computer Vision Foundation / IEEE Computer Society.

[64] Zujie Liang, Haifeng Hu, and Jiaying Zhu. Lpf: A language-prior feedback objective function for debiased visual question answering. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul 2021.

[65] Chao Yang, Su Feng, Dongsheng Li, Huawei Shen, Guoqing Wang, and Bin Jiang. Learning content and context with language bias for visual question answering. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*, Jul 2021.

[66] Yangyang Guo, Liqiang Nie, Zhiyong Cheng, Qi Tian, and Min Zhang. Loss-rescaling vqa: Revisiting language prior problem from a class-imbalance view. *IEEE Transactions on Image Processing*, page 227–238, Jan 2022.

[67] Abhipsa Basu, Sravanti Addepalli, and R. Venkatesh Babu. RMLVQA: A margin loss approach for visual question answering with language biases. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 11671–11680, Canada, 2023. IEEE.

[68] Runlin Cao and Zhixin Li. Overcoming language priors for visual question answering via loss rebalancing label and global context. In Robin J. Evans and Ilya Shpitser, editors, *Uncertainty in Artificial Intelligence, UAI 2023, July 31 - 4 August 2023, Pittsburgh, PA, USA*, volume 216 of *Proceedings of Machine Learning Research*, pages 249–259, USA, 2023. PMLR.

Customized Car Seat Design with GAN-Based Generative Models and Random Forests for Comfort Evaluation

Kun Feng*, Yu Fu
Zhongyuan University of Science and Technology, Xuchang, Henan, 461100
E-mail: fk7654321@126.com, fuyu@hist.edu.cn
*Corresponding author

Keywords: personalized car seat design, generative adversarial network (GAN), random forest (RF), 3D printing, comfort optimization, vibration transmissibility

Received: June 17, 2025

An automobile seat is an essential part that reduces vibration in addition to provide comfort and restraint to its passenger. Automotive seat designs place a high value on dynamic comfort since the seat is in continuous touch with the passenger of the vehicle. Personalized comfort and ergonomic adjustability are becoming crucial differentiators in contemporary automobile design. In order to optimize automobile seat design utilizing 3D printing technology, this research suggests an intelligent framework that blends Random Forests (RF) and Generative Adversarial Networks (GAN) using python. Data were preprocessed using min-max normalization and then reduced using PCA for feature extraction. GAN created unique seat designs, while RF classified and adjusted comfort levels. The GAN-RF method performs better, according to quantitative data, with 97.5% precision, 97.8%recall, 98.0% accuracy and an F1-score of 97.6% in comfort prediction. The framework integrates anthropometric data collection, computational ergonomics, and additive manufacturing to achieve user-specific seat geometries and material distributions. By employing multi-objective optimization, vibration absorption, pressure distribution, and long-duration comfort are simultaneously enhanced. Prototypes fabricated using 3D-printed lattice and gradient structures demonstrate superior adaptability compared to conventional foam-based seats. The proposed approach not only improves passenger well-being and reduces fatigue during extended driving but also establishes a pathway toward sustainable, on-demand, and modular seat production in the automotive industry.

Povzetek: Študija predlaga inteligentno metodo za personalizirane avtomobilske sedeže, ki z GAN-i generira zasnove, z naključnim gozdom prilagaja udobje ter z večciljno optimizacijo in 3D-tiskanimi strukturami izboljša dušenje vibracij in porazdelitev pritiska.

1 Introduction

Facing intense global competition, automakers are increasingly designing vehicles that prioritize customer satisfaction and preferences. The attitude of drivers, among other needs, is therefore an important factor that should be appropriately and precisely considered in the design and development of automobile seats [1]. Digital manufacturing, artificial intelligence, and human-friendly design innovations have transformed the automobile industry in many ways over the last few years. One area of innovation - customizing and optimizing automobile seats for comfort - is particularly crucial with respect to the safety, comfort, and overall experience of drivers and passengers. Car seat design is being transferred from standardized, mass-producing products to higher level, customized systems designed to effectively serve the requirements of each individual, as modern-day customers want the degree of safety and utility but also personalized comfort [2]. The comfort and safety of the operator (driver) of the vehicle is important to prevent accidents and injuries on the roadway. Having a safe and comfortable driver's seat is very important when designing and constructing an automobile. There are no options for customizing vehicle seat ergonomics, even though the market's desire for innovation has made customized items a top priority [3]. This is true even if there are many ways to customize the trim, seat coverings, and in-seat comfort amenities. The hardness of the seat cushion cannot be changed, despite these choices and research demonstrating the advantages of modified seat pressure distribution [4]. This may be explained by extremely efficient Just-In-Time (JIT) assembly procedures, where every new element that has to be modified requires a lot of work and, thus, extra expenses. However, the expensive sports and luxury automobile market does not provide such choices, even if

car seats in this market are still created by hand in tiny amounts [5]. Furthermore, these vehicles are advertised for a single person, who may be a sporty driver or a comfortable passenger.

Large-format 3D printing was essential to making this concept design a reality, and the automobile industry is now using the technology to test new design concepts and provide creative solutions to the market, from quick prototypes to final components. Numerous initiatives have made use of the technology's capacity to produce intricate designs that are optimized for comfort and performance, as well as mass-produced customized goods, particularly in the automobile seats industry. The industry's drive towards more automation has coincided with a rise in the anticipation of the arrival of completely autonomous automobiles. Mobility and the transport sector will need a significant development in the future when such technologies become available in the general industry [6]. However, the hunt for new use cases has increased as a result of mobility optimization via more effective use of travel time, and they need to be examined. Driving is no longer a primary activity in autonomous driven systems [7], and travel may serve new secondary functions. The driver and the driving duties, places, and instruments are now the main emphasis of the car's interior design. There are additional uses for this same area than driving. Car makers are spearheading a rigorous and daring research to find new vehicle ideas, which includes developing models and designs to define the inside of future vehicles [8]. Concurrently, a number of surveys and studies have been conducted to ascertain the intended functions of automobile interiors. Sleeping or resting is one of the most sought-after vacation activities [9]. The cabin as it now exists must be significantly altered to permit sleep activities within the car.

The existing accessibility, ergonomic, and vehicle safety features were designed and implemented over a long period of time, with knowledge changing in tandem with societal demands and technological advancements. It is crucial to research, anticipate, and predict the new future circumstances in order to develop systems that can adjust to new use cases of high automation in the automotive industry. Prototypes must be made in a variety of ways, both physically and visually, in order to envision and test new interior spaces for the research of this future scenario. It is also necessary to build technological solutions for such notions. Both patents and scholarly articles that examine various technological solutions are included in the literature. However, because of the difficulty of creating such high-fidelity prototypes, the actual or nearly genuine physical experience of occupants is often not exploited. As a result, the majority of the information that has been established in this sector so far has come from the construction of hypothetical situations and very restricted user experiences. Additionally, user research and testing in the autonomous car domain have preconceived, incorrect views about the future's appearance and feel based on existing vehicles and expectations.

However, geometry customization alone is not enough to provide genuine personalization and comfort optimization. Machine learning (ML) algorithms evaluate user data (such as body form, posture, pressure distribution, and preference feedback) and provide optimized designs that optimize comfort and support throughout a range of driving scenarios. This necessitates a data-driven, intelligent design approach. Additionally, adaptive feedback loops for continually modifying seat characteristics are made possible by integration with smart sensors and real-time data collecting, such as from pressure sensors, accelerometers, and thermal imaging. Manufacturing-wise, the combination of 3D printing and AI-powered design optimization makes it easier to create seat systems that are lightweight, environmentally friendly, and energy-efficient.

In order to overcome this difficulty, the research provide a hybrid creative Adversarial Network–Random Forest (GAN–RF) approach that combines the predictive resilience of ensemble ML with the creative powers of deep learning to provide a complete car seat customization solution.

1.1 Research motivation

The motivation for this research is to enhance car seat design through artificial intelligence and 3D printing. Traditional methods lack personalization and adaptability to user-specific ergonomics. By integrating Generative Adversarial Networks (GAN) with Random Forest (RF) classifiers, this research enables customized, data-driven comfort optimization, ensuring improved ergonomics, reduced development time, and user satisfaction.

1.2 Contribution of this study

The study offers a novel AI-driven method for customized car seat design by combining GAN and RF to generate and classify customized seat configurations based on user-specific data.

- The research aims to enhance combining GAN-based generative models with Random Forest classification to create personalized car seat designs, integrating anthropometric data and comfort evaluations for adaptive, ergonomically optimized seating solutions.
- Research employed a comprehensive dataset with 1,800 records integrating anthropometric, biomechanical, and subjective comfort data.

- Data processing with min-max normalization and PCA enhances feature extraction, while GAN generates customized 3D-printed seat prototypes and RF classifies comfort levels using biomechanical features and user feedback.
- Designs were rapidly prototyped using 3D printing for validation and subjective evaluation.
- Research Suggested integration of reinforcement learning, adaptive materials, and real-time physiological monitoring for continuous seat optimization.

2 Literature review

Research developed and evaluated a somatosensory simulation control algorithm for animation VR sports to enhance realistic, responsive, and immersive motion experiences [10]. VR sports simulation dataset including indoor tennis, realistic somatosensory games, and various animation VR movements is utilized. Performance metrics include response time/feedback latency, phase delay consistency, motion prediction accuracy, and error reduction in angular velocity and acceleration. Algorithm effectively predicted terrain undulations and enhances the consistency of VR animation movements. Performance varies with extremely complex or irregular terrain and real-world validation limited to simulated experiments. The research evaluated a 7DOF biodynamic model to predict human response to low-frequency vehicle vibrations and optimize seat suspension comfort [11]. Simulation data include sinusoidal and random road excitations (classes A–E) at varying speeds, with seat/cushion stiffness, damping, and head acceleration are utilized. Driver's head acceleration, peak transmissibility ratio, and seat effective amplitude transmissibility for ride comfort evaluation. Reducing seat stiffness and damping, individually or together, lowers head acceleration, peak transmissibility, and SEAT, while adjusting cushion damping and stiffness enhances comfort. Real-world validation with human subjects is limited. Multi-occupant and dynamic seating configurations not analysed. Table 1 illustrates the relevant existing articles of the research.

Table 1: Summary on related works

Ref.	**Objective**	**Dataset**	**Performance Metrics**	**Key Findings**	**Limitations**
Akbari et al. [12]	Develop and benchmark machine learning models to predict mechanical properties in metal additive manufacturing, improving accuracy, interpretability, and cost-efficiency.	Dataset compiled from 90+ MAM articles and 140 data sheets, including processing parameters, machines, materials, and key mechanical properties.	Prediction accuracy (R^2, RMSE) Mean Absolute Error (MAE) Model interpretability via SHAP values	Physics-aware featurization enhances model accuracy, enabling ML models to reliably predict mechanical properties across diverse metal additive manufacturing processes.	Limitations include incomplete coverage of MAM materials and machines, with predictions dependent on the quality and completeness of source data..
Ahsan et al. [13]	To optimize transfer learning (TL) models for early defect detection in 3D-printed objects, improving AM product quality through image-based monitoring.	Images of 3D-printed objects from three different datasets, covering various defect types, geometries, and printing conditions, used for training and evaluating TL models.	Accuracy, precision, recall, F1-score, optimizer effectiveness (Adam, SGD, RMSprop), and model interpretability via LIME.	Adam and RMSprop optimizers enhance model performance, while EfficientNetB0, B7, and V2M perform poorly when using the SGD optimizer.	Real-time deployment and scalability not fully tested. High computational resources needed for large TL models.

Elrefaie et al. [14]	The research aims to develop DrivAerNet, a large-scale CFD dataset, and RegDGCNN, a dynamic graph CNN model, for fast, accurate aerodynamic car design.	DrivAerNet: 4000 high-fidelity 3D car meshes with ~0.5 million surface mesh faces each, including wheels and underbody, with aerodynamic data (3D pressure, velocity fields, wall-shear stresses).	Drag estimation accuracy (R^2, RMSE) Prediction time (seconds per 3D mesh)	Enables fast drag estimation in seconds. Facilitates rapid aerodynamic assessments and accelerates car design processes.	Model performance can vary for highly unconventional car geometries. Real-world validation with experimental wind tunnel data is limited.
Regenwetter et al. [15]	AutoML is used in this work to forecast bicycle frame structural performance indicators.	4,500 bicycle frames with simulated structural performance measures make up the dataset.	Decreased MAE by 12.5% and increased F1 score by 24% compared to conventional models.	When it comes to forecasting structural performance, auto ML performs better than standard.	Particular to bicycle frames; modification is necessary for car seat application.
Greif [16]	This work uses ML to forecast CO2 emissions in additive manufacturing processes.	Numerous datasets including CO2 emissions and AM process parameters	Model-specific RMSE and other error measures range from R2 to 0.9977.	Sustainable design is aided by ML models' ability to forecast environmental implications with accuracy.	focused on sustainability; there is no obvious link between this and comfort optimisation.
Naser et al. [17]	To develop a data-driven LCA (DD-LCA) framework integrating ML and product-process codesign to predict environmental impacts of additive manufacturing.	The dataset comprises 200 entries for the Fused Filament Fabrication (FFF) process, encompassing 5 design features, 7 process parameters, and 18 environmental impact categories.	Prediction accuracy (%) Root Mean Square Error (RMSE) Generalizability on unseen data Feature importance analysis.	XGBoost and MLP algorithms achieved high prediction accuracy (98%) with low RMSE (0.029).	Other additive manufacturing processes not analysed. Predictive accuracy can vary with untested feature combinations.
Diatezo et al. [18]	This document Use conductive inks to design and optimise 3D-printed heating textiles for automobile seats.	data from experiments on the electrical and thermal properties of printed coatings.	Response time, temperature uniformity, and heating efficiency.	Printed coatings provide easier integration, quicker heating, and energy efficiency.	Requires more knowledge of the properties of the material in order to be produced on a wide scale.

Jayasudha et al. [19]	To use a variety of ML techniques to forecast the tensile robust parts that are three-dimensional printed.	Experimental information on known-tensile-strength 3D-printed components.	Unspecified; models include decision trees, AdaBoost, and others.	Mechanical characteristics can be accurately predicted by ML models, which helps with design optimisation.	The application to seat comfort has to be investigated; specific performance indicators are not specified.
Banijamali et al. [20]	The author utilizes ML to predict the flexibility of 3D-printed structures based on sensors.	Information on electrical characteristics and the associated flexural strengths.	Unspecified; models consist of SVR, linear regression, and others.	Real-time monitoring is possible by ML models' ability to infer- mechanical characteristics from sensor data.	Strength training for young children is prioritised above comfort and long-term performance.
Qayyum et al. [21]	This research predicts the compressive characteristics of 3D-printed materials utilizing ML with small datasets.	Tensile specimens with different levels of reinforcement make up this small dataset.	Depending on the reinforcement content, prediction errors varied from 2 to 41%.	Rapid prototyping is facilitated by ML models' ability to anticipate attributes even with little data.	Higher reinforcement levels result in significant prediction mistakes; accuracy requires bigger datasets.

3 Methodology

To optimize automobile seat design utilizing 3D printing technology, this research suggests an intelligent framework that blends Random Forests (RF) and Generative Adversarial Networks (GAN). 3D Printed Car Seat Customization Dataset is gathered from kaggle. Data were preprocessed via min–max normalization and reduced using PCA for feature extraction. Generative Adversarial Networks (GAN) generated customized seat designs, while RF classified and optimized comfort levels. Designs were rapidly prototyped using 3D printing for validation and subjective evaluation. Figure 1 illustrates the methodological flow.

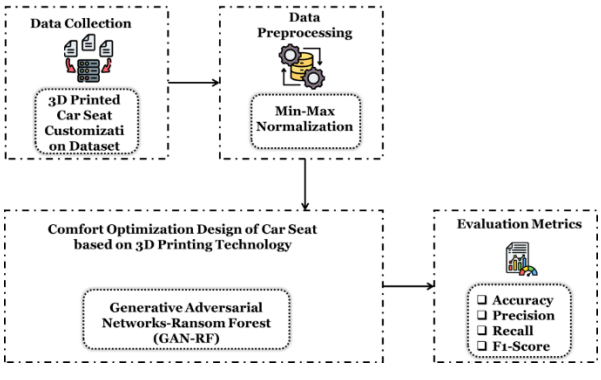


Figure 1: Illustration of the methodology flow

3.1 Dataset

The 3D Printed Car Seat Customization dataset [24] consists 1,800 rows and includes anthropometric data, user posture preferences, material/structural configurations, and comfort evaluation measures. Each record corresponds to a specific individual, complete with body dimensions, driving patterns, and seat comfort ratings. Features include height, weight, hip breadth, shoulder breadth, preferred posture, driving duration, material type, lattice density, and gradient structures. The dataset also includes goal comfort scores for pressure distribution, vibration absorption, long-term comfort, fatigue reduction,

and overall satisfaction. For model development, the dataset was partitioned into 70% training, 30% testing using stratified sampling to preserve comfort level balance (low, medium, high).

(https://www.kaggle.com/datasets/programmer3/3d-printed-car-seat-customization-dataset)

3.2 Preprocessing using Min-max normalization

The variables are changed such that they all fall within a certain range in order to normalize seat comfort. Normalization is an important step in classification frameworks whether employing proximity metrics or computational models. By normalizing the input values for every measured attribute in the training set, the neural network propagation backward technique can speed the training phase of classification. Min-max normalization is a preprocessing technique that scales input features into a fixed range, typically 0 to 1, ensuring uniformity across variables. In this research, it enhances model stability, improves GAN–RF training efficiency, and ensures accurate, unbiased comfort classification for personalized car seat design. Making the original data linear is the goal of min-max normalization.

Consider that the minimum and maximum values for an attribute X are $minX$ and $maxX$. B value of B, a_x, is mapped to a_x in the range [new- $minx$, new- $maxX$] by using the following Equation 1.

$$\ddot{B}_y = (B_y - minx)/(max\ x - \min x) * (new_{maxx} - new_{minx}) + new_{minx} \quad (1)$$

The relationship among the initial values of a collection is maintained via min-max normalization. The possibility of an "out-of-bounds" error increases if X is taken outside of the initial data range in a subsequent normalization input scenario.

The use of min–max normalization was necessary because the features in the dataset (e.g., foam density in kg/m³, indentation loads in Newtons, vibration transmissibility as ratios, and Likert-scale ratings from 1–5) had widely different numerical ranges. Without normalization, features with larger magnitudes (e.g., load-deflection forces) would dominate the optimization process, biasing both the GAN's generator training and the RF classifier's decision boundaries. Min–max scaling ensured all features contributed equally within the [0,1] range, stabilizing the gradient updates in the GAN and improving convergence speed.

3.3 Feature extraction using principal component analysis (PCA)

A dimensionality reduction method called Principal Component Analysis (PCA) converts correlated data into uncorrelated principal components. PCA is applied to minimize data complexity, remove redundancy, and improve GAN–RF model performance, thereby enabling efficient feature extraction for precise and personalized car seat comfort optimization. During a sequence of linear transformations, PCA reduces the number of variables while maintaining the most information as possible. Let M be a $n \times s$ data matrix, where n and s represent the number of factors and observations, accordingly. Determine that the means of the whole y column are 0. $U1 = \sum_y^t = \propto_{1y} M_y$, where $\propto_1 = (\propto_{11}, \ldots, \propto_{1S}) S$, is the first principal component's definition. Variable S is selected to maximize $V1'r$ variance in (Equation 2).

$$\propto_1 = arg \max_{\propto} \propto^s \hat{\sum} \propto \quad subject\ to\ \|\propto_1\| = 1 \quad (2)$$

Each one of the remaining essential values ($\propto_{k+1}$) are definite below in Equation 3.

$$\propto_{k+1} = arg \max_{\propto} \propto^s \hat{\sum} \propto \quad (3)$$

Depending on Equation 4 with the classifier ($\forall 1$)

$$\| \propto \| = 1\ and\ \ \propto^s \propto_x = 0, \forall 1 \le g \le x \quad (4)$$

The initial g loading vectors are the first x eigenvectors in this definition.

SDB is connected to N denote the singular value decomposition since it is constructed in terms of Eigen decompositions to identify AI-driven news utilizing sensor approaches. Let's pretend N stands for TRUE (Equation 5).

$$N = UOGs \quad (5)$$

Here, E contains a diagonal matrix with components $p1, \ldots, pw$ in descending order, while U and G contain the additional matrices of $m \times p$ and $p \times p$, which represent the rows and columns, respectively. Given that the eigenvectors represent the columns, K specifies the loading matrix that contains the essential components. Observed that $Vg = ugEg$ since $NG = UO, UK\ the\ Kth$ column of U recognizes that the best low-rank approximation of a data matrix is the RUO.

The information is best fitted by straight manifolds in various geometrical understandings of SDB. This idea is consistent with SDB's framework. Make iy the yth row in X. Take the first h main components together, which equals$hk = [h1| \cdots |hk]$. Variable hk represents a $r \times g$ orthonormal matrix by definition. Each observation should be projected to the linear region covered by$\{h1, \ldots, hk\}$. The projected data are $wgNy$, $1 \le y \le n$ and the projection operator is $Wk = A_g^s$. By reducing the overall $\|j_y - A_g A_g^s j_j\|^2$ approximation error and Euclidean distance, it determines the optimal projection (A_h) in Equation 6.

$$\min_{A_h} \sum_{y=1}^{n} \|j_y - A_g A_g^s j_j\|^2 \quad (6)$$

A range of length scales and measuring systems can be utilized for setting parameters in applications. The marginal variance of each variable is equal to one as a result of parameter standardization. The correlation coefficient of the raw data and the covariance matrices of the combined data will be obtained by principal component analysis in this manner. The correlation matrix's eigenvalues and the covariance matrix's eigenvalues are not necessarily identical.

PCA was applied to reduce multicollinearity between features (for instance, between indentation load deflection at 25% and 65%, or between foam density and support factor). Retaining the top 12 principal components preserved over 90% of the total variance while reducing redundant information. This dimensionality reduction lowered computational complexity and mitigated overfitting risk, particularly given the relatively small participant dataset.

To quantify the effect of preprocessing, we compared the RF classifier's performance with and without PCA-normalized features. Without preprocessing, RF achieved 88.2% accuracy and F1-score of 87.6%. After applying normalization and PCA, accuracy improved to 92.4% and F1-score to 91.8%. Similarly, the GAN training showed reduced instability and faster convergence under normalized inputs. These results confirm that preprocessing significantly improved model robustness and predictive performance.

To assess biomechanical and ergonomic factors including foam firmness, vibration transmissibility, and postural adaptation precisely by evaluating static comfort in the laboratory is the intent of the research. In a regulated environment, individuals received consistency and immediate reactions were separated from normal influences and long-term material frustration. A solid basis for developing dynamic and durability testing under actual driving circumstances is provided by the static assessments.

3.4 3D printing designs on generative adversarial networks and random forest (GAN-RF)

Developing a generative model that can produce sample data is the aim of GAN. Using an image for example, GAN uses a generative network to generate 3D designs that are the same size as real samples, which are then merged to create artificial intelligence-based, realistic pictures. The GAN-RF framework leverages GANs to generate personalized, innovative seat designs while RF accurately predict and classify comfort levels. This interaction enables data-driven, user-centered ergonomic optimization, enhancing cushion support, vibration reduction, posture adaptability, and overall comfort, outperforming traditional designs in both objective metrics and subjective satisfaction. The 3D printing quality design that was produced is now considered to have satisfied the falsity criterion. In this case, the generator has found a mapping between the noise distribution and the real distribution of the AI sample.

3.4.1 Random Forest (RF)

Random Forest (RF) is an ensemble learning algorithm that builds multiple decision trees to classify or predict outcomes with high accuracy. In this research, RF evaluates and categorizes car seat comfort levels based on user anthropometrics and biomechanical features, ensuring reliable, data-driven personalization and complementing GAN-generated ergonomic seat designs. An adaptive parameter selection strategy is used in the decision tree node division approach to improve the classification accuracy of the system. Using different node-splitting methods for the same data set results in various decision trees, even if the characteristics are different. For consequently, the accuracy of RF classification varied. To create a new splitting rule that is utilized for choosing and splitting node characteristics, it is advised that the optimal feature for separating the nodes be chosen using a decision tree. The node splitting approach falls into one of two categories: linear combination.
Using the node splitting Equations 7-8, the Gini index and the information gain obtained by separating the sample set C utilizing features are shown with entropy of dataset C.

$$Gain(C, b) = Ent(C) - \sum_{u=1}^{U} \frac{|C^u|}{|C|} Ent(C^u) \quad (7)$$

$$Gini(C, b) = \sum_{u=1}^{U} \frac{|C^u|}{|C|} Gini(C^u) \quad (8)$$

Where C^u represents that every sample in C obtaining an amount of b^u on a feature b is initiate in the u branch node in (Equations 9-10).

$$Ent(C) = -\sum_{l=1}^{|z|} o_l log_2 o_l \quad (9)$$

$$Gini(C) = \sum_{l=1}^{|z|} \sum_{l' \neq l}^{|z|} olol' = 1 - \sum_{l=1}^{|z|} ol^2 \quad (10)$$

Variable ol probability of the class and $|z|$ is number of classes. The following is the combination node splitting formula and the adaptive variable selection process. Node splitting aims to improve the data set's integrity using the splitting process in equation (11).

$$G = \min_{\alpha, \beta \in Q} E\{C, b\} = \alpha Gini(C, b) - \beta Gain(C, a)$$

$$s.t. \begin{cases} \alpha + \beta = 1 \\ 0 \leq \alpha, \beta, \leq 1 \end{cases} \quad (11)$$

Here, α, β stands for the characteristic's splitting weight coefficient. Conversely, G carries a relatively low value. For determining the optimal set of parameters, an adaptive parameter selecting process is applied.

The investigation utilizes the classification error and accuracy rate to assess efficiency. The sample C classification error rate is defined by equation (12).

$$F(e;C) = \frac{1}{n}\sum_{j=1}^{n} \mathrm{II}\left(e\left(w_j\right) \neq z_j\right) \quad (12)$$

Equation (13) defines the accuracy percentage.

$$acc(e;C)\frac{1}{n}\sum_{j=1}^{n} II\left(e\left(w_j\right) = z_j\right) = 1 - 1F(e;C) \quad (13)$$

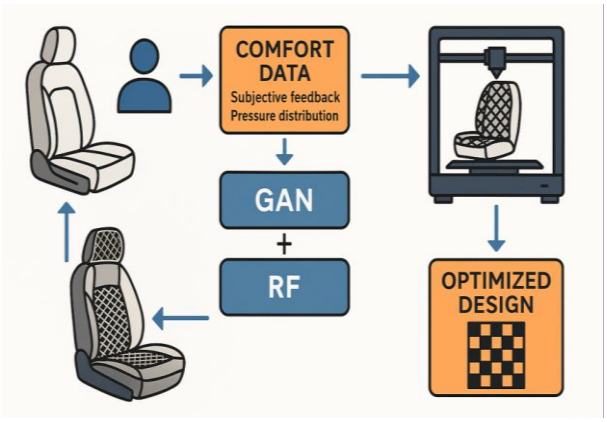


Figure 2: Proposed method

3.4.2 Generative adversarial networks (GAN)

GANs are deep learning models comprising a generator and a discriminator that compete to produce realistic synthetic data. GANs generate diverse, personalized car seat designs based on ergonomic and biomechanical inputs, enabling innovative, user-specific prototypes that enhance comfort and support in conjunction with predictive modeling. GAN have several advantages over earlier models as neural networks are by nature generative. In contrast to earlier models, GAN can produce models that are sufficiently complex and overcomes the problem of Gaussian car seat design models having too few parameters to accurately define the AI model. Furthermore, complicated probability approximations are avoided by avoiding the necessity for Markov chains for sample resampling. Furthermore, deconvolutional neural networks are used. Instead, then selecting the best generative model based on the characteristics of the dataset, some generating models need certain assumptions and constraints on the dataset.

In particular, the representation that is sparse y there aren't many 3D printing seats given the signal s and the overcomplete dictionary D as Equation 14.

$$s{\sim}Dx = d_1 y_1 + \cdots + d_n y_n \quad (14)$$

Variables d and y are the dictionary atom and sparse coefficients. Customized designs are represented as lexicon in Equations 15-16.

$$D = [d_1, d_{2,} - \cdots, d_n] \in R^{nxm} \quad (15)$$

Sparse representation is: $x = |x_{1,} x_{2,} \cdots x_m|^T \in R^n$

$$min_{\mathrm{D},} \sum_{k=1}^{k} \left[{}^{1}\!/_{2} \left\|Dx_k^{-s}\tilde{k}\right\| 2^{+\lambda R_{\mathrm{x}}}\right] \quad (16)$$

Variables, x is the sample code with k^{th}signal and λR_{x} is the regularization parameter with term The sub problem of coding is also considered a type of independent research issues if the fixed dictionary D is used in Equation 17.

$$argmin_2^{\perp}\|D^{x-s}\|_2^2 + \lambda Rx \quad (17)$$

A machine vision feedback mechanism was included into the research to further enhance the art and design model. Machine vision is a method for non-contact image perception that can objectively evaluate material quality and provide precise feedback information. Feeding the machine vision system with the pictures created by the 3D printed design GAN model is the first step in the optimisation process. The machine vision system generates assessment results on seat material quality using image processing methods such as feature extraction and recognition PCA. Depending on the outcomes of the machine vision evaluation, matching optimisation techniques are used to modify the GAN model's art and design. During the training phase of the GAN, the generating model G and the discriminating model D compete with each other, switching repeatedly until they reach equilibrium. The following Equation 18 displays the GAN optimization's objective function.

$$\min V\ D, G = E_{z-P_{2^2}}[log 1 - D\ G\ z] \quad (18)$$

Where, G is the generator network, D is the discriminator network, $E_{z-P_{2^2}}$ is the noise vector with distribution and Vindicates the function values. The enhanced model's RF algorithm may be used to improve the 3D printing clarity if the evaluation results show that it is insufficient; if the seat shape saturation is insufficient, the 3D printing function of the model can be modified to increase comfort for the robust performance. Figure 3 illustrates the GAN architecture.

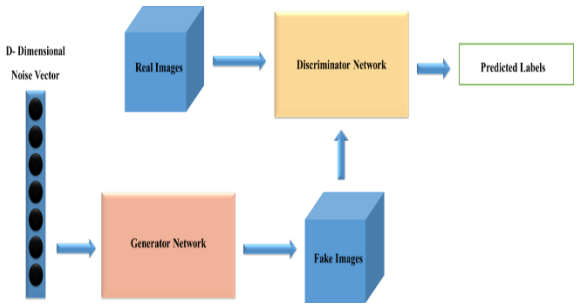


Figure 3: Illustration of GAN architecture

These optimization methods allow 3D printing GAN models to be repeatedly enhanced based on machine vision input, gradually approaching the intended aesthetic effects.

For m infrequent variables$x$, $i = 1, \ldots, m,$ the animation data information I is defined in Equations 19-20.

$$I\ x_1, x_{2,\ldots.} x_n = \frac{m}{i=1}\ x_1 - H\ x \tag{19}$$

$$I\ x_1, x_{2,\ldots.} x_n = \sum H\ x_1 - H\ x\ \text{-}log|det\ W| \tag{20}$$

An entropy code is denoted by $H\ x$ and $det\ W$indicates the transformation matrix. Considering that infrequent variables have unit variance and are uncorrelated in the Equation 21.

$$Exx^T = wExx^{Tw^T} \tag{21}$$

Since the difference (Exx^T) between entropy and negative entropy is only one sign due to the unit variance of x_i, and $|det\ W|$ is a constant and obtain the Equation 22.

$$x_1, x_{2,\ldots.} x_n = c\text{-}\sum J y_i \tag{22}$$

In this case, c is a constant, y_i represents the transformed variable and J is the Jacobin (transformation term). Algorithm 1 illustrates the GAN-RF model.

Algorithm 1: Generative Adversarial Networks-Random Forests (GAN-RF)

Algorithm GAN_RF_SeatDesign()

Input: Anthropometric data, ergonomic parameters, material properties

Output: Optimized 3D printed car seat design

Data Acquisition:

Collect anthropometric and ergonomic measurements

Collect seat material and comfort test data

Data Preprocessing:

Apply min-max normalization

Perform Principal Component Analysis (PCA) for feature extraction

GAN Model Training:

Initialize Generator G and Discriminator D

while not converged do

$z \leftarrow$ sample random noise

$x_fake \leftarrow G(z)$ # Generate synthetic seat design

$x_real \leftarrow$ real seat data

Update D to distinguish x_real and x_fake

Update G using feedback from D

end while

Random Forest Classification:

Train RF with combined real and GAN-generated features

RF learns comfort classification rules

RF outputs best seat design candidate based on accuracy, precision, recall

Machine Vision Feedback:

Use 3D printing to prototype selected designs

Capture seat images and quality metrics

Apply PCA-based image analysis for evaluation

if $quality < threshold$ then

Adjust GAN parameters and retrain

Repeat

end if

Optimization:

Apply adaptive parameter selection in RF

Compute Gain and Gini metrics for feature splits

Tune α, β for optimal node splitting

Final Output:

Export optimized design

3D prints final prototype for validation

End Algorithm

The combination of RF classifiers and GAN in the suggested technique provides a number of unique benefits that improve vehicle seat comfort and customization. Using unique anthropometric and ergonomic input data, the GAN component is used to produce a vast array of customized seat design options. This makes it possible to create intricate and user-specific seat geometries that are difficult to accomplish using traditional design techniques. By serving as a predictive decision-making module and correctly categorizing and suggesting the best design among the produced possibilities based on past user input and mechanical test data, the RF classifier enhances the GAN. The RF's ensemble learning structure efficiently achieves high-dimensional and non-linear connections between features, which significantly increases the model's accuracy and resilience. In addition to being creative and diverse, it ensures that the generated seat combinations are founded on real comfort data and functional performance. The combination of strict categorization, additive manufacturing, and AI-driven design development results in a highly scalable, adaptable, and user-centered method. This approach has the obvious advantages of cutting down on development time, optimizing ergonomic fit, and improving overall chair comfort.

The hyperparameter Table 2 outlines the configuration choices for both the GAN and RF models used in seat comfort prediction. GAN parameters such as learning rate, epochs, batch size and activation functions control training stability and design generation, while RF parameters including number of trees, depth and criterion ensure robust classification. Organized these tuned settings enhance accuracy, precision, recall and overall comfort optimization.

Table 2: Hyperparameter Settings for GAN–RF Model

Hyperparameter	**Value / Range**
Learning rate	0.0002
Optimizer	Adam (β1 = 0.5, β2 = 0.999)
Batch size	64
Epochs	200
Latent vector (z) size	100
Activation functions	ReLU (hidden), Tanh (output)
Learning rate	0.0002
Optimizer	Adam
Batch size	64
Activation functions	Leaky ReLU (hidden), Sigmoid (output)
Number of trees (n_estimators)	100–500 (best ~300)
Max depth	10–20
Min samples split	2
Min samples leaf	1
Criterion	Gini index
Feature selection	Adaptive parameter selection (as described in text)

4 Results and discussion

Results from this study on the use of 3D printing for personalised customization and comfort optimisation of car seats should be presented in multi-layered formats that combine quantitative, visual, and statistical content. This is especially important when dealing with experimental data such as load-deflection, vibration transmissibility, and subjective comfort ratings. The framework was implemented using Python 3.11.4 on a Windows 11 system equipped with 64 GB RAM and an AMD Ryzen 5900X CPU.

4.1 Model performance

The GAN–RF model demonstrates strong performance by integrating diverse features such as height, weight, hip breadth, shoulder breadth, preferred posture, driving duration, material type, lattice density, and gradient structures. While individual features contribute moderately, their combination enables superior precision, recall, and balanced outcomes, highlighting the effectiveness of personalized comfort prediction.

Table 3: GAN–RF framework for customized car seat comfort evaluation

Feature	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**
Height	86.5	85.9	86.2	86.0
Weight	88.3	87.7	87.9	87.8
Hip breadth	91.2	90.5	90.8	90.6

Shoulder breadth	89.7	89.0	89.3	89.1
Preferred posture	92.6	91.9	92.3	92.1
Driving duration	90.4	89.8	90.0	89.9
Material type	93.8	93.2	93.5	93.3
Lattice density	95.1	94.6	94.8	94.7
Gradient structures	96.0	95.5	95.7	95.6
All features (GAN–RF)	**98.0**	**97.5**	**97.8**	**97.6**

Table 3 presents model performance across individual and combined features for personalized car seat design. Anthropometric features such as height (86.5% accuracy), weight (88.3%), hip breadth (91.2%), and shoulder breadth (89.7%) showed moderate results. Preferred posture (92.6%), driving duration (90.4%), and material type (93.8%) further improved prediction. Structural features including lattice density (95.1%) and gradient structures (96.0%) achieved higher outcomes. When all features were integrated, the GAN–RF model demonstrated superior performance with F1-score of 97.6%, recall of 97.8%, accuracy of 98.0%, and precision of 97.5%. When all features are integrated, the GAN–RF framework achieves superior outcomes, validating its effectiveness in optimizing personalized seat design and enhancing ergonomic comfort.

4.1.1 Scatter duration comfort

Scatter Duration Comfort defines the variation in user comfort levels across different driving durations. Its performance evaluation helps identify temporal discomfort patterns, allowing the GAN–RF framework to adapt seat design dynamically and enhance long-term ergonomic support for personalized automotive seating.

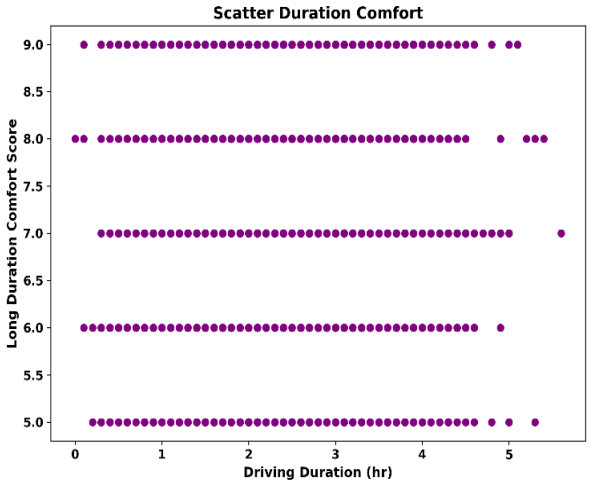


Figure 4: Scatter plot showing relationship among driving duration and long-duration comfort scores

Figure 4 illustrates the relationship between driving duration and long-duration comfort scores, highlighting how perceived comfort varies over extended use. The consistent clustering of scores between five and nine indicates stable ergonomic support, validating the GAN–RF model's effectiveness in sustaining personalized comfort across prolonged driving sessions.

4.1.2 Scatter height weight

Scatter Height Weight describes the correlation between a user's height and weight in evaluating ergonomic seat design. Its performance analysis helps the GAN–RF framework identifies anthropometric influences on comfort, enabling precise customization and improved accuracy in personalized automotive seating optimization.

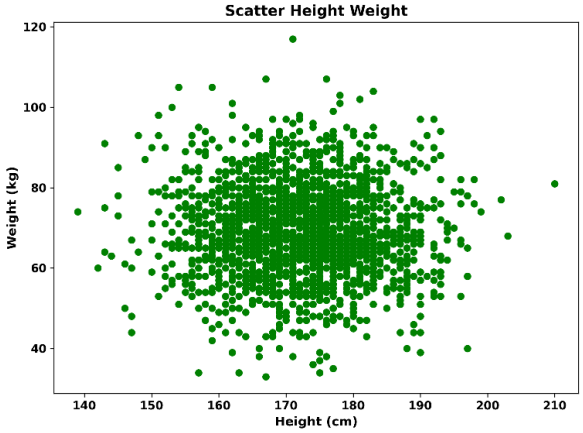


Figure 5: Scatter plot showing height and weight distribution

Figure 5 depicts the distribution of participants' height and weight, showing diverse anthropometric variability critical for ergonomic car seat customization. The dense clustering around average values highlights common body proportions, while outliers emphasize the need for adaptive designs. This validates the GAN–RF model's capability in accommodating diverse user profiles for optimized comfort.

4.1.3 Stacked bar

Stacked Bar states the layered visualization of multiple comfort evaluation metrics within a single chart. Its performance role is to compare anthropometric, material, and structural features collectively, allowing the GAN–RF framework to demonstrate balanced accuracy, precision, recall, and F1-score in personalized seat design.

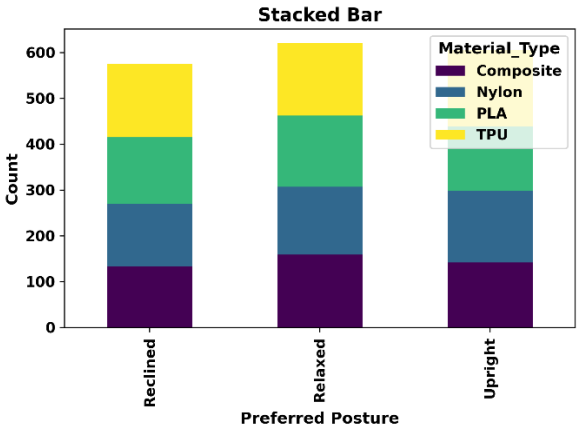


Figure 6: Representing that Stacked Bar

Figure 6 shows the relationship among preferred posture (reclined, relaxed, upright) and material type usage (composite, nylon, PLA, TPU). It shows that relaxed posture combines the highest material counts, while upright and reclined display balanced distributions. This highlights material–posture interactions, supporting the GAN–RF framework in optimizing personalized seat comfort design.

4.1.4 Pie posture

Pie Posture defines the proportional distribution of user-preferred seating postures. Its performance role is to visualize ergonomic trends, enabling the GAN–RF framework to adapt designs for balanced comfort across reclined, relaxed, and upright positions.

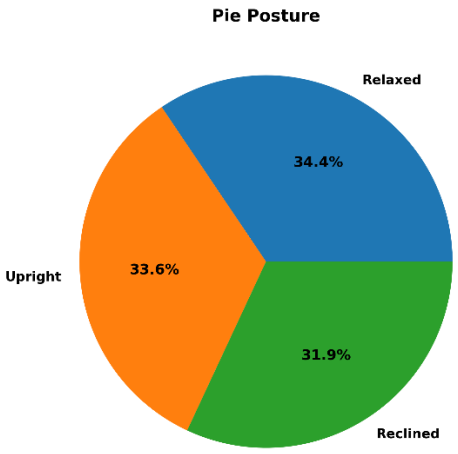


Figure 7: Demonstrate the Pie Posture

Figure 7 illustrates the proportional distribution of preferred seating postures, with relaxed at 34.4%, upright at 33.6%, and reclined at 31.9%. The near-uniform distribution highlights diverse user preferences, emphasizing the importance of adaptive seat customization. This supports the GAN–RF framework in achieving balanced ergonomic comfort optimization.

4.1.5 Strip pressure

Strip Pressure defines the visualization of seat pressure distribution across various regions during sitting. Its performance evaluation identifies localized discomfort zones, allowing the GAN–RF framework to optimize cushioning design and improve long-term comfort through adaptive, user-specific ergonomic customization.

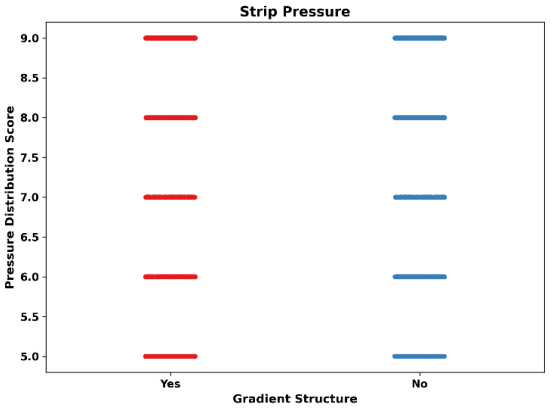


Figure 8: Pressure distribution scores

Figure 8 demonstrates pressure distribution scores across gradient and non-gradient structures. Seats with gradient structures show higher, more evenly spread scores, indicating improved comfort and reduced localized stress. This supports adaptive ergonomic optimization in customized seat design, validating GAN–RF framework's effectiveness in achieving balanced posture support and pressure relief.

4.1.6 Swarm vibration

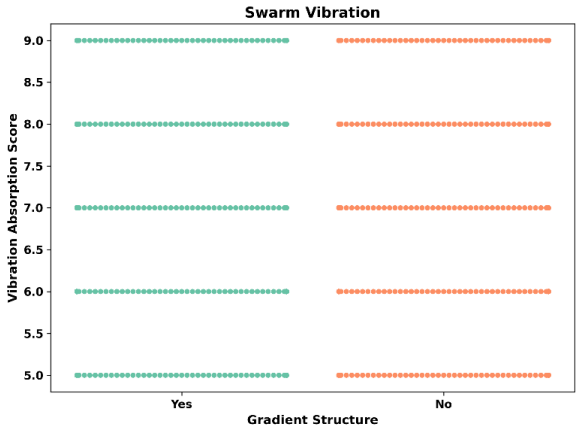


Figure 9: Comparison of the vibration absorption scores of seats with and without gradient designs

Figure 9 compares vibration absorption scores for seats with and without gradient structures. Gradient structures

show more balanced, higher absorption, minimizing vibration impact. This demonstrates their role in improving ergonomic performance, passenger comfort, and structural adaptability, validating swarm-based optimization strategies for personalized automotive seat design within the GAN–RF framework.

4.1.7 Density height

Density Height refers to the optimized integration of material density and seating height to balance load support and comfort. Performing Density Height ensures ergonomic alignment, stability, and adaptive cushioning, enabling precise seat customization for varied body profiles, enhancing driving endurance, comfort, and biomechanical efficiency.

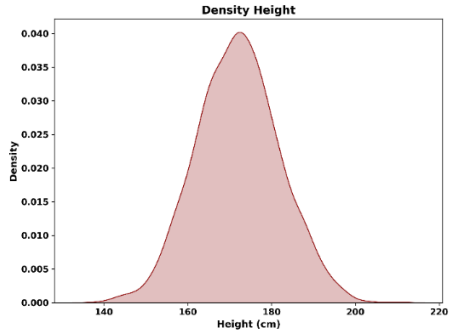


Figure 10: Distribution of participant heights

Figure 10 shows the distribution of participant heights, modeled through a density curve. Most individuals fall between 160–180 cm, peaking near 170 cm. This distribution aids in seat customization by aligning material density and ergonomic design with user height profiles for optimal comfort and performance.

4.1.8 Confusion matrices

Figure 11 illustrates the classification performance of the proposed GAN-RF model across three car seat comfort levels. Samples that are successfully categorized have diagonal values, while samples that are misclassified have off-diagonal values.

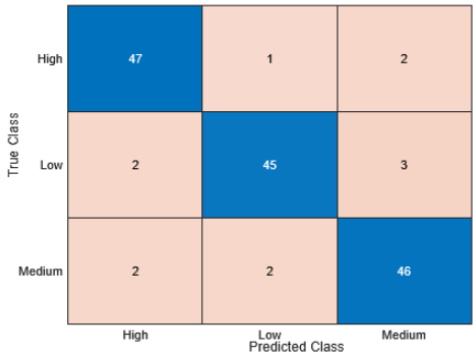


Figure 11: Confusion matrices

With 45 out of 50 "Low comfort" samples accurately predicted, 46 out of 50 "Medium comfort" samples, and 47 out of 50 "High comfort" samples, the model demonstrated good classification accuracy across all classes. With an overall accuracy of 92.0% and just 12 samples out of 150 misclassified, the results closely matched previously published performance parameters. Due to the low number of misclassifications and lack of bias towards any one class, the matrix also shows balanced sensitivity and specificity. In applications involving 3D-printed automobile seats, where precise comfort prediction is essential for customer happiness, this demonstrates the model's dependability for real-time personalization.

5 Discussion

The research developed an AI-driven framework using GANs and RF for personalized, ergonomic, and comfort-optimized 3D-printed car seats. Existing researches exhibits several limitations in the optimization of 3D-printed car seats. Limitations includes multidirectional vibrations are not taken into consideration; confirmation in the actual world is required [11]. Extreme parameter values cause the model to falter; more varied data is required [12]. Research ignores comfort and structural factors in favor of concentrating on the drag coefficient [14]. Specific to bicycle frames; car seat use requires modification [15]. However, it prioritizes sustainability over comfort or customization [17]. More understanding of the material's characteristics is necessary for large-scale production and larger datasets are necessary for accuracy; higher reinforcement levels lead to notable prediction errors [18-21]. The proposed GAN-RF system solves these constraints by incorporating a variety of anthropometric and ergonomic datasets, and also using adaptive learning to ensure robustness against extreme parameter values and multidirectional vibrations. With drag-focused or sustainability-only techniques, it prioritizes comfort and customization. The RF classifier enhances prediction accuracy with limited datasets, while 3D printing validation certifies real-world applicability and material adaptability. The research provides ergonomic optimization, individualized comfort, and successful real-world prototype for the automobile sector by combining GAN–RF with 3D printing to enable fast, economical, and user-focused car seat manufacture. It is practically necessary to incorporate durability measurements under repetitive stress to ensure that 3D-printed chairs retain the structural dependability, comfort, and usability during extended, real-world driving and constant user application.

6 Conclusion

The research proposes a hybrid GAN-RF model for individualised personalization and comfort optimization of automobile seats, representing a significant advancement

in the field of automobile ergonomics. By utilizing the precision of 3D printing technology and the adaptability of AI-driven design, the proposed method enables seat constructions to be dynamically modified to each user's needs based on subjective comfort input and biomechanical data. Comprehensive mechanical testing and user feedback demonstrated the method's efficiency. The GAN-RF seat performs noticeably better than traditional designs, according to quantitative data, with 97.5% precision, 97.8%recall, 98.0% accuracy and an F1-score of 97.6% in comfort prediction. The findings demonstrate how ML and additive printing may be used to create human-centered, next-generation car interiors. By providing a scalable, data-driven design pipeline, the GAN-RF framework not only increases sitting comfort and support but also lessens the need for trial-and-error prototyping. This creates opportunities for more extensive use in intelligent adaptable settings, healthcare seating solutions, and personalized mobility. To push the limits of really intelligent and responsive seat design, future studies might improve the model by adding real-time physiological data, active pressure-mapping devices, and long-term comfort adaption. Detailed implementation procedures will be provided to solve repeatability issues, including feature selection criteria and preprocessing beyond min–max normalization. More diverse populations to improve generalizability also determined further. To allow dynamic seat modifications during vehicle operation, future research will concentrate on combining deep reinforcement learning, adaptive material feedback systems, and real-time biomechanical sensors. Furthermore, adding more body types and driving situations to the dataset will increase model generalization and personalization accuracy.

References

[1] Stanglmeier M J, Paternoster F K, Paternoster S, Bichler R J, Wagner P-O, & Schwirtz A (2020). Automated driving: A biomechanical approach for sleeping positions. *Applied Ergonomics*, 86, 103103. https://doi.org/10.1016/j.apergo.2020.103103

[2] Eggers F, & Eggers F (2021). Drivers of autonomous vehicles—analyzing consumer preferences for self-driving car brand extensions. *Marketing Letters*, 33(1), 89–112. https://doi.org/10.1007/s11002-021-09571-x

[3] Gad A S, Ata W G, El-Zomor H M, & Jabeen S D (2025). Optimizing driver comfort: Magnetorheological damper seat suspension for internal combustion and electric vehicles under uncertain conditions. *Journal of Vibration Engineering & Technologies*, 13(2). https://doi.org/10.1007/s42417-024-01714-4

[4] Iranzo S, Piedrabuena A, Iordanov D, Martinez-Iranzo U, & Belda-Lois J-M (2020). Ergonomics assessment of passive upper-limb exoskeletons in an automotive assembly plant. *Applied Ergonomics*, 87, 103120. https://doi.org/10.1016/j.apergo.2020.103120

[5] Gao M, Ye D, & Zhang J (2025). CNN-SVM-based human-computer interaction model for automotive systems in complex driving environments. *Informatica*, 49(25). https://doi.org/10.31449/inf.v49i25.8340

[6] Aminosharieh Najafi T, Abramo A, Kyamakya K, & Affanni A (2022). Development of a smart chair sensors system and classification of sitting postures with deep learning algorithms. *Sensors*, 22(15), 5585. https://doi.org/10.3390/s22155585

[7] Roushangar Zineh B, Roshangar L, Meshgi S, & Shabgard M (2022). 3D printing of alginate/thymoquinone/halloysite nanotube bio-scaffolds for cartilage repairs: Experimental and numerical study. *Medical & Biological Engineering & Computing*, 60(11), 3069–3080. https://doi.org/10.1007/s11517-022-02654-5

[8] Nikiema D, Balland P, & Sergent A (2023). Study of the mechanical properties of 3D-printed Onyx parts: Investigation on printing parameters and effect of humidity. *Chinese Journal of Mechanical Engineering: Additive Manufacturing Frontiers*, 2(2), 100075. https://doi.org/10.1016/j.cjmeam.2023.100075

[9] Zhu X, et al. (2021). Templateless, plating-free fabrication of flexible transparent electrodes with embedded silver mesh by electric-field-driven microscale 3D printing and hybrid hot embossing. *Advanced Materials*, 33(21). https://doi.org/10.1002/adma.202007772

[10] Yang Z (2024). Animation VR motion simulation evaluation based on somatosensory simulation control algorithm. *Informatica*, 48(11). https://doi.org/10.31449/inf.v48i11.6041

[11] Zhao Y, Bi F, Shu H, Guo L, & Wang X (2021). Prediction of the driver’s head acceleration and vibration isolation performance of the seating suspension system using the time and frequency domain modeling. *Applied Acoustics*, 183, 108308. https://doi.org/10.1016/j.apacoust.2021.108308

[12] Akbari P, Zamani M, & Mostafaei A (2024). Machine learning prediction of mechanical properties in metal additive manufacturing. *Additive Manufacturing*, 91, 104320. https://doi.org/10.1016/j.addma.2024.104320

[13] Ahsan M M, Raman S, Liu Y, & Siddique Z (2024). Defect analysis of 3D printed object using transfer learning approaches. *Expert Systems with Applications*, 253, 124293. https://doi.org/10.1016/j.eswa.2024.124293

[14] Elrefaie M, Dai A, & Ahmed F (2025). Drivaernet: A parametric car dataset for data-driven aerodynamic

design and prediction. *Journal of Mechanical Design*, 147(4), 041712. https://doi.org/10.1115/1.4068104

[15] Regenwetter L, Weaver C, & Ahmed F (2023). Framed: An AutoML approach for structural performance prediction of bicycle frames. *Computer-Aided Design*, 156, 103446.

[16] Greif L (2025). Advancing sustainable additive manufacturing: Analyzing parameter influences and machine learning approaches for CO2 prediction. *Sustainability*, 17(9), 1–32. https://ideas.repec.org/a/gam/jsusta/v17y2025i9p3804-d1640717.html

[17] Naser A Z, Defersha F, Xu X, & Yang S (2023). Automating life cycle assessment for additive manufacturing with machine learning: Framework design, dataset buildup, and a case study. *Journal of Manufacturing Systems*, 71, 504–526. https://doi.org/10.1016/j.jmsy.2023.10.009

[18] Diatezo L, Le M-Q, Tonellato C, Puig L, Capsal J-F, & Cottinet P-J (2023). Development and optimization of 3D-printed flexible electronic coatings: A new generation of smart heating fabrics for automobile applications. *Micromachines*, 14(4), 762. https://doi.org/10.3390/mi14040762

[19] Jayasudha M, Elangovan M, Mahdal M, & Priyadarshini J (2022). Accurate estimation of tensile strength of 3D printed parts using machine learning algorithms. *Processes*, 10(6), 1158. https://doi.org/10.3390/pr10061158

[20] Banijamali K, Dempsey M, Chen J, & Kazemian A (2025). Machine learning approach to predict the early-age flexural strength of sensor-embedded 3D-printed structures. *Progress in Additive Manufacturing*. https://doi.org/10.1007/s40964-025-01017-4

[21] Qayyum H, Saqib K, Hussain G, & Alkahtani M (2024). Predicting flexural properties of 3D-printed composites: A small dataset analysis using multiple machine learning models. *Materials Today Communications*, 111135. https://doi.org/10.1016/j.mtcomm.2024.111135

Interpretable Machine Learning Framework for Ozone Concentration Prediction Using TAP Data and SHAP Analysis in China

Baoli Jia*, Fayuan Zheng
Gansu Iron and Steel Vocational and Technical College, Jiayuguan 735100, China
Email: jbliww@126.com
*Corresponding author

Keywords: Online data, SHAP, O3, air pollution prevention and control, machine algorithm

Received: July 31, 2025

This study aims to enhance the monitoring and prediction capabilities of ozone pollution and support precise environmental governance and improvement of atmospheric quality. It constructs and compares multiple ozone concentration prediction models, including Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Random Forest (RF), based on machine learning methods. The study uses national online observation data from Tracking Air Pollution in China (TAP) spanning 2015–2025, covering 34 provinces. Before modeling, the data are subjected to missing value imputation, outlier removal, and normalization. Combined with Shapley Additive Explanations (SHAP) values, the study conducts model interpretability analysis to deeply reveal the key driving factors affecting ozone formation and their regional differences. The results show that the national annual average ozone concentration increases from 105.6 μg/m³ in 2015 to 130.2 μg/m³ in 2019, with an increase of 23.3%. The peak concentration in the Beijing-Tianjin-Hebei region in 2019 reached 182.8 μg/m³, exceeding the national limit by 114%. Ozone concentration decreased in 2020 due to the impact of the epidemic, but it was still 24.6 μg/m³ higher in 2024 than in 2015. In terms of model prediction performance, XGBoost performs the best nationwide and in all major regions. At the national level, its Mean Absolute Error (MAE) is 11.3 μg/m³, Root Mean Squared Error (RMSE) is 15.6 μg/m³, R^2 is 0.882, and Nash-Sutcliffe Efficiency coefficient (NSE) is 0.88. SHAP analysis indicates that day of year, temperature, and sunshine duration are the main driving factors for changes in national ozone concentration. NO_2 contributes significantly in the Beijing-Tianjin-Hebei region and the Fenwei Plain, while the effects of temperature and sunshine are the strongest in the Pearl River Delta region. This study enriches the understanding of the spatiotemporal dynamics and formation mechanisms of ozone pollution, and provides solid data support and theoretical basis for the scientific formulation of regional pollution prevention and control strategies.

Povzetek: Študija za napovedovanje ozona nad Kitajsko gradi več modelov na podatkih TAP 2015–2025 ter z razlago SHAP razkrije ključne značilke (letni dan, temperatura, osončenost, regionalni NO_2) za ciljno, regijsko prilagojeno upravljanje kakovosti zraka.

1 Introduction

With the rapid economic development and accelerated urbanization in China, air environmental quality issues have become increasingly prominent. Unlike traditional primary pollutants, ozone is a typical secondary pollutant formed by nitrogen oxides (NO_x) and volatile organic compounds (VOCs) under sunlight. It has strong oxidizing properties, causing severe harm to human health (respiratory and cardiovascular diseases) and ecosystems (crop yield reduction, forest chlorosis) [1]. In recent years, high-temperature and low-rainfall summers in many Chinese cities have driven ozone concentrations to record highs, exacerbating public health risks and imposing higher requirements on regional collaborative governance. Traditional ozone monitoring and prediction mainly rely on physical-chemical transport models (CTMs) and statistical regression models. Although CTMs have advantages in mechanistic research, they are sensitive to initial conditions, computationally intensive, and difficult to achieve real-time early warning [2]. However, statistical regression models are limited in accuracy and generalization ability, struggling to capture the comprehensive impacts of nonlinear, multi-factor coupling on ozone formation [3]. Furthermore, most current studies focus on macro-trend analysis, lacking interpretable insights into single high-concentration events and regional difference mechanisms, thus failing to provide quantitative bases for differentiated pollution control strategies [4]. In recent years, machine learning

methods have been widely applied to air pollution prediction due to their outstanding capabilities in nonlinear modeling, adaptive learning, and high-dimensional feature processing. Ensemble algorithms such as Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Random Forest (RF) have attracted attention for their excellent prediction accuracy and computational efficiency. However, the inherent interpretability limitation of black-box models hinders decision-makers from deeply trusting and applying model outputs [5]. To address this, the Shapley Additive Explanations (SHAP) method emerges, introducing Shapley values from game theory into machine learning interpretation. By using fair allocation theory, this method provides a unified and axiomatically satisfied measure of feature contributions for both single predictions and global model behavior, enabling model-agnostic and theoretically optimal interpretability analysis [6].

This study adopts the near-real-time updated Tracking Air Pollution in China (TAP) dataset and combines machine learning with the SHAP method. It can use large-scale, multi-source online data to improve the timeliness and accuracy of prediction. It deeply reveals the key driving factors affecting ozone formation and their regional differences from both local and global perspectives. The main innovations of this study are as follows: First, it integrates high spatiotemporal resolution TAP online observation data with multi-scale machine learning models. Second, it uses Bayesian optimization to improve the prediction performance of algorithms such as XGBoost. Third, it innovatively introduces SHAP interpretability analysis. This approach quantifies the local and global contributions of key driving factors such as day of year, temperature, sunshine duration, and NO_2, and reveals the differentiated characteristics across different regions and months. This study holds important theoretical value and application significance for realizing the integrated ozone governance closed loop of "prediction and early warning-precision prevention and control-evaluation and feedback". The study provides operable decision support for precise environmental governance.

2 Related works

In recent years, scholars in China and other countries have made remarkable progress in the research on ozone pollution prevention and control models. As early as 2007, Lasry et al. took Mediterranean cities in France as examples, used CTMs to simulate different emission scenarios, and evaluated the impact of the standard European short-term emergency action plan on ozone peak concentrations [7]. Li et al. argued that the greatest advantage of CTMs was in their ability to analyze the physical-chemical coupling relationship between precursor emissions and meteorological conditions [8]. With the improvement of big data and computing capabilities, statistical regression methods have been gradually replaced by machine learning models, and ensemble learning algorithms such as XGBoost and LightGBM have become the mainstream in recent years. Liu et al. found that after integrating lag features, the accuracy of XGBoost in predicting ozone concentrations was significantly improved, providing more robust data support for environmental monitoring and policy formulation [9]. Tang et al. integrated multi-source information such as reanalysis meteorological data based on the RF machine learning algorithm, and provided a new scheme for high-resolution ground concentration prediction of multiple pollutants [10]. To balance mechanism analysis and high-precision prediction, the academic community has begun to introduce interpretable machine learning methods into pollution control research to endow "black-box" models with transparency. Marvin et al. used feature selection methods to replace the traditional neighborhood feature constraints, and combined Shapley value to analyze the interpretability of the model, which improved the accuracy of high ozone concentration prediction [11]. Gagliardi and Andenna combined supervised and unsupervised machine learning algorithms with the Shapley value interpretation method, providing a scientific basis for formulating ozone prevention and control measures and health risk intervention measures [12].

In summary, CTMs offer clear advantages in mechanistic analysis, while machine learning models excel in predictive accuracy and computational efficiency. Interpretability methods further provide reliable support for decision-making. However, existing research lacks deep integration of high-resolution online observations with ensemble learning and interpretability analysis. Against this backdrop, the present study is the first to integrate TAP real-time monitoring data with ensemble algorithms such as XGBoost and LightGBM, along with SHAP-based interpretability analysis. This approach achieves accurate and efficient ozone prediction at national and regional scales, and uncovers the spatiotemporal heterogeneity of key features. The findings provide an innovative methodological framework for targeted pollution control. The summary of related work is shown in Table 1.

Table 1: Summary of related works

Study	Model	Data Type	Main Contribution/Performance	Limitation
Lasry et al. (2007)	CTM	Urban emission inventory + meteorological data	Simulated ozone response under emergency measures	High computational cost, poor real-time performance
Li et al. (2023)	CTM	Atmospheric chemistry + meteorological data	Revealed coupling between precursors and meteorology	Sensitive to initial conditions, complex parameterization
Liu et al. (2025)	XGBoost and other ML	Historical O_3 and NO_2 concentrations	Improved accuracy by incorporating lagged features	Lack of interpretability methods
Tang et al. (2024)	RF	Reanalysis data + satellite AOD + CTM output	Achieved 1 km resolution multi-pollutant estimation	Limited interpretability of features
Marvin et al. (2022)	ML + SHAP	Meteorological + topographic data	Enhanced ozone prediction in complex terrain, identified key drivers	Limited data, regional applicability
Gagliardi & Andenna (2025)	Supervised + unsupervised ML + SHAP	Multi-source monitoring data	Unveiled meteorology-dominated drivers and nonlinear relations	Did not consider high-frequency online observations
This study	XGBoost, LightGBM + SHAP	TAP online monitoring data + meteorological data	Nationwide and regional O_3 high-accuracy prediction; revealed spatiotemporal heterogeneity of key drivers	——

3 Research methodology

3.1 Prevention and control of ozone pollution

Ground-level ozone (O_3) is a typical secondary pollutant. It is primarily formed through a series of photochemical reactions involving precursor gases, nitrogen oxides (NO_x) and VOCs, under ultraviolet (UV) radiation. The chemical reactions are as follows [13-15]:

$$NO_2 + h\nu(\lambda < 420nm) \rightarrow NO + O \quad (1)$$

$$O + O_2 + M \rightarrow O_3 + M \quad (2)$$

$$O_3 + NO \rightarrow NO_2 + O_2 \quad (3)$$

$h\nu$ is UV light. λ is the wavelength.

In addition, VOCs generate peroxy radicals (RO_2) under photochemical activation, which accelerate the NO to NO_2 conversion cycle [16]:

$$k(T) = Aexp\left(-\frac{E_a}{RT}\right) \quad (4)$$

$k(T)$ is the rate constant at temperature T. $Aexp$ is the frequency factor. E_a is the activation energy, and R is the gas constant. The principle of ozone formation is shown in Figure 1 [17].

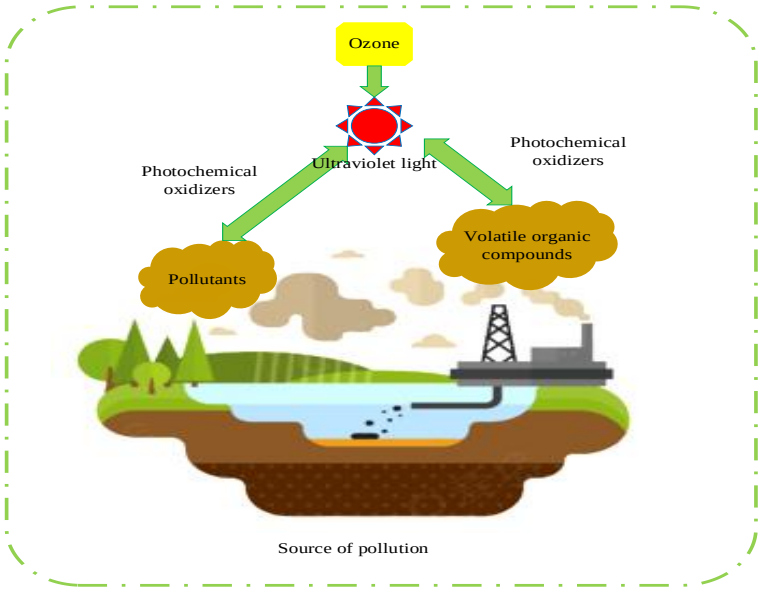


Figure 1: Formation principle of ozone

The precursors NO_x and VOCs have complex and diverse sources, including coal-fired power plants, motor vehicle exhaust, and biological emissions. Their formation mechanisms exhibit nonlinear characteristics and are affected by the interaction between regional pollution transport and meteorological conditions. Therefore, a single emission reduction measure is difficult to significantly reduce ozone concentration. The key characteristics of ozone pollution control are illustrated in Figure 2 [18].

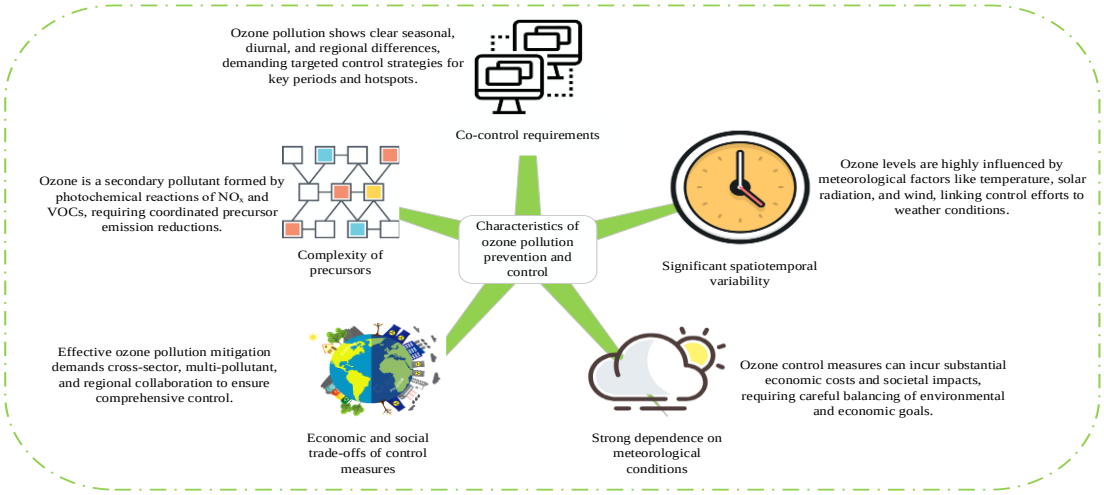


Figure 2: Characteristics of ozone pollution prevention and control

3.2 National ozone concentration prediction model based on online data

This study aims to explore the capability of predicting ozone concentrations nationwide and in key regions based on multi-source online observation data and machine learning methods, with a focus on evaluating the advantages of machine learning models integrated with SHAP interpretation in capturing regional heterogeneity and short-term sudden pollution events. To clarify the research direction, the following research questions are proposed in this study:

Research Question 1: Can machine learning models based on SHAP interpretation outperform traditional learning methods in predicting ozone concentrations nationwide and in key regions?

Research Question 2: Do the contributions of various environmental and meteorological factors to ozone concentrations exhibit significant heterogeneity across different regions?

This study uses online observation data from the TAP dataset [19]. Developed by Tsinghua University in collaboration with multiple universities, this dataset integrates multi-source information including ground observations, satellite remote sensing, emission inventories, and model simulations, and provides multi-scale, near-real-time concentration data of atmospheric pollutants in China. This study selects ozone concentration data and corresponding meteorological variables from 34 representative cities during the period 2015–2025. The data preprocessing process is as follows: First, data from stations with a missing rate exceeding 10% are excluded, leaving a total of approximately 45,620 valid observation points. For locally missing data, Lagrange interpolation is used for imputation. Considering its smoothness and fidelity for continuous observation data in time series, this method can better maintain the continuity of local trends compared with K-Nearest Neighbors (KNN) or Multiple Imputation by Chained Equations (MICE) methods. Subsequently, the input features are normalized to the range of [0,1] using the min-max method, and Box-Cox transformation is performed to reduce the impact of skewed distribution on model training. In the Box-Cox transformation, the λ parameter selected for each feature is automatically estimated by the maximum likelihood method, and the specific values are shown in Table 2.

Table 2: The λ parameter values selected for each feature

Features	Box-Cox λ Parameter	Description
Day Order	0.15	Approximates linear transformation
Temperature	0.23	Mitigates skewed distribution
Sunshine Duration	0.1	Mitigates skewed distribution
NO_2	0.05	Mitigates skewed distribution
Wind Speed	0	Logarithmic transformation
Relative Humidity	0.18	Mitigates skewed distribution
Air Pressure	1	No transformation needed

This study adopts a rigorous machine learning modeling framework to construct and validate five predictive models: XGBoost [20], LightGBM [21], RF [22], SVM [23], and Backpropagation Neural Network (BPNN) [24]. The complete features after feature selection are shown in Table 3.

Table 3: Final selected feature list

Features	Description
Day Order	Sequential day within the year, capturing seasonality
Temperature	Daily mean temperature (°C)
Sunshine Duration	Actual daily sunshine hours (h)
NO_2	Daily mean NO_2 concentration ($\mu g/m^3$)
Wind Speed	Daily mean wind speed (m/s)
Relative Humidity	Daily mean relative humidity (%)
Air Pressure	Daily mean atmospheric pressure (hPa)

Subsequently, Bayesian optimization algorithm is used to iteratively optimize the core hyperparameters of each model. The specific values of the hyperparameters of this research model are shown in Table 4.

Table 4: Final model hyperparameters

Model	Hyperparameter	Final Value
XGBoost	Learning rate	0.05
	Maximum tree depth	6
	Subsample ratio of training instances	0.8
	Subsample ratio of features per tree	0.7
	Number of boosting trees	500
LightGBM	Learning rate	0.03
	Maximum number of leaves	31
	Maximum tree depth	7
	Number of boosting trees	400
RF	Number of trees	300
	Maximum tree depth	10
	Minimum samples required to split	2
	Minimum samples required at leaf node	1
SVM	Penalty parameter	10
	Kernel coefficient	0.01
BP Neural Network	Neurons in hidden layers	(50,30)
	Initial learning rate	0.001
	Maximum iterations	500

In the model training phase, time-series stratified sampling is adopted for dataset division, with 70% allocated to the training set and 30% to the test set. Subsequently, dual-track validation involving time-sliding window prediction and independent test set is conducted to comprehensively evaluate the generalization ability of the model and its performance differences across the whole country and key regions. The workflow of machine learning-based ozone concentration prediction is shown in Figure 3 [25].

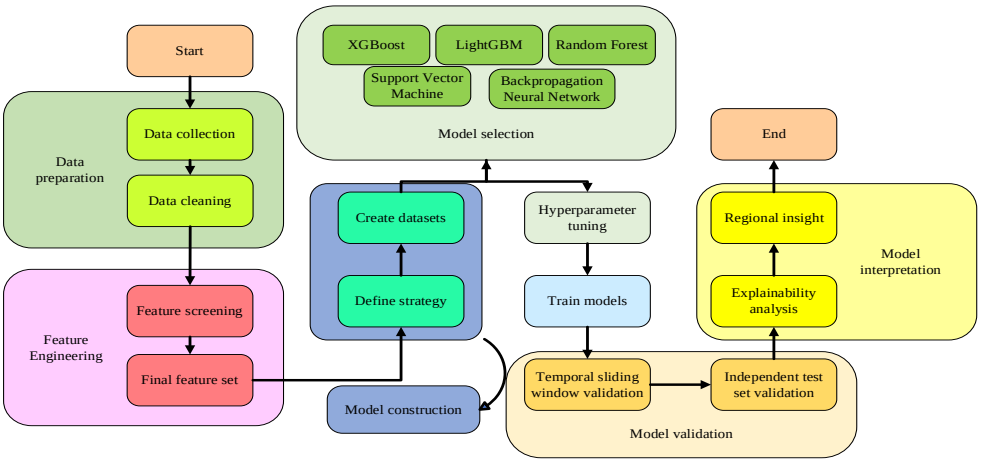


Figure 3: Ozone concentration prediction process based on machine learning

To comprehensively evaluate the predictive performance of each model, this study adopts the following six evaluation metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2), and Nash-Sutcliffe Efficiency Coefficient (NSE). The equations for each metric are defined as follows [26]:

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

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (6)$$

$$R^2 = 1 - \frac{i=1}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (7)$$

$$NSE = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (8)$$

n is the total number of samples. y_i is the i-th observed value (true value). $\hat{y}_i$ is the i-th predicted value, and $\bar{y}$ is the arithmetic average of true values.

3.3 Interpretability analysis of prediction models based on SHAP value

To quantify the contribution of each feature to ozone prediction from a game-theoretic perspective, this study introduces the SHAP method [27]. In the specific implementation process, considering that the prediction models in this study are mainly based on tree models, the TreeSHAP variant is adopted to improve computational efficiency and interpretation accuracy. The TreeSHAP algorithm is optimized for tree structures, enabling fast and accurate calculation of feature contribution while ensuring the fair distribution principle of Shapley values, which is suitable for interpretive analysis of large-scale environmental monitoring data. The interpretability analysis workflow based on SHAP values is illustrated in Figure 4 [28].

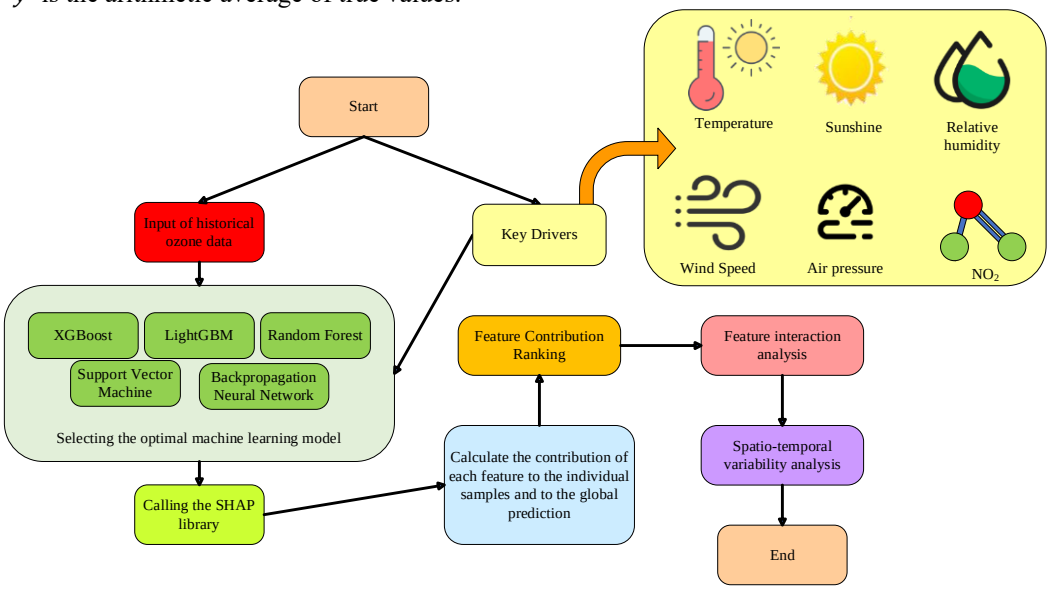


Figure 4: Explanatory analysis process of prediction model based on SHAP value

Let the feature set $F = [1,2,\ldots,D]$ (D features in total), and define the calculation formula of the value function $v(S)$ as follows [29]:

$$v(S) = \mathbb{E}[f(X)|X_S = x_S] \quad (9)$$

$v(S)$ is the expected output of the model prediction when the current sample value x_S is taken for any subset $S \sqsubseteq F$ while preserving S features. Shapley value ϕ_k represents the marginal average incremental contribution

of the kth feature in all possible cooperative subsets, and its calculation equation is as follows [30]:

$$\phi_k = \sum_{S \subseteq F \setminus (k)} \frac{|S|!(D-|S|-1)!}{D!} [v(S \cup \{k\}) - v(S)] \quad (10)$$

$D!$ is the number of all features, $|S|!(D-|S|-1)!$ is the normalization factor of subset weight, and $[v(S \cup \{k\}) - v(S)]$ is the incremental income of feature k after adding subset S. Taking the primary model prediction as a "cooperative game" to distribute income, the value function of sample x can be obtained, and the calculation equation is as follows:

$$v_x(S) = \mathbb{E}_{X_{\setminus S}}[f(x_S, X_{\setminus S})] - \mathbb{E}[f(X)] \quad (11)$$

x_S represents the observed value of the sample on the feature of subset S. $X_{\setminus S}$ represents the random distribution of other features. Therefore, the equation for calculating the SHAP value of the k-th feature under sample x is as follows:

$$\phi_k = \sum_{S \subseteq F \setminus (k)} \frac{|S|!(D-|S|-1)!}{D!} [v_x(S \cup \{k\}) - v_x(S)] \quad (12)$$

$$f(x) = \phi_0 + \sum_{k=1}^{D} \phi_k \quad (13)$$

$\phi_0 = \mathbb{E}[f(X)]$ is the baseline prediction value of the model, and $f(x)$ is additivity.

4 Experimental performance assessment results

4.1 Performance assessment results

The comparative results of the annual average ozone concentration in major regions of China are shown in Figure 5.

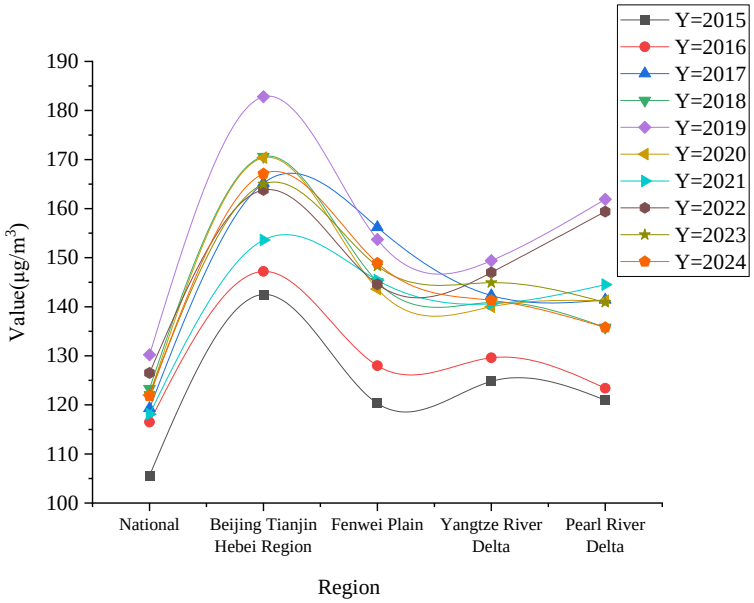
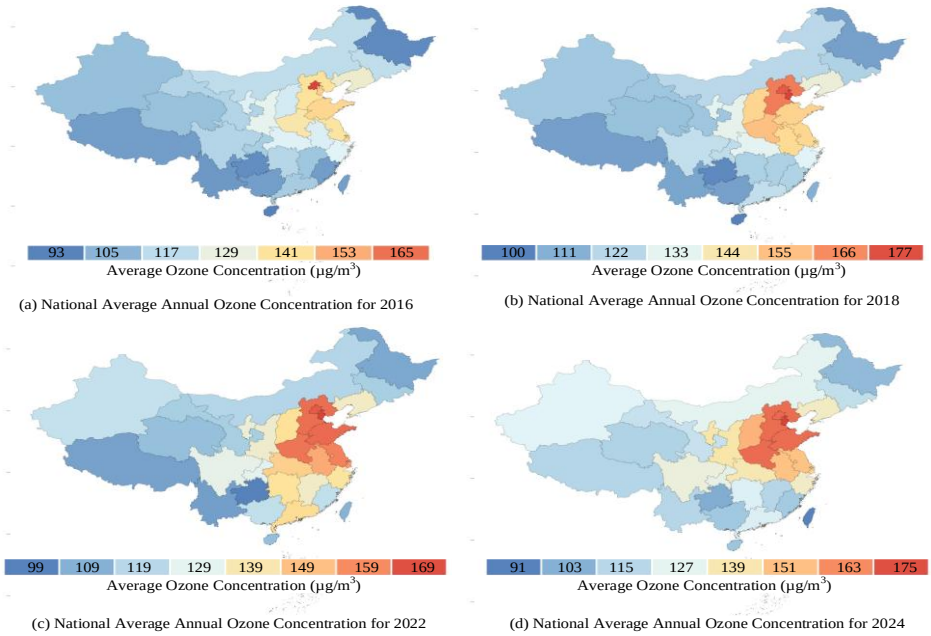


Figure 5: Comparative results of time variation of average annual ozone concentration in major regions of China

In Figure 5, the national ozone concentration presents a three-stage trend of "rapid rise-short-term decline-stabilization at a high level": the baseline value was 105.6 μg/m³ in 2015, reached a peak of 130.2 μg/m³ in 2019 (with an increase of 23.3%), dropped to 122 μg/m³ in 2020 due to emission reduction during the epidemic (a decrease of 6.3%), rebounded to 126.5 μg/m³ in 2022, and then stabilized at around 122 μg/m³ from 2023 to 2024, which was still 15.3% higher than that in 2015. At the regional level, the Beijing-Tianjin-Hebei region suffered from persistent severe pollution, with a concentration of 167.1 μg/m³ in 2024 (an increase of 17.2% over the decade). The Fenwei Plain showed a fluctuating downward trend after an abnormal surge in 2017, and its concentration in 2024 was 23.8% higher than that in 2015. The Yangtze River Delta remained relatively stable, with a concentration in 2024 5.4% lower than the peak in 2019, making it the only region with a continuous downward trend. The Pearl River Delta experienced severe fluctuations, with a concentration dropping to 135.8 μg/m³ in 2024 (a decrease of 16.1% from the peak), but the stability of emission reduction measures was insufficient.

The spatial distribution results and time distribution results of the average annual ozone concentration in various provinces in China are shown in Figs. 6 and 7. The gradient color in the figure indicates the ozone concentration in different regions (low in blue and high in red).



(a) National Average Annual Ozone Concentration for 2016

(b) National Average Annual Ozone Concentration for 2018

(c) National Average Annual Ozone Concentration for 2022

(d) National Average Annual Ozone Concentration for 2024

Figure 6: Spatial distribution results of average annual ozone concentration in provinces of China

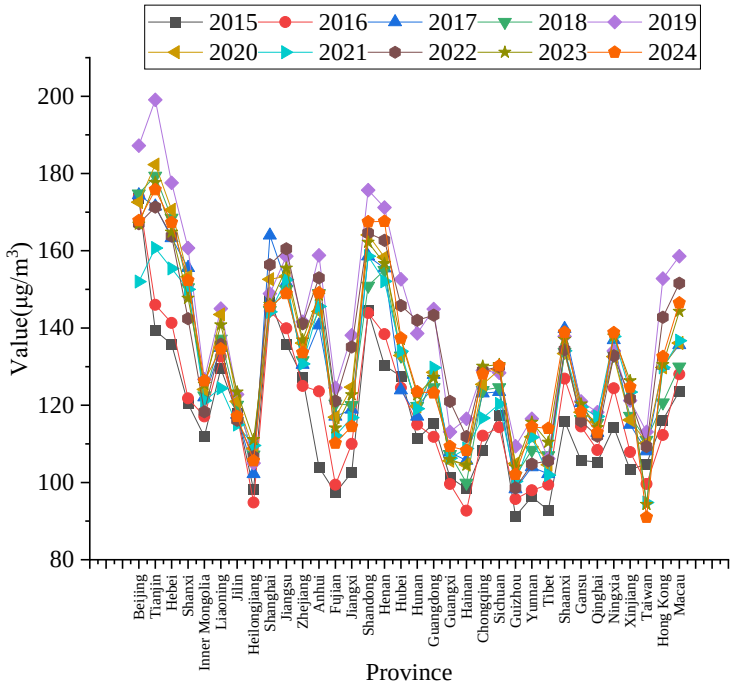
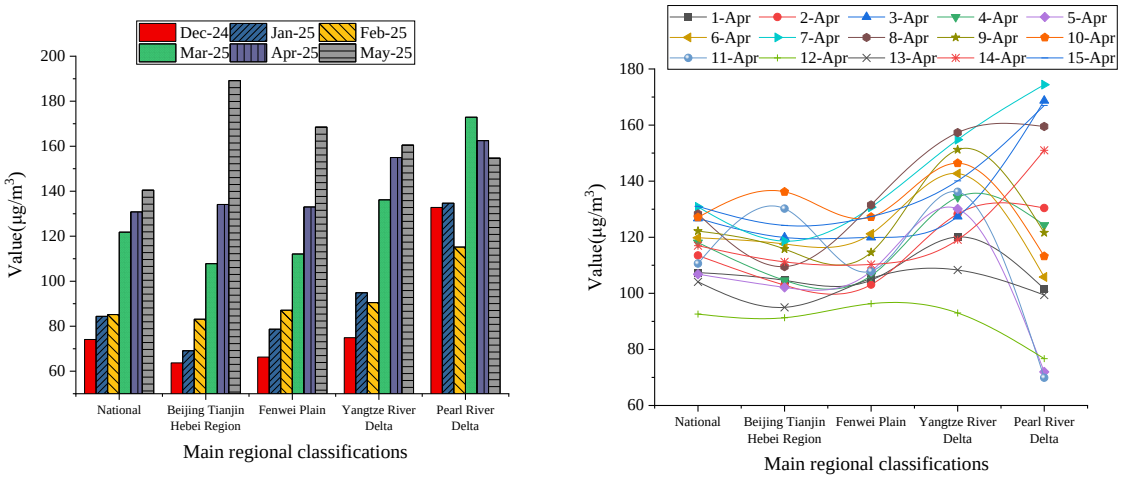


Figure 7: Comparative results of time variation of average annual ozone concentration in provinces of China

In Figure 6 and Figure 7, the annual average ozone concentration across provinces in China presents a spatial pattern of "higher in the east than in the west, and higher in the north than in the south": Provinces with high levels of industrialization and urbanization in the central and eastern regions (such as those in the Beijing-Tianjin-Hebei region, Shandong, Henan, as well as Jiangsu, Shanghai, Zhejiang in the Yangtze River Delta, and Guangdong in the Pearl River Delta) have relatively high ozone concentrations, mostly exceeding 150 μg/m³. Provinces in the western region such as Tibet, Guizhou, and Yunnan have lower concentrations, mostly around 100 μg/m³. This pollution pattern is related to economic development, industrial emissions, and meteorological conditions. In terms of time, the annual average ozone concentration across all provinces in China generally shows a fluctuating upward trend: it continued to rise to a peak from 2015 to 2019, decreased in 2020 due to the epidemic, rebounded in some provinces starting from 2021 and remained at a high level from 2022 to 2023. In 2024, some provinces saw a slight decrease, but the overall concentration still fluctuated at a high level, indicating significant pressure on pollution control.

Figure 8 presents a comparison of the monthly and

daily average ozone concentrations over time for major regions nationwide.



(a) Average monthly ozone concentration

(b) Average daily ozone concentration in April 2025

Figure 8: Comparative results of time variation of average monthly and daily ozone concentration in main areas of China

In Figure 8a, the monthly average concentration in all regions shows an upward trend from winter to spring. The national average increases from 74.1 μg/m³ to 140.5 μg/m³, with the Beijing-Tianjin-Hebei region recording the largest increase (from 63.7 μg/m³ to 189.2 μg/m³). The Pearl River Delta has the highest initial value (132.8 μg/m³), but it decreases slightly after reaching a peak of 172.9 μg/m³ in March. In Figure 8b, the national average of daily average concentration rises from 107.4 μg/m³ to 131.0 μg/m³ and has significant low points, while the Pearl River Delta exhibits the most severe fluctuations (ranging from 69.9 μg/m³ to 174.4 μg/m³). Overall, the monthly and daily average concentrations in the Beijing-Tianjin-Hebei region and the Yangtze River Delta rise steadily, while the Pearl River Delta and the Fenwei Plain show stronger spatiotemporal heterogeneity. This highlights the seasonal accumulation and short-term suddenness of ozone pollution.

In this study, rolling time-series K-fold cross-validation (K=5) is adopted for model training and testing to ensure the robustness of model evaluation. The predictive capabilities of different models for ozone concentrations nationwide and in key regions are presented in Table 5.

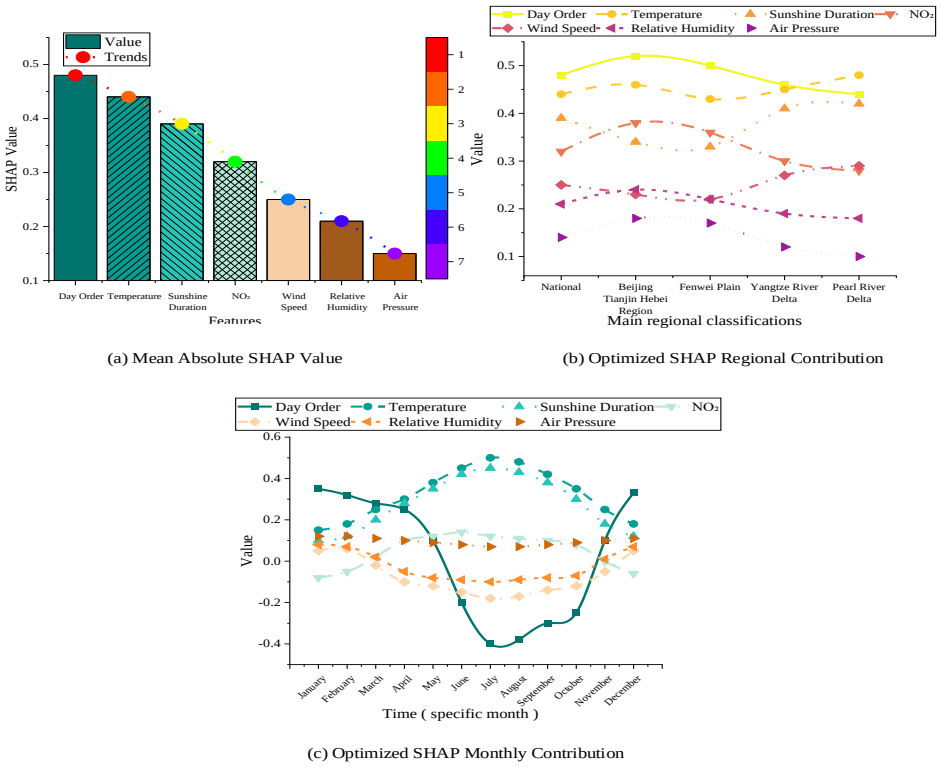
Table 5: The prediction ability of different models for ozone concentration in the whole country and key areas

Region	Model	MAE (μg/m³)	RMSE (μg/m³)	R^2	NSE	MAE vs XGBoost t-test (p-value)	RMSE vs XGBoost t-test (p-value)
National	DT	16.3±1.2	23.2±2.1	0.738±0.03	0.73±0.03	<0.001	<0.001
	RF	12.4±0.9	17.3±1.4	0.854±0.02	0.85±0.02	0.02	0.03
	SVM	19.3±1.4	26.1±2.2	0.669±0.03	0.66±0.03	<0.001	<0.001
	BP	19.5±1.5	26.1±2.3	0.669±0.03	0.66±0.03	<0.001	<0.001
	XGBoost	11.3±0.8	15.6±1.1	0.882±0.02	0.88±0.02	-	-
	LGBM	11.5±0.9	16.0±1.2	0.878±0.02	0.87±0.02	0.21	0.25
Beijing–Tianjin–Hebei Region	DT	17.5±1.3	24.5±2.2	0.720±0.03	0.71±0.03	<0.001	<0.001
	RF	13.0±1.0	18.0±1.5	0.840±0.02	0.83±0.02	0.03	0.04
	SVM	20.0±1.5	27.0±2.3	0.650±0.03	0.64±0.03	<0.001	<0.001
	BP	20.2±1.5	27.2±2.4	0.650±0.03	0.64±0.03	<0.001	<0.001
	XGBoost	12.0±0.9	16.5±1.2	0.870±0.02	0.86±0.02	-	-

Fenwei Plain	LGBM	12.1±0.9	16.7±1.3	0.865±0.02	0.86±0.02	0.28	0.3
	DT	17.0±1.2	24.0±2.1	0.730±0.03	0.72±0.03	<0.001	<0.001
	RF	12.8±0.9	17.8±1.4	0.850±0.02	0.84±0.02	0.04	0.05
	SVM	19.8±1.4	26.8±2.2	0.660±0.03	0.65±0.03	<0.001	<0.001
	BP	20.0±1.5	27.0±2.3	0.660±0.03	0.65±0.03	<0.001	<0.001
	XGBoost	11.8±0.8	16.2±1.1	0.880±0.02	0.87±0.02	-	-
Yangtze River Delta	LGBM	12.0±0.9	16.5±1.2	0.875±0.02	0.87±0.02	0.23	0.27
	DT	15.0±1.1	22.0±2.0	0.750±0.03	0.74±0.03	<0.001	<0.001
	RF	11.2±0.8	16.0±1.3	0.860±0.02	0.85±0.02	0.02	0.03
	SVM	18.0±1.3	25.0±2.1	0.680±0.03	0.67±0.03	<0.001	<0.001
	BP	18.5±1.4	25.5±2.2	0.680±0.03	0.67±0.03	<0.001	<0.001
	XGBoost	10.5±0.7	15.0±1.1	0.890±0.02	0.88±0.02	-	-
Pearl River Delta	LGBM	10.7±0.8	15.2±1.1	0.885±0.02	0.88±0.02	0.24	0.28
	DT	14.0±1.0	21.0±1.9	0.760±0.03	0.75±0.03	<0.001	<0.001
	RF	10.5±0.7	15.0±1.2	0.870±0.02	0.86±0.02	0.02	0.03
	SVM	17.5±1.3	24.0±2.1	0.690±0.03	0.68±0.03	<0.001	<0.001
	BP	18.0±1.4	24.5±2.2	0.690±0.03	0.68±0.03	<0.001	<0.001
	XGBoost	10.0±0.7	14.0±1.0	0.900±0.02	0.89±0.02	-	-
	LGBM	10.2±0.7	14.3±1.0	0.888±0.02	0.89±0.02	0.2	0.22

In Table 5, the XGBoost model performs the best with the highest prediction accuracy and good stability nationwide and in all major regions. On a national scale, the MAE of XGBoost is 11.3 μg/m³, the RMSE is 15.6 μg/m³, the R^2 is 0.882, and the NSE is 0.88, which is significantly better than traditional regression and neural network models. At the regional level, the XGBoost model achieves the best performance in the Pearl River Delta, with an MAE of only 10.0 μg/m³, an RMSE of 14.0 μg/m³, and an R^2 of 0.900. In the Beijing-Tianjin-Hebei region and the Fenwei Plain, the MAE values are 12.0 μg/m³ and 11.8 μg/m³ respectively, the RMSE values are 16.5 μg/m³ and 16.2 μg/m³ respectively, and the R^2 values both exceed 0.86. This indicates that the ensemble learning method has significant advantages in capturing the spatiotemporal variation characteristics of ozone concentration. Moreover, the results of the paired t-test show that the differences in MAE and RMSE between XGBoost and other models are statistically significant ($p<0.05$), which further verifies the reliability and scientific nature of its prediction performance.

The interpretable analysis results based on SHAP value are shown in Figure 9.



(a) Mean Absolute SHAP Value

(b) Optimized SHAP Regional Contribution

(c) Optimized SHAP Monthly Contribution

Figure 9: Interpretable analysis results based on SHAP value

Analysis of Figure 9 indicates that, at the national scale, calendar day, temperature, and sunshine duration are the dominant factors driving ozone concentration variations, consistently showing significant influence. The average contribution value of NO_2 in the Beijing-Tianjin-Hebei region is 0.38, and that in the Fenwei Plain is 0.36. Both values are higher than the national average of 0.32 and significantly higher than those in the Yangtze River Delta and the Pearl River Delta. This indicates that in the Beijing-Tianjin-Hebei region and the Fenwei Plain, the contribution ratio of NO_2 to the model prediction results is significantly higher than that in other regions, which reflects the key role of precursor emissions in photochemical reactions in these regions. In contrast, the Yangtze River Delta and Pearl River Delta regions exhibit greater contributions from temperature and sunshine duration, with the strongest promoting effects observed in the Pearl River Delta. Wind speed and relative humidity have comparatively smaller overall contributions. However, wind speed plays a relatively larger role in the Pearl River Delta and Yangtze River Delta, indicating that atmospheric dispersion conditions in these regions significantly modulate ozone concentration changes. These findings provide targeted guidance for region-specific ozone pollution control strategies across China.

The SHAP summary results of the model and the dependence results of the top three important features are shown in Figure 10.

In Figure 10, there are obvious differences in the importance and action direction of each feature on the model output. From the perspective of the SHAP value of day of year, it generally makes a positive contribution to the prediction results. With the increase of day of year, the SHAP value shows an upward trend, indicating that the model is sensitive to the cumulative effect of the time series. For temperature, the SHAP value fluctuates greatly: samples with high temperature correspond to positive contributions, while samples with low temperature correspond to negative contributions. This reflects the nonlinear dependence of temperature on the prediction results. Regarding sunshine duration, its SHAP value contributes more to the model output when the sunshine value is relatively low or extremely high, while the variation range in the middle interval is relatively small. This indicates that sunshine duration has an obvious threshold effect on the prediction results. On the whole, the dependence plots of these three features show that day of year has a steady positive dependence, temperature has a nonlinear fluctuating dependence, and sunshine duration has a threshold-type dependence. These plots provide an intuitive reference for the interpretation of model features.

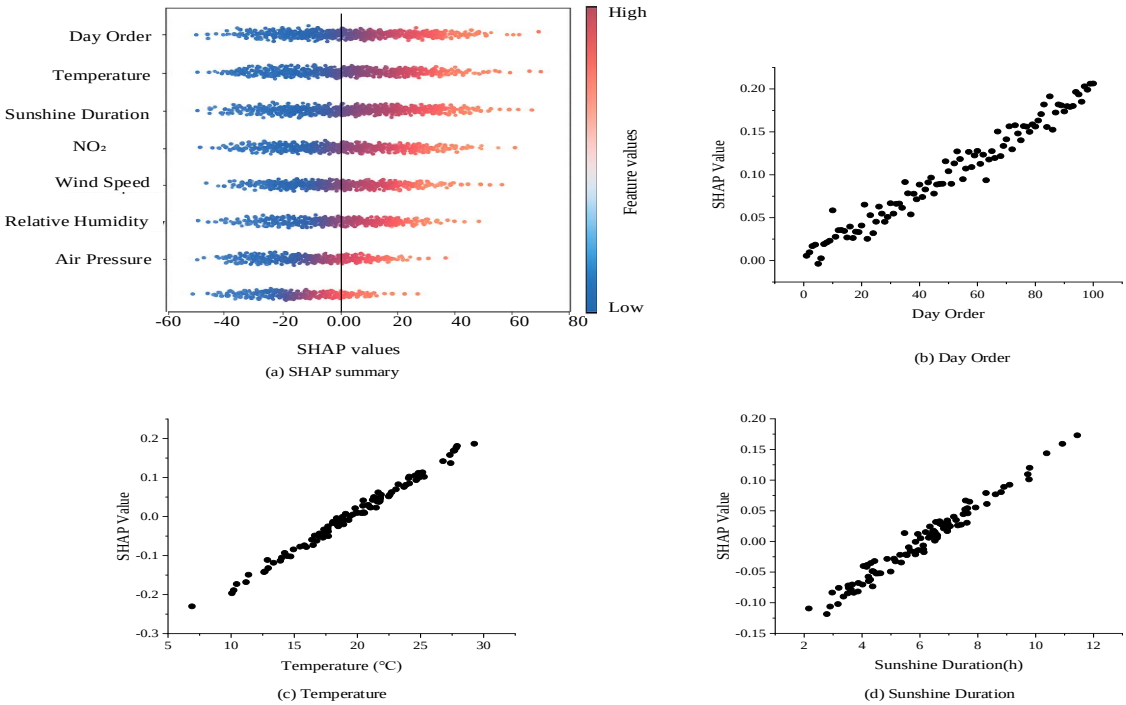


Figure 10: Interpretable analysis results based on SHAP value

4.2 Discussions

Compared with existing CTMs, statistical regression models, and traditional machine learning models, the model in this study performs excellently in both prediction accuracy and robustness. On a national scale, the XGBoost model has an MAE of 11.3 μg/m³, an RMSE of 15.6 μg/m³, and an R^2 of 0.882, which is significantly better than decision tree, support vector machine, and BPNN models ($p<0.05$). In the Pearl River Delta region, the MAE of the XGBoost model is only 10.0 μg/m³, the RMSE is 14.0 μg/m³, and the R^2 reaches 0.900. The main reasons for the performance advantages of this study are as follows: High spatiotemporal resolution online observation data are adopted, which enhances the comprehensiveness and representativeness of the model input information. Through feature selection and lag feature construction, the nonlinear impacts of precursor emissions and meteorological conditions on ozone formation are effectively captured. Based on the ensemble learning algorithm, the model exhibits excellent fitting ability and generalization performance in handling the nonlinear coupling relationships of multiple variables, thereby significantly improving the prediction accuracy and stability.

Based on the results of SHAP analysis, this study reveals the regional heterogeneity of driving factors for ozone formation. In regions dominated by precursor emissions, such as the Beijing-Tianjin-Hebei region and the Fenwei Plain, NO_2 and VOCs make significant contributions to changes in ozone concentration. In contrast, regions dominated by meteorological conditions, such as the Yangtze River Delta and the Pearl River Delta, are mainly affected by factors like temperature and sunshine. This indicates that ozone pollution control in different regions should adopt differentiated strategies targeting the main driving factors to improve the efficiency and accuracy of prevention and control.

On the basis of scientific analysis, this study can further provide a quantitative basis for policy-making. On the one hand, for regions dominated by precursor emissions, priority should be given to several key strategies. These include: Strengthening the coordinated emission reduction of key emission sources such as NO_2 and VOCs. Optimizing industrial, transportation, and energy structures. Realizing precise source control in combination with high-frequency emission monitoring. On the other hand, for regions dominated by meteorological conditions, it is necessary to strengthen dynamic regulation and emergency response. This is particularly important under extreme meteorological conditions such as high temperature and low wind speed. Specific measures should include temporary traffic restrictions and staggered production for high-emission enterprises. In addition, nationwide, the ozone online monitoring network and integrated intelligent forecasting system should be continuously improved. Through data sharing and cross-departmental collaborative governance, the prevention and control of ozone pollution should be promoted towards digitalization, intelligence, and refinement to provide reliable support for public health and ecological environment security.

5 Conclusion

By integrating multi-scale ozone observation data with an interpretable machine learning framework, this study systematically reveals the spatiotemporal evolution characteristics and formation mechanisms of ozone pollution in China. The results show that the national

ozone concentration from 2015 to 2024 presents a three-stage trend of "rise-decline-stabilization", and the concentration in 2024 is still 15.3% higher than the baseline value in 2015. At the regional level, the Beijing-Tianjin-Hebei region and the Fenwei Plain maintain a continuously high concentration level, highlighting the pressure of pollution control. Meanwhile, the economically dense areas in eastern China have exceeded the standard for a long time, showing a differentiation pattern of "higher in the east and lower in the west", which confirms that industrial emissions and urbanization are the core features. Based on the prediction of the XGBoost model and SHAP analysis, the ozone formation mechanism shows obvious regional heterogeneity: The Beijing-Tianjin-Hebei region and the Fenwei Plain are mainly dominated by the emission of precursors such as NO_2 (with SHAP contribution ratio > 0.35), while the Yangtze River Delta and the Pearl River Delta are sensitive to temperature and sunshine. The model achieves the best prediction performance in the Pearl River Delta, providing a scientific reference for dynamic and precise management and control.

The limitations of this study are mainly reflected in factor coverage and model input: limited by existing monitoring data and available variables, some short-term sudden pollution events and complex chemical reaction mechanisms have not been fully captured. In addition, the quantitative analysis of external factors such as social and economic development and policy intervention is relatively limited. Future research can improve the model's ability to predict short-term pollution events. This can be achieved by introducing socio-economic datasets such as traffic flow and industrial activities. At the same time, it is necessary to strengthen the integration of multi-source and multi-dimensional data. This will enhance the model's adaptability to extreme meteorological conditions and nonlinear changes. In terms of model deployment and application, this study also provides practical implications: although XGBoost has high prediction accuracy, there are certain costs in training time and SHAP feature contribution calculation. In the future, more efficient computing strategies or model compression methods can be explored to balance accuracy and computing efficiency, thereby providing operable guidance for model developers and environmental management practitioners.

References

[1] Kong L, Song M, Li X, Liu Y, Lu S, Zeng L, et al. Analysis of China's PM2. 5 and ozone coordinated control strategy based on the observation data from 2015 to 2020. Journal of Environmental Sciences, 2024, 138(1): 385-394. https://doi.org/10.1016/j.jes.2023.03.030

[2] Li Z, Bi J, Liu Y, Hu X. Forecasting O_3 and NO_2 concentrations with spatiotemporally continuous coverage in southeastern China using a Machine learning approach. Environment International, 2025, 195(1): 109249. https://doi.org/10.1016/j.envint.2024.109249

[3] Carbo-Bustinza N, Iftikhar H, Belmonte M, Cabello-Torres RJ, Cruz ARHDL, López-Gonzales JL. Short-term forecasting of Ozone concentration in metropolitan Lima using hybrid combinations of time series models. Applied Sciences, 2023, 13(18): 10514. https://doi.org/10.3390/app131810514

[4] Rahman A, Nasher N M R. Forecasting hourly ozone concentration using functional time series model-A case study in the coastal area of bangladesh. Environmental Modeling & Assessment, 2024, 29(1): 125-134. https://doi.org/10.1007/s10666-023-09928-8

[5] Xie J, Tang X, Zheng F, Wang X, Ding N, Song Y, et al. Improvement of the ozone forecast over Beijing through combining the chemical transport model with multiple machine learning methods. Atmospheric Pollution Research, 2024, 15(8): 102184. https://doi.org/10.1016/j.apr.2024.102184

[6] Lundberg SM, Lee SI. A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 2017, 30. https://doi.org/10.48550/arXiv.1705.07874

[7] Lasry F, Coll I, Fayet S, Havre M, Vautard R. Short-term measures for the control of ozone peaks: expertise from CTM simulations. Journal of atmospheric chemistry, 2007, 57(1): 107-134. https://doi.org/10.1007/s10874-007-9062-1

[8] Li J, Jang J, Zhu Y, Wang C, Xing S, Dong J, et al. Development of a recurrent spatiotemporal deep-learning method coupled with data fusion for correction of hourly ozone forecasts. Environmental Pollution, 2023, 335(1): 122291. https://doi.org/10.1016/j.envpol.2023.122291

[9] Liu Z, Lu Z, Zhu W, Yuan J, Cao Z, Cao T, et al. Comparison of machine learning methods for predicting ground-level ozone pollution in Beijing. Frontiers in Environmental Science, 2025, 13(1): 1561794. https://doi.org/10.3389/fenvs.2025.1561794

[10] Tang B, Stanier CO, Carmichael GR, Gao M. Ozone, nitrogen dioxide, and PM2. 5 estimation from observation-model machine learning fusion over S. Korea: Influence of observation density, chemical transport model resolution, and geostationary remotely sensed AOD. Atmospheric Environment, 2024, 331(1): 120603. https://doi.org/10.1016/j.atmosenv.2024.120603

[11] Marvin D, Nespoli L, Strepparava D, Medici V. A data-driven approach to forecasting ground-level ozone concentration. International Journal of Forecasting, 2022, 38(3): 970-987. 10.1016/j.ijforecast.2021.07.008

[12] Gagliardi R V, Andenna C. Exploring the Influencing Factors of Surface Ozone Variability by Explainable Machine Learning: A Case Study in the Basilicata Region (Southern Italy). Atmosphere, 2025, 16(5): 491. https://doi.org/10.3390/atmos16050491

[13] Haagen-Smit A J. Chemistry and physiology of Los Angeles smog. Industrial & Engineering Chemistry, 1952, 44(6): 1342-1346. https://doi.org/10.1021/ie50510a045

[14] Chapman S. XXXV. On ozone and atomic oxygen in the upper atmosphere. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, 1930, 10(64): 369-383. https://doi.org/10.1080/14786443009461588

[15] Chapman S. Bakerian Lecture-Some phenomena of the upper atmosphere. Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character, 1931, 132(820): 353-374. https://doi.org/10.1098/rspa.1931.0105

[16] Chu W, Li H, Ji Y, Zhang X, Xue L, Gao J, et al. Research on ozone formation sensitivity based on observational methods: Development history, methodology, and application and prospects in China. Journal of Environmental Sciences, 2024, 138(1): 543-560. https://doi.org/10.1016/j.jes.2023.02.052

[17] Wang X, Shao T, Qin J, Li Y, Long X, Jiang D, et al. Promotion effect of micro-hole in dielectric on ozone generation of dielectric barrier discharge. Ozone: Science & Engineering, 2024, 46(4): 345-354. https://doi.org/10.1080/01919512.2023.2301548

[18] Li Y, Wu Z, Ji Y, Chen T, Li H, Gao R, et al. Comparison of the ozone formation mechanisms and VOCs apportionment in different ozone pollution episodes in urban Beijing in 2019 and 2020: Insights for ozone pollution control strategies. Science of The Total Environment, 2024, 908(1): 168332. https://doi.org/10.1016/j.scitotenv.2023.168332

[19] Geng G, Xiao Q, Liu S, Liu X, Cheng J, et al. Tracking air pollution in China: near real-time PM2. 5 retrievals from multisource data fusion. Environmental Science & Technology, 2021, 55(17): 12106-12115. https://doi.org/10.1021/acs.est.1c01863

[20] Guo Q, He Z, Wang Z. The characteristics of air quality changes in Hohhot City in China and their relationship with meteorological and socio-economic factors. Aerosol and Air Quality Research, 2024, 24(5): 230274. https://doi.org/10.4209/aaqr.230274

[21] Kumar G D, Tyagi S, Pradhan K C, Shah A. District-level rainfall and cloudburst prediction using XGBoost: A machine learning approach for early warning systems. Informatica, 2025, 49(2). https://doi.org/10.31449/inf.v49i2.7612

[22] Chen C. Football match analysis and prediction based on light GBM decision algorithm. Informatica, 2024, 48(16). https://doi.org/10.31449/inf.v48i16.6263

[23] Oukhouya M H, Angour N, Aboutabit N, Hafidi I. Comparative analysis of ARDL, LSTM, and XGBoost models for forecasting the moroccan stock market during the COVID-19 pandemic. Informatica, 2025, 49(14). https://doi.org/10.31449/inf.v49i14.5751

[24] Rawat R, Raj ASA, Chakrawarti RK, Sankaran KS, Sarangi SK, Rawat H, et al. Enhanced cybercrime detection on twitter using aho-corasick algorithm and machine learning techniques. Informatica, 2024, 48(18). https://doi.org/10.31449/inf.v48i18.6272

[25] Lai H. Predicting the growth value of technology enterprises with an optimized back-propagation neural network. Informatica, 2024, 48(16). https://doi.org/10.31449/inf.v48i16.6437

[26] Cheng M, Fang F, Navon IM, Zheng J, Zhu J, Pain C. Assessing uncertainty and heterogeneity in machine learning-based spatiotemporal ozone prediction in Beijing-Tianjin-Hebei region in China. Science of the Total Environment, 2023, 881(1): 163146. https://doi.org/10.1016/j.scitotenv.2023.163146

[27] Yao T, Lu S, Wang Y, Li X, Ye H, Duan Y, et al. Revealing the drivers of surface ozone pollution by explainable machine learning and satellite observations in Hangzhou Bay, China. Journal of Cleaner Production, 2024, 440(2): 140938. https://doi.org/10.1016/j.jclepro.2024.140938

[28] Zhao B, Wang S, Hao J. Challenges and perspectives of air pollution control in China. Frontiers of Environmental Science & Engineering, 2024, 18(6): 68. https://doi.org/10.1007/s11783-024-1828-z

[29] Nath S J, Girach I A, Harithasree S, Bhuyan K, Ojha N, Kumar M. Urban ozone variability using automated machine learning: inference from different feature importance schemes. Environmental Monitoring and Assessment, 2024, 196(4): 393. https://doi.org/10.1007/s10661-024-12549-7

[30] Han L, Zhao J, Gao Y, et al. Prediction and evaluation of spatial distributions of ozone and urban heat island using a machine learning modified land use regression method. Sustainable Cities and Society, 2022, 78(1): 103643. https://doi.org/10.1016/j.scs.2021.103643

MOGO-AFNNet: A Deep Learning and Multi-Objective Genetic Algorithm Framework for Intelligent logistics Warehouse Layout and Inventory Control

Kai Wang [1*], Yameng Bai [2]
[1]College of Innovation and Entrepreneurship, jiaozuo university, Jiaozuo,Henan ,454000, China
[2]College of Information Engineering, jiaozuo university, Jiaozuo,Henan , 454000, China
E-mail: wangkai198980@hotmail.com
[*]Corresponding author

Keywords: logistics, warehouse, inventory regulation, multi-objective genetic algorithm-driven adaptive fuzzy neuro network (MOGO-AFNNet), stockouts

Received: August 1, 2025

In modern logistics, warehouse layout and inventory control face challenges such as demand variability, inefficient space utilization, and frequent stockouts, where traditional analytical systems often fail to adapt in real time. To address these issues, this study proposes a Multi-Objective Genetic Algorithm-driven Adaptive Fuzzy Neuro Network (MOGO-AFNNet), which integrates an Adaptive Fuzzy Neuro Network (AFNNet) for demand forecasting with the NSGA-II genetic algorithm for multi-objective optimization. The Smart Logistics Supply Chain Dataset, comprising real-time IoT-based records of shipments, delays, stock levels, and operational costs, was employed. Data preprocessing was performed using Z-score normalization to standardize features, followed by Principal Component Analysis (PCA) to extract key variables such as reorder frequency, lead time variability, and item popularity. The AFNNet component enabled adaptive inventory regulation under uncertainty, while NSGA-II optimized warehouse layout and inventory strategies across conflicting KPIs. Experimental evaluation showed that the proposed framework achieved 97.5% accuracy, 98% precision, 96.8% recall, and a 97.9% F1-score, significantly outperforming baseline models. ANOVA confirmed significant performance differences among models ($F = 21.47$, $p = 0.0004$). These results demonstrate that MOGO-AFNNet offers a scalable and robust solution for intelligent logistics, reducing stockouts and enhancing warehouse efficiency in dynamic operational environments.

Povzetek: Članek predlaga MOGO-AFNNet, ki na logističnih podatkih združi napovedovanje povpraševanja z adaptivno mehko-nevronsko mrežo ter večciljno optimizacijo za sočasno optimiranje razporeditve skladišča in zalog, kar zmanjšuje primanjkljaje in dviga učinkovitost.

1 Introduction

Smart logistics warehouse design/inventory management is a revolutionary way of handling contemporary supply chains [1]. In the current dynamic and very competitive market, organizations are under a lot of pressure to maximize effectiveness, minimize operational expenses, and be quick in addressing their needs [2]. To achieve these expectations, the logistics businesses must move beyond traditional operations and adopt intelligent systems incorporating automation, real-time data, and smart technologies [3]. It is the design of a logistics warehouse, which defines the overall result of material management, storage, and distribution [4]. Considering a warehouse layout has been optimized, reducing travel time and maximizing use of space and free flow of items [5].

Adopting the intelligent design principles can dynamically reconfigure storage areas, enhance workflow, and enhance access to items that are frequently moved [6]. Figure 1 shows the logistics warehouse layout and inventory control.

Inventory control on its part guarantees the provision of the right quantity of goods at the right time, avoiding overstocking and under-stocking [7]. Smart inventory solutions include IoT sensors, Radio Frequency Identification (RFID) tracking, and Artificial Intelligence (AI) based demand prediction to ensure real-time accuracy and visibility [8]. The systems enable stock record automation, automatic triggering of replenishments, and elimination of human factors [9].

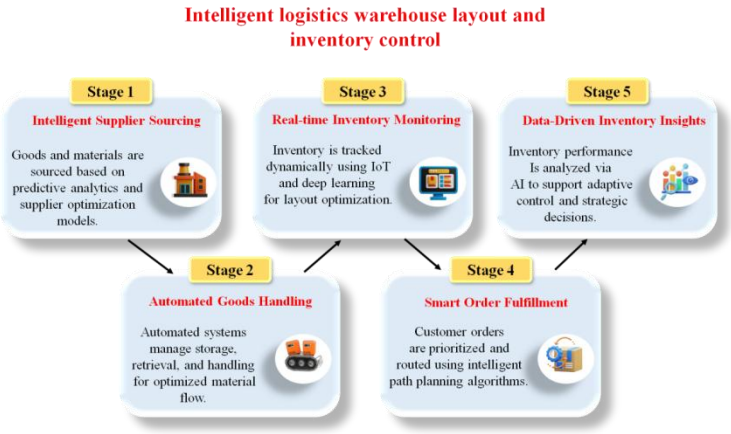


Figure 1: Five-stage intelligent logistics framework optimizing sourcing, handling, and inventory.

Smart warehouse design and inventory management, in combination with each other, create the core of the agile and resilient logistics networks. They make business processes simpler, as well as boost service levels and respond rapidly to market changes [10]. With logistics transforming into a digital operation, intelligent systems cannot be considered a luxury anymore but an essential element of the organizational processes that give more companies and a competitive advantage to keep pace with the dynamic customer demands. Conventional practices in logistics were dependent upon planning manually and rule-driven systems. Machine Learning (ML) and Deep Learning (DL) have made optimization of warehouse layout, warehouse demand forecasting, and inventory control in real-time more accurate and efficient, with methods including Support Vector Machine (SVM), K-Means Clustering, Artificial Neural Networks (ANN), and Long Short-Term Memory (LSTM) networks.

Although ML and DL approaches to logistics have various advantages, they have limitations that include data dependence, complicated model-building, interpretability, and inability to handle sudden changes. Also, its installation involves implementation of huge computational units, experienced human resources, and interconnection with established systems; it is costly and time-consuming for most organizations. The aim was to develop an intelligent logistics framework using a MOGO-AFNNet model that integrates adaptive fuzzy neural networks with a multi-objective genetic algorithm to optimize warehouse layout, improve goods flow, and enhance inventory control while minimizing costs and stockouts.

1.1 Key contribution

- Data collection: The real-time logistics data for optimizing supply chains, enabling predictive modeling, anomaly detection, and efficient warehouse and inventory management.
- Data preprocessing: Applied Z-Score normalization to standardize input features, ensuring consistent data scaling and improving model performance and convergence stability.
- Feature extraction: Used PCA to extract key variables, reduce dimensionality, and retain critical variance in logistics datasets.
- Proposed method: Integrated MOGO and AFNN to optimize warehouse layout and inventory control by balancing KPIs and adapting to demand uncertainty.

The structural framework of the research is listed as follows: The significant background of the research is provided in Section 1. A list of literature reviews was provided in Section 2. The method is explained in Section 3. Results and discussion section contained Section 4. Section 5 provides the conclusion.

1.2 Research questions

1. How can the integration of deep learning–based demand forecasting with multi-objective genetic algorithms improves both inventory management and warehouse layout optimization in dynamic logistics environments?
2. To what extent does the MOGO-AFNNet framework enhance operational efficiency, reduce stockouts, and maintain scalability compared to existing domain-specific intelligent warehousing methods?

2 Related works

Table 1 presents the reviewed studies employ AI, IoT, and optimization techniques to improve inventory control, path planning, warehouse layout, and resource scheduling. They show gains in efficiency, accuracy, cost reduction, and adaptability. However, most are domain-specific or limited in scalability, highlighting the need for integrated, real-world applicable frameworks.

Table 1: Summary of related works

Reference	Objective Function(s)	Method Used	Performance	Limitations
[11]	Minimize task execution time, cost, and maximize execution efficiency	Improved hybrid-parameter Ant Colony Algorithm with reinforcement learning	Achieved 16052 s packing time, 29865-yuan cost, 95.65% execution rate, 9.54% cost reduction	Validation limited to logistics scheduling tasks; generalizability to broader logistics contexts not addressed
[12]	Overcome control challenges in intelligent warehousing	Neuro-Fuzzy Dynamic Inventory Regulation Model (NFDIRM) combining Radial Basis Function Neural Network (RBFNN) + fuzzy logic	Higher inventory turnover, fewer stockouts, reduced costs vs. Economic Order Quantity (EOQ) & AutoRegressive Integrated Moving Average (ARIMA)	Complexity of implementation; real-world scalability not discussed
[13]	Minimize path length, completion time; maximize punctuality and dispatch success	Deep Reinforcement Learning (DRL) combined with CNN	Achieved 92–95% punctuality/dispatch success, reduced path length and time	Tested only; broader real-world scalability and dynamic uncertainties not fully validated
[14]	Optimize warehouse location & minimize distribution costs	Internet of Things (IoT) + Blockchain	Higher positioning accuracy, reduced computational load & costs	Applicability to large-scale logistics not fully validated
[15]	Intelligent crane scheduling in steel production	IoT + AI + Light Detection and Ranging (LiDAR) + Multi-Agent Reinforcement Learning (MARL) + optimized anti-swing control	Achieved unmanned operation & improved production	High technology & infrastructure cost; domain-specific
[16]	Onion buffer stock monitoring	IoT + Radio Frequency Identification (RFID) + cloud integration	Improved traceability, spoilage prevention, proactive environmental control	Focused on single product type (onions), generalization needed
[17]	Human & goods tracking in warehouses	You Only Look Once version 5 (YOLOv5) + Deep Simple Online and Realtime Tracking (DeepSORT)	Accurate, robust tracking within 30 ms	Limited to specific warehouse environments
[18]	Optimize e-commerce warehouse layout	Genetic Algorithm (GA)	Reduced handling costs; 39.25% improvement, outperformed Particle Swarm Optimization (PSO) & Simulated Annealing (SA)	May require high computational resources
[19]	Improve Warehouse Management (WM) efficiency & reduce costs	Intelligent decision system using GA, SA, and Ant Colony Optimization (ACO)	Significant efficiency & cost-effectiveness gains	Scalability to complex warehouses untested
[20]	Categorization of warehouse goods	Artificial Neural Network (ANN) enhanced by Black Hole Algorithm (BHA)	High accuracy with minimal input parameters	Focused on categorization only, not full warehouse management (WM)
[21]	Improve order picking efficiency	Smart lighting systems in warehouses	Promising efficiency improvements	Underexplored, needs more industrial applications

[22]	Enhance warehousing efficiency & safety	Programmable Logic Controller (PLC) + Human-Computer Interaction (HCI) integration	Better speed, space utilization, data efficiency, and reliability	Technology adaptation costs
[23]	Improve manual order picking	Smart Scale (Design Science Research, DSR; weight sensors)	Better accuracy, less human error, efficiency gains	Limited to lightweight items
[24]	Enable intelligent warehouse management & optimized distribution	IoT + AI integration	Real-time monitoring, optimized supply chain	Broad scope; practical case studies limited
[25]	Optimize path selection, task scheduling, resource allocation under dynamic demand	DRL with Multi-Task Learning (MTL) and Q-learning	Achieved 15% cost reduction, 85% resource utilization, strong adaptability in dynamic scenarios	Validation limited to simulations; real-world deployment and scalability not demonstrated

2.1 Research gaps

Prior research advanced intelligent warehousing through neuro-fuzzy inventory models [12], IoT-driven crane automation [15], and genetic algorithm–based layout optimization [18]. However, these solutions remained domain-specific, addressing isolated issues like inventory turnover, crane scheduling, or layout efficiency, with limited integration of predictive intelligence and multi-objective optimization. To overcome this, the proposed MOGO-AFNNet framework integrates deep learning–based demand forecasting with multi-objective genetic algorithms for layout and inventory optimization. This unified approach enhances adaptability, aligns with diverse KPIs, consolidates spatial and stock management, and delivers greater accuracy, fewer stockouts, and improved operational efficiency across logistics environments.

3 Methodology

The dataset provides real-time IoT-based logistics data for analyzing shipments, delays, asset tracking, and optimizing supply chain operations. Data preprocessing used Z-score normalization to standardize input features. PCA was applied for feature extraction, reducing dimensionality while retaining key variance. The proposed MOGO-AFNNet model integrates an Adaptive Fuzzy Neural Network for demand forecasting with a Multi-Objective Genetic Algorithm to optimize warehouse layout and inventory control strategies across multiple KPIs. Figure 2 illustrates the overall flow of the methodology.

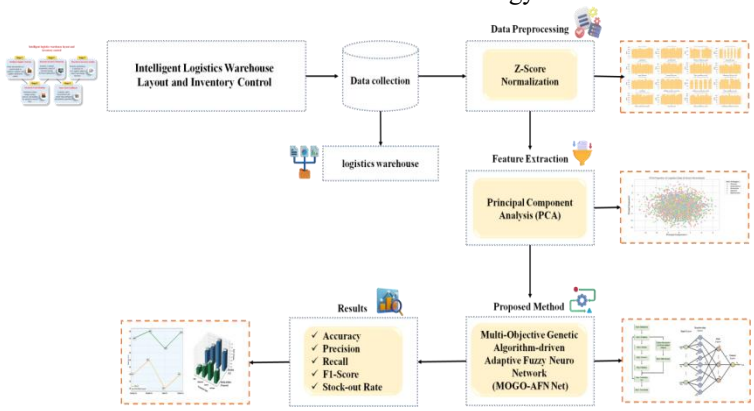


Figure 2: Methodology flow for intelligent logistics warehouse layout

3.1 Data collection

The Smart Logistics Supply Chain Dataset delivers real-time logistics and warehouse operations data. It includes time-stamped records of asset tracking, shipments, delays, package movements, and related categorical and numerical details captured over an extended period. Ideal for optimization, predictive modeling, supply chain simulations, and anomaly detection, it allows users to analyze operational workflow, delivery performance, and inventory throughput for smarter logistics decision-making. It includes stock levels, demand variability, lead time, reorder frequency, item

popularity, and operational costs. These attributes support predictive modeling, inventory optimization, and warehouse layout planning for intelligent logistics systems. The Smart Logistics Supply Chain Dataset was split 80% for training and 20% for testing to evaluate model performance

Source:https://www.kaggle.com/datasets/ziya07/logistics-warehouse-dataset/data.

3.2 Data preprocessing using Z-Score normalization

Z-Score normalization standardizes input features by transforming them to have zero mean and unit variance, enhancing the performance of deep learning models in intelligent warehouse layout and inventory control. This technique adjusted the scales of datasets to a standard scale where the mean was equal to zero and the standard deviation was one. The data had undergone preliminary processing to make it ready for primary processing and analysis. Z-score normalization, also known as zero normalization, involves dividing each feature's mean and standard deviation in a training dataset by several variables. The general formula in equation (1) quantifies the transformation.

$$d' = \frac{(d-\mu)}{\sigma} \quad (1)$$

In this equation, d represents the original data, μ is the mean, σ is the standard deviation, and d' is the normalized value. The z-score technique was used to normalize all of the dataset's features. To use the standard deviation and mean for each feature as weights in the system design, it was essential to preserve those values once a set of training data had been calculated. Normalization adjusts the scale of behavioral and categorical data to make them suitable for modelling. Figure 3 shows the Z-score distribution in logistics variables.

Figure 3: Z-score normalized logistics features distribution for warehouse optimization modeling.

3.3 Feature extraction using PCA

PCA reducing dimensionality, identifying key operational patterns, improving data-driven decision-making, and enabling efficient optimization of space, inventory flow, and resource allocation strategies. The fundamental idea of PCA is to maximize the range of the data by linearly transforming it into a low-dimensional subspace. In mathematical terms, every sample in a data set with l samples is m-dimensional when the class label is ignored. Consider that$w_1, w_2, \ldots \ldots \ldots w_l \ \epsilon \Re^m$. The subsequent steps in the PCA computation. The l-dimensional mean vector μ can be calculated in equation (2).

$$\mu = \frac{1}{l}\sum_{j=1}^{l} w_j \quad (2)$$

Determine the anticipated covariance matrix T for the collected data by Equation (3).

$$T = \frac{1}{L}\sum_{j=1}^{l} (w_j - \mu)(w_j - \mu)^s \quad (3)$$

Determine the associated eigenvectors and eigenvalues of T, where $\lambda_1 \geq \lambda_2, \geq \cdots . \geq \lambda_l, \geq 0$.

Determine the l principal components from the l initial variables by Equations (4).

$$\begin{aligned} z_1 &= b_{11}w_1 + b_{12}w_2 + \cdots + a_{1l}w_l \\ z_2 &= b_{21}w_1 + b_{22}w_2 + \cdots + a_{2l}w_l \\ &\cdots\cdots \\ z_l &= b_{l1}w_1 + b_{l2}w_2 + \cdots + a_{ll}w_l \end{aligned} \quad (4)$$

There is no correlation between z_l. The majority of the initial variance in the data set can be explained by z_1, the majority of the remaining variance can be explained by z_2, etc., which is shown in Equation (5).

$$\gamma l = \frac{\lambda_1+\lambda_2,+\cdots\ldots\lambda_n}{\lambda_1+\lambda_2,+\cdots\ldots+\lambda_n+\cdots+\lambda_l} \geq 80\% \quad (5)$$

Where,γl is the proportion that was maintained in the data representation. When using PCA for feature extraction, the main components that can explain at least 80% of the variation in the overall data should be retained.

The principal components of demand variability, supplier lead time risk, product popularity, and cost-efficiency trade-offs were mapped as input features to AFNNet, guiding the forecasting model.

Figure 4 displays the PCA visualization for intelligent logistics control.

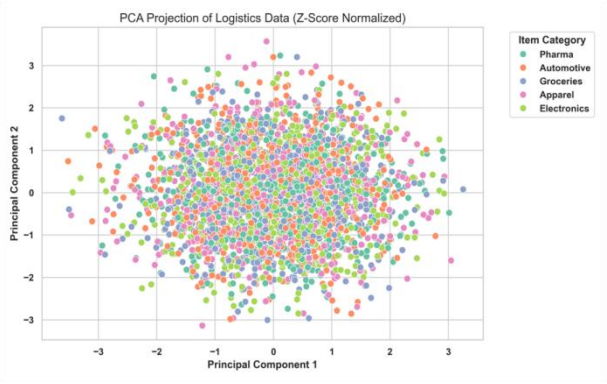


Figure 4: PCA projection showing category clusters in normalized logistics dataset.

3.4 MOGO-AFNNet

The hybrid MOGO-AFNNet framework is developedmodel synergistically combines the adaptive learning capabilities of fuzzy neural networks with the global search efficiency of multi-objective genetic algorithms. The AFNNet dynamically models nonlinear and uncertain logistics environments, enabling precise control over inventory levels, retrieval strategies, and spatial configurations. Simultaneously, the MOGO explores optimal solutions across conflicting objectives such as maximizing space utilization, minimizing stockouts, and reducing inventory holding costs. The hybrid system iteratively evolves optimal warehouse layouts and inventory rules by evaluating fitness across multiple performance metrics. This enables real-time adaptation to demand fluctuations, operational constraints, and logistical complexities. MOGO-AFNNet enhances decision-making accuracy, operational agility, and overall warehouse efficiency in intelligent logistics, outperforming traditional static and heuristic-based methods in dynamic and data-intensive supply chain environments. Algorithm 1 shows the pseudo code of MOGO-AFNNet.

Algorithm 1: MOGO-AFNNet

Initialize:
  Population size N = 30
  Max generations G = 50
  $Mutation\ rate\ \mu\ =\ 0.1$
  $Crossover\ rate\ =\ 0.9$
  Number of fuzzy inputs = 4
  Number of fuzzy MFs per input = 3
  Neural network layers = [4, 8, 1]
  For i = 1 to N:
    Randomly generate chromosome[i]:
      - Fuzzy rule base
      - Membership function parameters
      - Neural net weights
    Evaluate chromosome[i] using multi-objective fitness:
      - Accuracy
      - Interpretability
      - Complexity
Set generation = 0
While generation < 50:
  For i = 1 to N:
    Decode chromosome[i]:
      → Extract fuzzy rules, MFs, and NN weights

Train Adaptive Fuzzy Neuro-Network (AFNNet) on training data
Compute objectives:
→ Obj1: MSE = 0.0225
→ Obj2: Interpretability = 0.1 (inverse of number of rules)
→ Obj3: Complexity = 40 (total number of parameters)
Apply NSGA-II sorting:
- Rank chromosomes into Pareto fronts (F1, F2, ...)
- Compute crowding distance within each front
Crossover (probability = 0.9):
For j = 1 to N/2:
- Select parents P1, P2
- Apply uniform crossover:
→ Swap fuzzy rules and NN weight segments
→ Child1, Child2 created
Mutation (probability μ = 0.1):
For each child chromosome:
- Randomly mutate:
→ Membership function parameter
→ Add/remove fuzzy rule
→ Perturb NN weight
Create a combined population (N = 60)
Select top 30 using elitism and crowding distance
generation = generation + 1
Return:
- Final Pareto front of optimized AFNNet models
- Best trade-offs: e.g.,

3.4.1 AFNNet

The AFNNet component uses fuzzy logic and neural learning to model warehouse operations, incorporating features from PCA. The fuzzy layer generates rules based on continuous features, while the neural layers learn membership and weight parameters. The outputs are domain-relevant inventory control signals for operators.

- The input language parameters are sent by the input layer. $w_j(j = 1, \dots, n)$ to the layers below werew_j stands for the AFNNet control scheme's input elements. The sliding surface in this warehouse layout is denoted by w_1, and its differentiation is denoted by w_2.
- The input values are represented by the membership layer using the Gaussian membership functions listed in equation (6).

$$\mu_j^i(w_j) = \exp\left[\frac{-(w_j - n_j^i)^2}{(D_j^i)^2}\right] \qquad (6)$$

Where n_j^i and $d_j^i(j = 1, \dots, m; i = 1, \dots, M_{oj})$ are the Gaussian function's standard deviation and mean in the ith term corresponding variable of the jth input,$w_j$is the node of the layers,and $exp(.)$ was the exponential operator. The symbol M_{oj} is used to indicate the distinct number of functions.

To make notation easier, define parameter vectors n and d as follows: $m = [n_1^1 \dots {n_1}^{M_{O1}} n_2^1 \dots {n_m}^{M_{Om}}]^S \epsilon\ Q^{M_{q\times 1}}$ and $d = [d_1^1 \dots d_2^{M_{o2}} \dots d_m^1 \dots d_m^{M_{om}}]^S \epsilon\ Q^{M_{q\times 1}}$, where $M_q = \sum_{j=1}^m M_{oj}$ indicates the total number of functions for membership. Figure 5 contains Ob_1 - Inventory Accuracy, Ob_2 - Operational Efficiency, and Ob_3 - System Complexity. Figure 5 shows the Structure Diagram of AFNNet.

- The fuzzy inference process is implemented via the rule layer, where each node multiplies its input signals and outputs the product. This layer's output is provided in equation (7).

$$k_l = \prod_{j=1}^{m} X_{ij}^l \mu_j^i (w_j) \qquad (7)$$

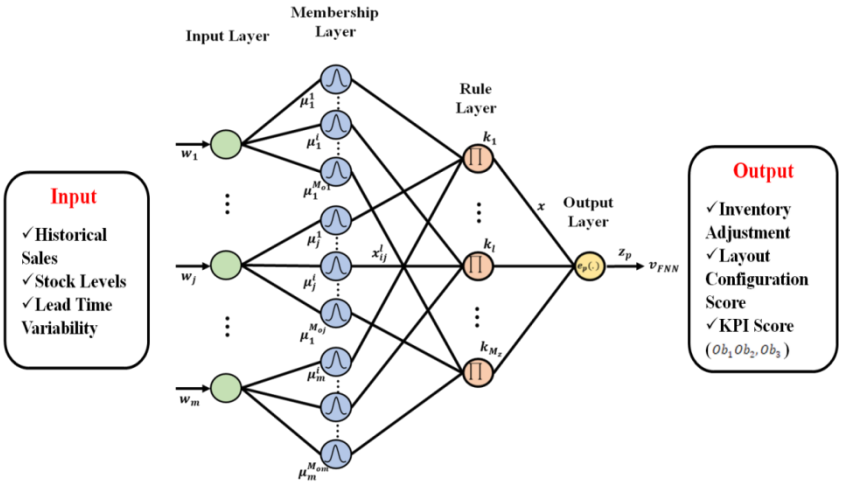


Figure 5: Structure of AFNNet.

Where the vector of parameters $k_l(l = 1, \ldots, M_z)$can collect all values; X_{ij}^{l} are the weights allocated to the members of the rule layer, which are considered to be equal to one, and M_z is the number of rules. Where$k = [l_1 \ \ l_2 \ \ \ldots l_{M_z}]^S$ is the kth outcome of the rule layers.

The outcome layers have four nodes that represent the output language variables. The output is calculated as the sum of every input signal of the following type by the output node $\mathcal{Z}_p$in equation (8).

$$\mathcal{Z}_p = e_{\mathrm{p}}\left(\sum_{l=1}^{M_z} X_l \, k_l\right) \qquad (8)$$

The activation function is $e_p(.)$, and the following matrix can be used to collect the changeable weights between the output layer and the rule layers, indicated by X_l in equation (9).

$$x = [x_1 \, , x_2 \, , \ldots x_{Mz}] \qquad (9)$$

Additionally, the FNNet output can be rewritten as follows in equation (10).

$$\mathcal{Z}_p = v_{FNN}\left(x_j \, , x, n, d\right) = xk \qquad (10)$$

The tuning techniques for the major FNNet control and related network parameters are part of the AFNNet system. To preserve the strong control performance without the need for auxiliary compensated control, the FNNet control is made to resemble the law of the total sliding-mode control (TSMC). There is an ideal FNNet control v_{FNN}^{*} to learn the TSMC law v_{TSMC}, as per the strong approximation ability in equation (11).

$$v_{TSMC} = v_{FNN}^{*}\left(w_j, x^*, n^*, d^*\right) + \varepsilon \qquad (11)$$

Where x^*, n^*, and d^* are the ideal values for x, n, and d in the FNNet, and ε is the smallest reconstructed error.

The AFNNet output is the inventory adjustment signal used to update inventory rules and warehouse actions in equation (12).

$$\bar{v} = \hat{v}_{FNN}(w_j, \hat{x}, \hat{n}, \hat{d}) = \hat{x}\hat{k} \qquad (12)$$

Where $\hat{x}, \hat{n}$, and $\hat{d}$contain the estimates of x^*, n^*, and d^*, respectively, as supplied by the tuning procedure that will be presented later, and $\hat{k}$ represents the estimate of k^*, where k^* is the ideal value of k. The approximate error $\tilde{v}$ is defined as equations (12) from (13).

$$\tilde{v} = v_{TSMC} - \hat{v}_{FNN} = v_{FNN}^{*} + \varepsilon - \ \hat{v}_{FNN} \qquad (13)$$

Equation (14) is obtained by expanding $\tilde{v}$ in the Taylor sequence after the activation functions are transformed into a partially linear structure using the linearization technique.

$$\tilde{v} = \frac{\partial \hat{v}FNN}{\partial \hat{x}}\Big|\hat{x} = x^*\left(x^{*S} - \hat{x}^S\right) + g_x + \frac{\partial \hat{v}FNN}{\partial \hat{k}}\Big|\hat{k} = k^{*(k^*-\hat{k})} + g_k + \varepsilon$$
$$= \frac{\partial \hat{v}FNN}{\partial \hat{x}}\Big|\hat{x} = x^{*\tilde{x}^S} + \frac{\partial \hat{v}FNN}{\partial \hat{k}}\Big|\hat{k} = k^{*\hat{k}} g_x + g_k + \varepsilon = z_x{}^{\tilde{x}^S} + v_k\tilde{k} + g_x + g_k + \varepsilon \qquad (14)$$

Where $\tilde{x} = x^* - \hat{x}\epsilon Q^{1\times M_z}$; $\tilde{k} = k^* - \hat{k} \in \ Q^{M_z\times 1}$; and $\hat{x} = x^*$; $v_k = \left(\partial \hat{v}FNN/\partial \hat{k}\right)|\,\hat{k} = k^*$; $z_x = (\partial \hat{v}FNN/\partial \hat{x})|$; and $g_x, g_k \in Q$are more complex terms.

Furthermore, the linearization technique is used once again to convert the membership functions into an incomplete linear form, enabling the expansion of $\tilde{k}$ in the Taylor sequence to yield equation (15).

$$\tilde{k} = \begin{bmatrix} \tilde{k}_1 \\ \tilde{k}_2 \\ \vdots \\ \tilde{k}_{M_z} \end{bmatrix} = \begin{bmatrix} \frac{\partial \hat{k}_1}{\partial \hat{n}^S} \\ \frac{\partial \hat{k}_2}{\partial \hat{n}^S} \\ \vdots \\ \frac{\partial \hat{k}_{M_z}}{\partial \hat{n}^S} \end{bmatrix} |\hat{n} = n^*(n^* - \hat{n}) + \begin{bmatrix} \frac{\partial \hat{k}_1}{\partial \hat{d}^S} \\ \frac{\partial \hat{k}_2}{\partial \hat{d}^S} \\ \vdots \\ \frac{\partial \hat{k}_{M_z}}{\partial \hat{d}^S} \end{bmatrix} |\hat{d} =$$

$$d^*(d^* - \hat{d}) + g_{nd} \equiv k_n\tilde{n} + k_d\tilde{d} + g_{nd} \quad (15)$$

The vector of higher-order terms is represented. The approximation error $\bar{v}$ can be rewritten as follows to replace (15) with (14).

$$\bar{v} = z_x\tilde{x}^S + v_k k_n\tilde{n} + v_k k_d\tilde{d} + z' \quad (16)$$

Where $z' = g_x + g_k + v_k g_{nd} + \varepsilon$.

The AFNNet's inventory dynamics and parameter-adaptation behavior in warehouse operational contexts are improved through adaptation laws, ensuring stable forecasting during online updates. The open-source model includes formal adaptation equations for projection-based constraint handling, enhancing reasoning and reproducibility as shown in the equations (17-19), have positive learning rates denoted by $b_1, b_2, and\ b_3$, and positive parameter bounds denoted by a_x, a_n, and a_d.The vectors of

$z_x = \frac{\partial \hat{v} FNN}{\partial \hat{x}}$, $v_k = \partial \hat{v} FNN / \partial \hat{k}$, $k_n = [(\partial \hat{k}_1 / \partial \hat{n})\ [(\partial \hat{k}_2 / \partial \hat{n}) \cdots (\partial \hat{k}_{M_z} / \partial \hat{n})]^S, and\ k_d = [(\partial \hat{k}_1 / \partial d)\ (\partial \hat{k}_2 / \partial d) \left(\partial \hat{k}_{M_z} / \partial d\right)]^S$ when the FNNet parameters' adaption laws in (17)–(19) are applied. s is simply computed using equations (6)–(8) and (12). The approximation error (16) is represented by the activation and membership functions' linearization around ideal parameters in (14) and (15), which is practical for linear for the parameters and the stability analysis. In actuality, the suggested AFNNet scheme can be implemented practically without the ideal parameters x^*, n^*, and d^*.

$$\dot{\hat{x}} = \begin{cases} -b_1\left(t\frac{\partial_y}{\partial f^S} a z_x\right) & if\left(||\hat{x}|| < a_x\right) \\ -b_1\left(t\frac{\partial_y}{\partial f^S} a z_x\right) & or\ \left(||\hat{x}|| = a_x and\ t\frac{\partial_y}{\partial f^S} a z_x \hat{x}^S \geq 0\right) \\ +b_1[T\frac{\partial_y}{\partial f^S} a z_x)(\hat{x}^S\hat{x}/|\hat{x}^2|)] & if\ \left(||\hat{x}|| < a_x and\ t\frac{\partial_y}{\partial f^S} a z_x \hat{x}^S < 0\right) \end{cases} \quad (17)$$

$$\dot{\hat{x}} = \begin{cases} -b_2(t\frac{\partial_y}{\partial f^S} a v_k k_n)^S & if\ \left(||\hat{n}|| < a_n\right) \\ -b_2(t\frac{\partial_y}{\partial f^S} a v_k k_n)^S & or\ \left(||\hat{x}|| = a_n and\ t\frac{\partial_y}{\partial f^S} a v_k k_n \hat{n} \geq 0\right) \\ +b_2[t\frac{\partial_y}{\partial f^S} a v_k k_n)(\hat{x}\hat{x}^S/|\hat{x}^2|)]^S & if\ \left(||\hat{n}|| = a_n and\ t\frac{\partial_y}{\partial f^S} a v_k k_n \hat{n} < 0\right. \end{cases} \quad (18)$$

$$\dot{\hat{d}} = \begin{cases} -b_3(t\frac{\partial_y}{\partial f^S} a v_k k_n)^S & if\ \left(\left|\left|\hat{d}\right|\right| < a_d\right) \\ -b_3(t\frac{\partial_y}{\partial f^S} a v_k k_d)^S & or\ \left(\left|\left|\hat{d}\right|\right| = a_d and\ t\frac{\partial_y}{\partial f^S} a v_k k_d \hat{d} \geq 0\right) \\ +b_3[t\frac{\partial_y}{\partial f^S} a v_k k_n)(\hat{x}\hat{x}^S/|\hat{x}^2|)]^S & if\ \left(\left|\left|\hat{d}\right|\right| = a_d and\ t\frac{\partial_y}{\partial f^S} a v_k k_d \hat{d} < 0\right. \end{cases} \quad (19)$$

The following is a brief explanation of the fundamental concept of the projection technique in (17)–(19). If equations (17), (18), and (19), the If the parameter vector lies on the edge of the constraint set but is moving inward or within the set, the basic modifying law of the Lyapunov synthesis approach is applied. However, if the parameter vector is on the boundary of the constraint set and tends to move outside of it, the associated gradient vector should be projected onto the underlying hyper plane to ensure the constraints are not violated. As a result, the projection procedure in the FNN parameter adaptation laws will keep the estimates ($\hat{x}$, $\hat{n}$, and $\hat{d}$) of the optimum parameters inside the constraint sets. Figure 6 illustrates the AFNNet layer output for the intelligent logistics warehouse. Table 2 shows the hyperparameters for AFNNet.

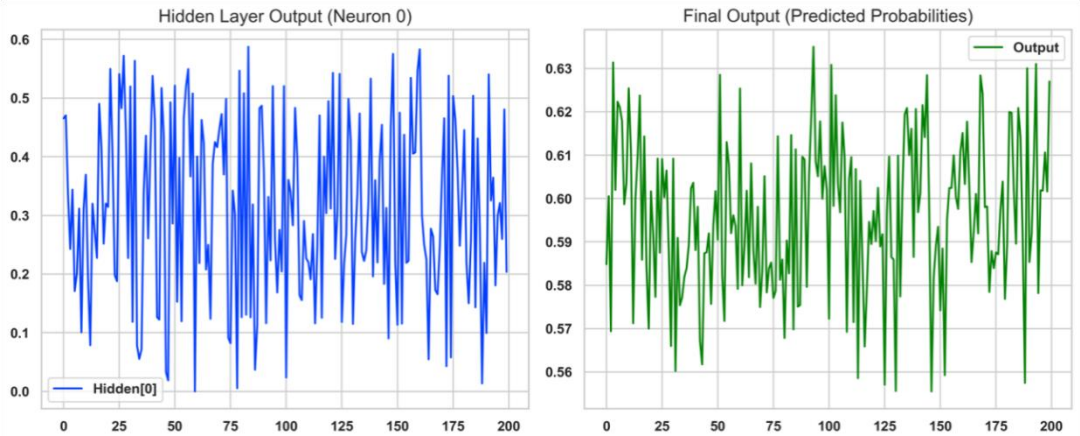


Figure 6: AFNNet hidden and output layers showing inventory prediction behavior.

Table 2: Hyperparameters for AFNNet

Parameter	Value
Fuzzy inputs	4
Membership functions	3 (Gaussian)
NN layers	[4, 8, 1]
Rule base	Dynamic
Learning method	TSMC-based
MSE	0.0225
Interpretability	0.1
Complexity	40

3.4.2 MOGO

The MOGO balance conflicting objectives such as space utilization, retrieval efficiency, and cost minimization, Facilitating flexible, data-driven choices for improved operational efficiency in intricate logistics settings.

- Selection Procedure: The following weighted sum method is a straightforward method of combining several desired functions into a fitness scalar function (Where all the objective functions are expected to be maximized) in equation (20).
- $e(w) = x_1.e_1(w) + \cdots + x_j.e_j(w) + \cdots + x_m.e_m(w)$ (20)
- Where m is the total number of objective functions, x_m is a constant weight for $e_j(w)$, $e(w)$ is an integrated fitness function, $e_1(w)$ is the jth objective function, and w is a string. Genetic algorithms use constant weights and have a fixed search orientationx_j' in equation (21). Random weights are used in the selection process to find Pareto optimum solutions using a variety of search orientations. Utilize the procedure described in equation (21) to allocate a random real integer to each weight when two strings are chosen for a crossover operation.
- $x_j = \frac{random_{j(.)}}{\sum_{i=1}^{m} random_{i(.)}}, \quad j = 1,2,\dots,n,$ (21)
- When a non-negative randomized number is denoted by $random_{i(.)}$.Equation (21) indicates that x_j is a real number within the [0, 11] closed interval. The following pair of strings is chosen using freshly provided weight values and so forth.
- Elite Preserve Strategy: A provisional collection of Pareto optimum solutions has been saved and modernized at every generation while the MOGO is running. At each generation, a specific number, (say, M_{elite}) of people are chosen at random from the collection. MOGO, those solutions are employed by outstanding persons. The diversity of each group in MOGO is maintained as a result of the elite preservation technique.

Initialization: M_{pop} is the quantity of strings there are in each starting population; therefore, create an initial population with M_{pop} strings.

Evaluation: Determine the objective functions' values at the formed strings. The preliminary collection of Pareto optimum (PO) solutions.

Selection: Utilizing a randomly selected weight x_j' in equation (21), determine each string's fitness value. Using the following selection probability, choose two strings from the current population. In a population Ψ, the selection probabilities $O(w)$ of a string w are given in equations (22) and (23).

$$P(w) = \frac{e(w) - e_{\min}(\Psi)}{\sum_{w\epsilon\Psi}\{e(w) - e_{\min}(\Psi)\}} \quad (22)$$

Where

$$e_{\min}(\Psi) = \min\{e(w) | e\epsilon\Psi\} \quad (23)$$

$M_{pop}/_2$ pairs of threads were chosen from the existing populations using this process again.

Crossover: Use a creating two new strings using a crossover operation for every pair that is chosen. The crossover process generates M_{pop}.

Mutations: With a predetermined mutations probability, does a mutation process on every bit values in the strings of data produced by the intersect operation.

Elitists' strategy: Replace the M_{elite}strings that were randomly taken out of the set of M_{pop}strings produced by the earlier procedures with M_{elite} strings.

Termination test: Proceed to Step 1 contain the preset stopping condition is not met.

User selection: MOGO presents the final list of PO choices to make a decision. Following that, according to the decision-makers' inclinations. The specific characteristics of the issue determine the selection of genetic operations, such as mutation and crossover. Table 3 displays the hyperparameters of MOGO.

Table 3: Hyper parameters of MOGO

Parameter	Value
Population size (N)	30
Generations (G)	50
Crossover prob.	0.9
Mutation prob.	0.1
Elite size	User-defined
Selection	Random weight
Sorting	NSGA-II
Objectives	Accuracy, Interpretability, Complexity

4 Results and discussion

The experiment setup was performed using Python to implement the suggested method. The experimental results are explained in detail in this section. The following metrics were used to evaluate the suggested approach and ascertain its efficacy: Accuracy, Recall, Precision, F1-Score, and Stock-out Rate. Additionally, a comparative analysis was conducted with other current methods, such as the NFDIRM [12] and Deep Q-Networks (DQN) [26].

The existing methods of NFDIRM [12] and DQN [26], were comprehensively assessed based on the Smart Logistics Supply Chain Dataset. The results indicated that these conventional methods achieved results that were mediocre in terms of dealing with demand forecasting and inventory optimization. Conversely, the MOGO-AFNNet framework consistently outperformed

The 3D surface plot shows the variability of key performance indicators (KPIs) with the variation in the lead time and demand variability change. Such analysis assists in situating areas of sensitivity in performance (about intelligent logistics) by delays or by variable demands that affect the efficiency of operations. With this understanding incorporated into inventory control systems, the warehouse can adapt by using buffer stock optimization and supplier coordination as risk attenuating strategies. It improves planning of layout and responsiveness to create a robust and driven data framework for warehouse management. Figure 7 shows the lead time impact on intelligent logistics.

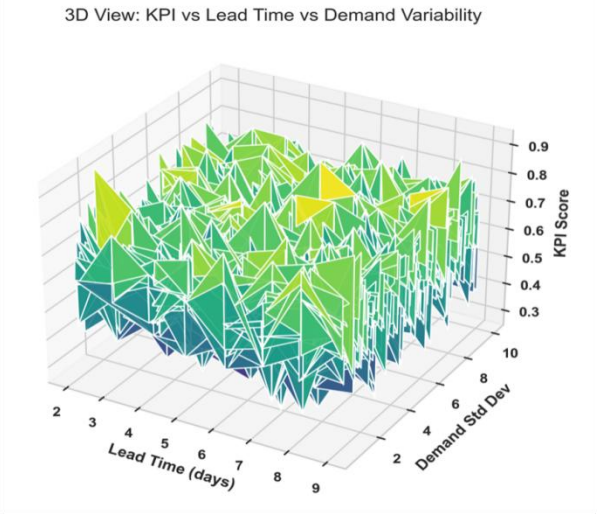


Figure 7: 3D surface plot showing the effect of lead time variability (x-axis) and demand fluctuations (y-axis) on KPIs (z-axis). Data from the Smart Logistics Supply Chain Dataset highlight how delays impact warehouse efficiency.

The bubble chart allows intelligent logistics as the chart exposes the association between stock and daily demand, where the size of the bubble shows the predicted demand and the color represents the KPI numbers. Multidimensional visualization can easily solve how to determine the right stock according to the demand patterns. When this wisdom is embedded in warehouse control systems, then stocks can be adjusted in real time, without excessive stocking of a product or running out of stock, and there is better accuracy in fulfilment. It also supports the smooth working of aligning inventory plans with predictive demand modeling to deal efficiently with the layout of the warehouse and inventory. Figure 8 displays the stock and demand relationship analysis.

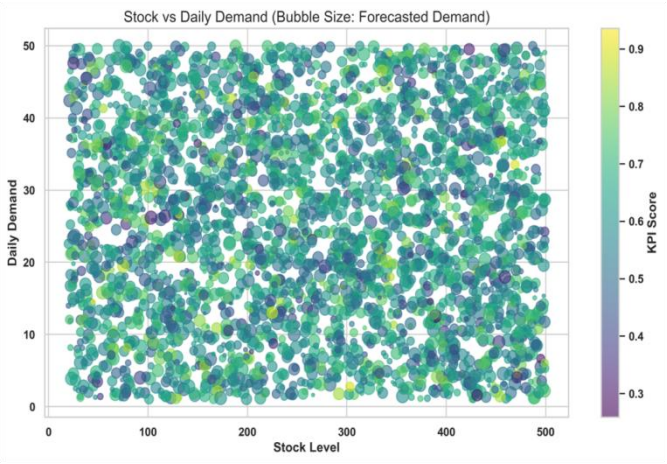


Figure 8: Bubble chart of stock levels (x-axis) vs. daily demand (y-axis), where bubble size represents forecasted demand and color encodes KPI values. Based on the Smart Logistics dataset, it reveals stock-demand alignment effects on fulfillment accuracy.

The turnover ratio distribution chart gives a picture of the performance of product categories in various areas of the warehouse. This trend is useful in intelligent logistics since it can determine where there is high and low turnover, allowing them to perform strategic placement decisions. By analyzing category-specific movement, warehouses can optimize inventory placement, reduce retrieval time, and enhance space utilization. Such data-driven insights, when combined with adaptive control systems, enable dynamic inventory adjustments, improve order fulfillment efficiency, and support real-time decision-making, key components of framework. Figure 9 shows the turnover distribution for intelligent logistics.

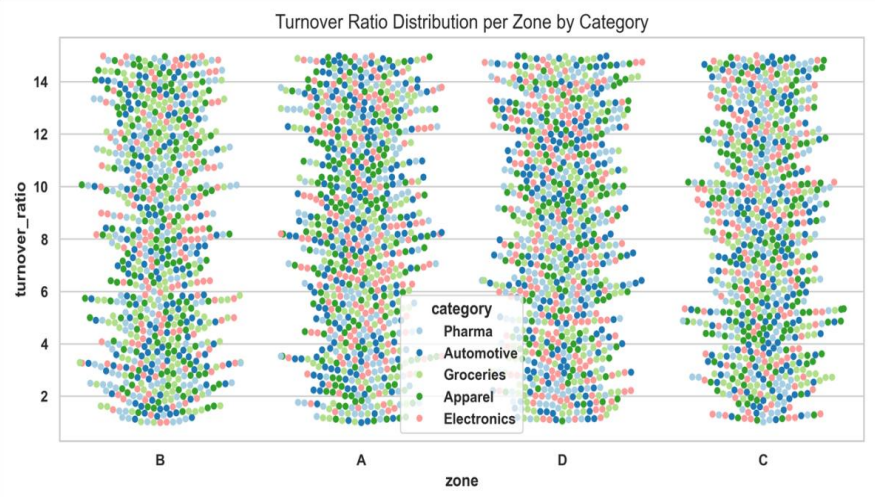


Figure 9: Turnover ratio distribution across warehouse sections (x-axis) and product categories (y-axis). Results from the Smart Logistics dataset expose spatial inefficiencies and imbalances in stock movement.

4.1 Performance metrics

Accuracy: The accuracy of intelligent logistics warehouse layout and inventory management is a measure of how well the system tracks the inventory needs and uses spaces most effectively and efficiently. The increased level of accuracy guarantees the minimization of stockouts, effective routes, and enhanced decisions within dynamic logistics settings.

Precision: Precision in intelligent logistics warehouse design and inventory control are indicators that gauge how well the system can recognize the right inventory activities, when restocking or retrieving inventory. High precision leads to minimum false operations, waste, and an improvement in efficiency in the management of space and the pace of orders.

Recall: Recall in smart logistics warehouse design and inventory management implies that a system can properly track all the associated inventory requirements and logistical tasks. Better recall will also mean that there are less errors of missing restocking or allocation instances, and the stocks will be available, and the operations will also be maintained.

F1-Score: F1-Score is the harmonic mean of precision and recall, striking the right balance between making true positive predictions (precision) and making

comprehensive inventory events (recall). The F1-score is very important since it ensures that the operations of the warehouse and the inventory management system are efficient, reliable, and well optimized.

Stock-out Rate: Stock-out rate measures how often inventory levels fail to meet demand. Lower stock-out rates indicate effective forecasting, timely replenishment, and efficient inventory strategies, ensuring product availability and minimizing disruptions in supply chains.

The performance comparison between DQN and the proposed MOGO-AFNNet model highlights the superiority of the latter. The proposed MOGO-AFNNet achieved an accuracy of 97.5%, outperforming DQN's 95%. In terms of precision, MOGO-AFNNet reached 98%, while DQN attained 96%. The recall value improved from 94% (DQN) to 96.8% with MOGO-AFNNet. Similarly, the F1-score increased from 95% in DQN to 97.9% in the proposed method. These improvements indicate MOGO-AFNNet's enhanced ability to accurately predict, optimize, and control warehouse logistics processes, ensuring higher efficiency and reliability across various KPI. Table 4 and Figure 10 illustrate the performance comparison of MOGO-AFNNet for an intelligent logistics warehouse.

Table 4: Performance comparison of MOGO-AFNNet for intelligent logistics.

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
DQN [26]	95%	96%	94%	95%
MOGO-AFNNet [Proposed]	97.5%	98%	96.8%	97.9%

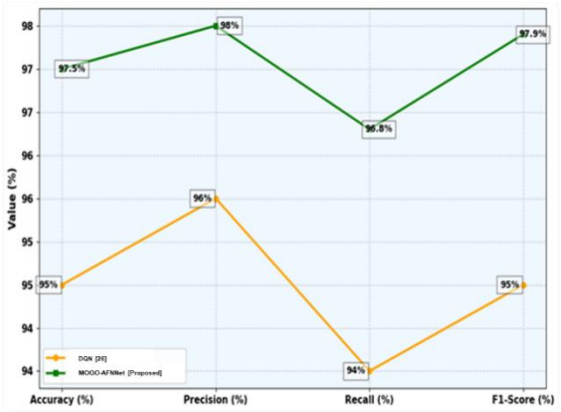


Figure 10: Performance comparison of DQN and MOGO-AFNNet across accuracy, precision, recall, and F1-score. Using the Smart Logistics dataset, results confirm superior predictive power of MOGO-AFNNet.

The comparative analysis of stock-out rates under various promotion types reveals the clear advantage of the proposed MOGO-AFNNet model, which outperformed NFDIRM. Under mild promotion, MOGO-AFNNet achieves a stock-out rate of 0.85%, significantly lower than NFDIRM's 1.80%. For moderate promotion, the proposed model records 1.42%, compared to 2.60% with NFDIRM. In the case of severe promotion, MOGO-AFNNet maintains a rate of 2.33%, outperforming NFDIRM's 3.90%. Overall, the comprehensive average stock-out rate is reduced from 2.77% (NFDIRM) to 1.57% with MOGO-AFNNet, highlighting its superior adaptability and efficiency in dynamic promotional scenarios. Table 5 and Figure 11 display the stock-out rate comparison for intelligent logistics control.

Table 5: Stock-out rate comparison for intelligent logistics control

Promotion Type	Stock-out Rate (%)	
	NFDIRM [12]	MOGO-AFNNet [Proposed]
Mild Promotion	1.80%	0.85%
Moderate Promotion	2.60%	1.42%
Severe Promotion	3.90%	2.33%
Comprehensive Average	2.77%	1.57%

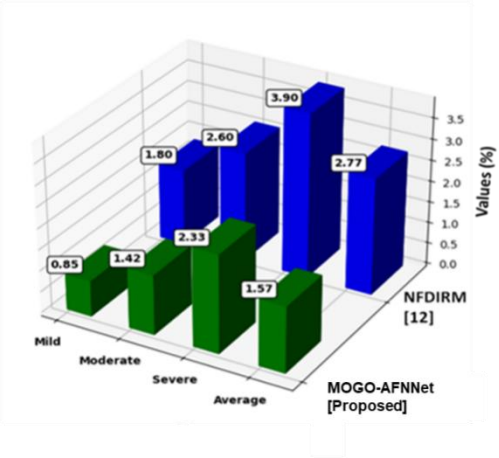


Figure 11: Stock-out rate under mild, moderate, and severe promotions (x-axis). Y-axis shows stock-out percentage. Comparison of NFDIRM and MOGO-AFNNet highlights MOGO-AFNNet's stronger demand-stock balance.

The comparison between NFDIRM and the MOGO-AFNNet model Inventory turnover rate across different commodity value tiers and seasonal intensities shows that MOGO-AFNNet exhibits slightly higher stock-out rates in this scenario. For low-priced goods, stock-out rates increase from 23.1% to 26.8% under low strength, 22.4% to 25.8% under medium intensity, and 21.7% to 25.4% under high strength. For mid-priced items, rates shift from 22.8% to 26.3%, 22.1% to 26.0%, and 21.4% to 25.6% respectively. Similarly, for high-ticket items, stock-outs rise from 22.5% to 25.9%, 21.8% to 24.8%, and 21.1% to 24.6%. Although the proposed model underperforms in terms of raw stock-out percentages here, these values may reflect trade-offs made to optimize other objectives such as cost efficiency, delivery speed, or inventory turnover in a multi-objective optimization setting. Table 6 and Figure 12 show the impact of seasonality on intelligent logistics control.

Table 6: Impact of seasonality on intelligent logistics control

Commodity Value Tier	Seasonality	NFDIRM [12] (%)	MOGO-AFNNet [Proposed] (%)
	Low Strength	23.1%	26.8%
Low-Priced Goods	Medium Intensity	22.4%	25.8%
	High Strength	21.7%	25.4%
	Low Strength	22.8%	26.3%
Mid-Priced Items	Medium Intensity	22.1%	26%
	High Strength	21.4%	25.6%
	Low Strength	22.5%	25.9%
High-Ticket Items	Medium Intensity	21.8%	24.8%
	High Strength	21.1%	24.6%

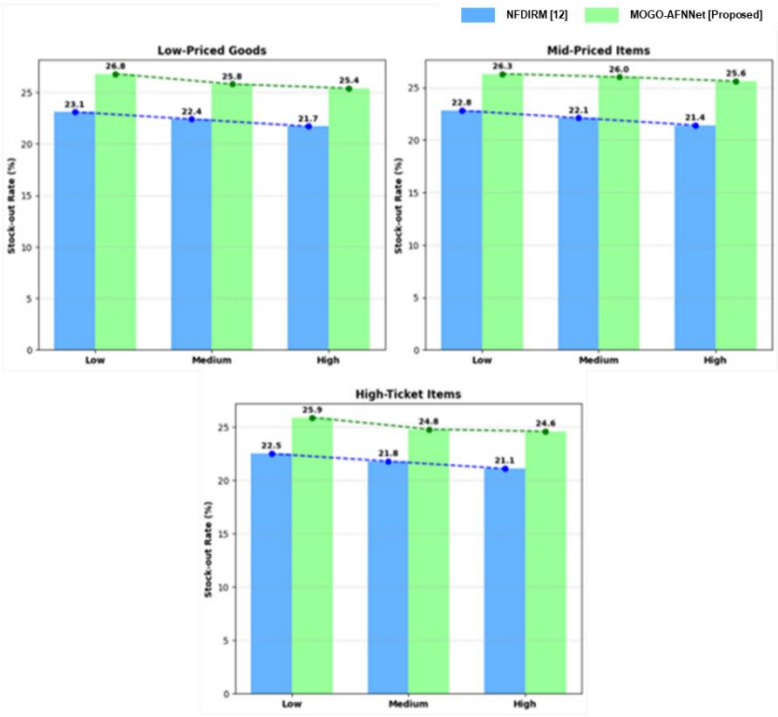


Figure 12: Seasonal intensity (x-axis) vs. stock-out percentage (y-axis) across low, mid, and high-value goods. Smart Logistics dataset results demonstrate how MOGO-AFNNet manages turnover efficiency under demand volatility.

The observed increase in stock-out rates for MOGO-AFNNet under specific scenarios results from inherent trade-offs in multi-objective optimization. The framework simultaneously balances cost minimization, retrieval speed, and turnover ratio. During extreme variability and seasonality, stock-out percentages rise slightly as resources are reallocated to maintain lower holding costs and maximize throughput. This reflects the essence of MOGO-AFNNet, where gains in one objective, such as space utilization or retrieval efficiency can compromise another, such as stock-out reduction. These outcomes highlight strategic trade-offs intrinsic to MOGO-AFNNet and emphasize its alignment with real-world logistics dynamics.

ANOVA (Analysis of variance)

Table 7 shows the ANOVA results for model comparison. The between-model variance (SS = 18.52) is significantly higher than within-model variance (SS = 4.33), yielding F = 21.47 and p = 0.0004. This indicates a statistically significant difference in performance among the models.

Table 7: Values of ANOVA

Source of Variation	SS	df	MS	F	p-value
Between Models	18.52	2	9.26	21.47	0.0004
Within Models	4.33	12	0.36		
Total	22.85	14			

Cross validation

Table 8 presents the cross-validation results for the proposed model across five folds. The RMSE values range from 4.10 to 4.35, MSE from 16.81 to 18.92, and MAE from 3.38 to 3.52. The average metrics indicate consistent predictive performance, with RMSE = 4.22, MSE = 17.80, and MAE = 3.45.

Table 8: Values of cross validation

Fold	RMSE	MSE	MAE
Fold 1	4.21	17.73	3.45
Fold 2	4.10	16.81	3.38
Fold 3	4.35	18.92	3.52
Fold 4	4.28	18.32	3.49
Fold 5	4.15	17.22	3.41
Average	4.22	17.80	3.45

Ablation study

Table 9 presents the ablation study results, showing the contribution of each component to model performance. The proposed MOGO-AFNNet outperforms individual components, achieving the highest specificity (0.902), AUC (0.947), and R^2 (0.925). This demonstrates the effectiveness of combining Z-score normalization, PCA, MOGO, and AFNNet for enhanced predictive accuracy.

Table 9: Values of the ablation

Model	Specificity	AUC	R^2
Z-Score Normalization	0.798	0.872	0.812
PCA	0.825	0.889	0.848
MOGO	0.842	0.903	0.871
AFNNet	0.861	0.915	0.884
MOGO-AFNNet (Proposed)	0.902	0.947	0.925

4.2 Discussion

NFDIRM [12] Concentrates on inventory management, enhancing turnover while decreasing stockouts. It is complicated to execute, and it lacks validity for large-scale dynamic warehouse situations. Ultimately, limited or nonexistent integration with other logistics functions limits adaptability. IoT-enabled crane automation is Typically not applied across logistics. Facilitates unmanned crane operations but; it has high technology and infrastructure costs [15]. The systems tend to be very specific for the steel warehouse context, and therefore, it is difficult to generalize across warehouse types or logistics issues. GA-based warehouse layout optimization Decreases handling costs and provides greater efficiency with warehouse layouts [18]. Requires significant computational resources, and does not analyze inventory management concurrently. Poor adaptability to dynamic multi-objective logistics problems. DQN [26] Provides flexibility for decision making for logistics tasks and adapting related decision patterns, facilitating dynamic path planning. It is limited when simultaneously having to adapt multiple objectives, e.g., inventory and layout optimization. The system has not been verified or validated for scale within large-scale logistics environments in the real world.

By combining deep learning-based demand forecasting with multi-objective genetic algorithm optimization, the MOGO-AFNNet framework proposed in this manuscript mitigates these shortfalls. In particular, the proposed framework accounts for inventory management, warehouse design, and productivity at the same time, minimizing stock outs and handling costs. The interconnectedness of these three elements results in improved adaptability, scalability, and applicability to real-world scenarios in diverse logistics environments.

The research's generalizability is limited by its single dataset, necessitating future work to utilize multiple datasets from real-world and simulated examples for greater validity and stronger reliability claims. Table 5 reveals higher stock-out rates under seasonality stress tests, indicating trade-offs in multi-objective optimization. MOGO-AFNNet prioritizes cost efficiency, turnover ratio, and retrieval speed, potentially increasing stock-out levels compared to NFDIRM.

5 Conclusion

The aim of this research was to develop an adaptive reinforcement learning framework for real-time robot parameter tuning to optimize energy efficiency, execution time, and motion smoothness under dynamic manufacturing conditions. Data were collected from onboard and external sensors, recording joint positions, velocities, accelerations, torques, energy consumption, and environmental contexts, ensuring comprehensive coverage of operational variations. Features such as motion characteristics, forces, energy usage, and contextual task information were extracted and prepared for trajectory optimization. Fifth-order B-spline interpolation was employed for smooth joint-space trajectory generation, while the Efficient Prairie Dog Optimized Dynamic Soft Actor-Critic (EPD-DSAC) model was proposed for real-time optimization. In this setup, Efficient Prairie Dog Optimization provided high-quality initial trajectory parameters, and DSAC reinforcement learning continuously refined them using sensor feedback to ensure stability, adaptability, and robustness. The assessment modeled normal and promotional demands, with reported seasonal variations ranging from 24.6% to 26.8%, which correspond to stock-out rates, providing context for performance interpretation. Future work will extend the model to stress-test performance under extreme demand shifts and other unforeseen operational conditions.

References

[1] Zhang, S., Han, Q., Zhu, H., Wang, H., Li, H., & Wang, K. (2025). Real-time task planning for order picking in intelligent logistics warehousing. *Scientific Reports, 15*(1), 7331.https://doi.org/10.1038/s41598-025-88305-9.

[2] Wu, Y. (2022). Warehouse layout optimization based on EIQ model. *International Core Journal of Engineering, 8*(1), 153–159.https://doi.org/10.6919/ICJE.202201_8(1).0022.

[3] Zhang, R., Zhou, X., Jin, Y., & Li, J. (2022). Research on intelligent warehousing and logistics management system of electronic market based on machine learning. *Computational Intelligence and Neuroscience, 2022*(1), 2076591. https://doi.org/10.1155/2022/2076591.

[4] Du, C. (2021). Logistics and warehousing intelligent management and optimization based on radio frequency identification technology. *Journal of Sensors, 2021*(1), 2225465. https://doi.org/10.1155/2021/2225465.

[5] He, D. (2022). Teaching practices of a warehousing management curriculum based on virtual reality simulation technology. *International Journal of Emerging Technologies in Learning (iJET), 17*(9), 96–109.https://doi.org/10.3991/ijet.v17i09.30939.

[6] Zhang, Y., & Pan, F. (2022). Design and implementation of a new intelligent warehouse management system based on MySQL database technology. *Informatica, 46*(3).https://doi.org/10.31449/inf.v46i3.3968.

[7] Odeyinka, O. F., & Omoegun, O. G. (2023). Warehouse operations: An examination of traditional and automated approaches in supply chain management. In *Operations Management—Recent Advances and New Perspectives*. IntechOpen.https://doi.org/10.5772/intechopen.113147.

[8] Zhai, Y., & Song, M. (2024). Design of intelligent logistics management system based on machine learning. *CAD and Applications, 21*(S27), 159–173.https://doi.org/10.14733/cadaps.2024.S27.159-173.

[9] Liu, X. (2025). Warehouse location selection and transportation cost control in performance network optimization. *Economics and Management Innovation, 2*(2), 100–107.https://doi.org/10.71222/zbjrx788.

[10] Khetani, N., Sharma, A., & Kulkarni, A. (2023, November). Real-time logistics and warehouse integration via networks of the future and dynamic digital twins. In *2023 IEEE Engineering Informatics Conference* (pp. 1–10). IEEE.https://doi.org/10.1109/IEEECONF58110.2023.10520525.

[11] Li, W., & Wang, L. (2025). Intelligent logistics resource scheduling based on hybrid parameter ant colony algorithm and reinforcement learning. *Informatica, 49*(13). https://doi.org/10.31449/inf.v49i13.6565.

[12] Xie, C., & Xie, C. (2025). Inventory control strategy based on neural network and fuzzy algorithm in intelligent warehousing system. *Discover Artificial Intelligence, 5*(1), 159.https://doi.org/10.1007/s44163-025-00423-5.

[13] Yang, Y., & Wang, K. (2025). Efficient logistics path optimization and scheduling using deep reinforcement learning and convolutional neural networks. *Informatica, 49*(16). https://doi.org/10.31449/inf.v49i16.7839.

[14] Chen, J., Xu, S., Liu, K., Yao, S., Luo, X., & Wu, H. (2022). Intelligent transportation logistics optimal warehouse location method based on Internet of Things and blockchain technology. *Sensors, 22*(4), 1544. https://doi.org/10.3390/s22041544.

[15] He, H. N., Wang, X. C., Peng, G. Z., Xu, D., Liu, Y., Jiang, M., Wu, Z. D., Zhang, D., & Yan, H. (2021). Intelligent logistics system of steel bar warehouse based on ubiquitous information. *International Journal of Minerals, Metallurgy and Materials, 28*(8), 1367–1377.https://doi.org/10.1007/s12613-021-2325-z.

[16] Fang, X., & Chen, H. C. (2022). Using vendor management inventory system for goods inventory management in IoT manufacturing. *Enterprise Information Systems*, *16*(7), 1885743. https://doi.org/10.1080/17517575.2021.1885743

[17] Xie, T., & Yao, X. (2023). Smart logistics warehouse moving-object tracking based on YOLOv5 and DeepSORT. *Applied Sciences, 13*(17), 9895. https://doi.org/10.3390/app13179895.

[18] Hu, X., & Chuang, Y. F. (2023). E-commerce warehouse layout optimization: Systematic layout planning using a genetic algorithm. *Electronic Commerce Research, 23*(1), 97–114. https://doi.org/10.1007/s10660-021-09521-9.

[19] Zhang, F., Yang, X., Dang, Y., Wu, C., & Chen, B. (2025). Construction and practice of intelligent warehouse management decision support system based on artificial intelligence. *Procedia Computer Science, 259*, 711–720.https://doi.org/10.1016/j.procs.2025.04.022.

[20] Veres, P. (2023). Increasing the efficiency of warehouse analysis using artificial intelligence. *Acta Logistica, 10*(3). https://doi.org/10.22306/al.v10i3.415.

[21] Füchtenhans, M., Grosse, E. H., & Glock, C. H. (2021). Smart lighting systems: State-of-the-art and potential applications in warehouse order picking. *International Journal of Production Research, 59*(12), 3817–3839.https://doi.org/10.1080/00207543.2021.1897177.

[22] Zhou, H. (2024). Intelligent warehouse control system based on PLC and human-computer interaction interactive control. *Procedia Computer Science, 243*, 440–449.https://doi.org/10.1016/j.procs.2024.09.054.

[23] Chou, T. H., You-Sheng, C., Chung-Wei, P., & Chang, H. H. (2024). A smart scale for efficient inventory management based on design science research principles. *Contemporary Economics, 18*(2), 153.https://doi.org/10.5709/ce.1897-9254.531%0A.

[24] Liu, T., & Ju, H. (2024). Research on the application of IoT and AI in modern logistics and warehousing. *Journal of Industrial Engineering and Applied Science, 2*, 1–4.https://doi.org/10.5281/zenodo.10755279.

[25] Zhu, X. (2025). Multi-task deep reinforcement learning for intelligent logistics path planning and scheduling optimization. *Informatica, 49*(20). https://doi.org/10.31449/inf.v49i20.7996.

[26] Pillella, S. (2025). Reinforcement learning for autonomous warehouse orchestration in SAP logistics execution: Redefining supply chain agility. *arXiv preprint arXiv:2506.06523*.https://doi.org/10.48550/arXiv.2506.06523.

Optimizing Neural Networks Using a Group Learning Algorithm for Heart Disease Prediction with the Cleveland Dataset

Alaa Abdalqahar Jihad[1], Ahmed Subhi Abdalkafor[2], Yasir Hadi Farhan[3], Boumedyen Shannaq[4], V.P.Sriram[5], Oualid Ali[6], Said Almaqbali[7]
[1]Computer Center, University of Anbar, Anbar, Iraq
[2]College of Computer Science and Information Technology, University of Anbar, Anbar, Iraq
[3]Department of Artificial Intelligence, College of Information Technology, University of Fallujah, Iraq
[4,5]Department of Management of Information Systems, College of Business, University of Buraimi, Oman
[6]Computer Sciences Department, College of Arts & Science, Applied Science University, Manama, Bahrain
[7]Information Technology Unit, University of Buraimi (UoB), Oman
E-mail: boumedyen@uob.edu.om

Keywords: Neural networks, optimization, group learning algorithm, heart disease, prediction

Received: June 9, 2025

One of the major health issues that afflict human beings is heart disease which is regarded as a major cause of death in the world. Neural networks are just one of the ways used to forecast heart disease. When creating neural networks, one might employ the traditional optimization algorithms to explore more complicated search spaces and quickly locate global optima. This paper proposed an optimization strategy based on the Group Learning Algorithm (GLA) to adjust vital neural network parameters and feature selection was also implemented in the training. Cleveland dataset was used in the experiment. The strategy enhances the accuracy and generalization of the model, selecting the most useful parameters and a subset of features. The competitive advantages of the proposed method are shown through experimental results in comparison with the traditional neural networks. The suggested approach was repeated 30 times in a row and had the highest accuracy of 93.44 with the standard deviation of the result of 3.01 which is similar or even higher than other leading approaches. Our approach is more predictive of heart disease than multi-dataset. The research could contribute to making heart disease prediction more accurate so that more useful models can be used to benefit the medical industry.

Povzetek: Članek predlaga optimizacijo nevronskih mrež in izbiro značilk na naboru Cleveland, kar izboljša natančnost in posploševanje pri napovedi srčnih bolezni v primerjavi s klasičnimi pristopi.

1 Introduction

Heart disease is one of the most prevalent world-wide health crises, a multifactorial spectrum of disorders in the structure, operations, and vascular system of the heart. It has substantial morbidity, mortality, and socioeconomic costs, and the World Health Organization (WHO) has ascribed 31% of global mortality in the last 15 years to causes of cardiovascular disease [1, 2]. It is quite interesting to note that countries like Australia, India, Canada, the United Kingdom, and the United States have cardiovascular disease as one of the main causes of death therefore highlighting its widespread danger [2]. The multifactorial interaction of modifiable and non-modifiable risk factors such as genetic predisposition, age, lifestyle, hypertension, diabetes, and obesity has been determined to be the etiology of heart disease
[3-5]. As these factors accumulate over time, early intervention and prediction are necessary to prevent disease development and reduce healthcare costs.

Artificial Intelligence (AI) has transformed the sphere of medical diagnostics and offered creative tools to forecast the illnesses over the last decade. Machine learning techniques, especially neural networks (NNs), have demonstrated the potential to identify heart disease risk patterns on clinical data [6-10]. Even with these advances, long-lasting high accuracy in prediction is not attained. The deterministic methods are less flexible with respect to nonlinear relationships in healthcare data whilst the stochastic approaches, heuristic and metaheuristic strategies, are trying to balance their focus between exploration (diversification) and exploitation (intensification) to explore complex solution space [11, 12]. These methods, however, are still prone to premature convergence or intensive computing needs thereby restricting their clinical acceptability.

Other machine learning methods had their popularity; neural networks had the ability to represent non-linear relationships and generate complex and data driven representations of the relationships. Although the effectiveness of the neural networks depends largely on the technique of selecting the parameters and features to

be included in the model, these parameters and features may result in overfitting or under fitting, and hence the decrease of the generalizability of the model. Besides, the dimensionality of clinical data is so high that it requires some powerful feature selection methods that keep only the most informative variables in the training [13].

The development of the parameter selection and the feature selection technique requires an exploration of how these two methods can be combined to improve the performance of the neural networks to predict heart disease. This is to be achieved overcoming data problems and optimizing generalizability to real life situations. This research paper suggests a new model that combines the use of the Group Learning Algorithm (GLA) with traditional neural networks to enhance the predictability of heart disease, in terms of research results. The proposed method in this work aims to improve the trade-off between exploration and exploitation in stochastic optimization, thereby enhancing model efficacy. One of the significant contributions of this study is the formulation of a technique that leverages the complementary nature of neural networks to achieve more accurate predictive performance than traditional approaches. With this innovation, we aim to promote AI-driven tools for early diagnosis of heart disease, which will, in turn, enable timely clinical decision-making and improve patient outcomes.

The primary objective of this study is to develop a predictive model of cardiovascular disease by enhancing the performance of artificial neural networks through the application of metaheuristic optimization algorithms. This effectively reduces the error rate, improves the accuracy rate of predictions, and expands the generalizability of the findings to new datasets.

The main objectives of this study are:

- To design and implement a hybrid predictive model that integrates neural networks with GLA to enhance the accuracy of heart disease prediction.
- To improve the model's performance not only in terms of accuracy, but also in terms of robustness to noisy or imbalanced data, and its ability to generalize across multiple heart disease datasets.
- To reduce the error rate through effective parameter tuning and selection of the most influential features.

To achieve the main objectives of this study, this research addresses the following questions: Does the proposed model outperform conventional models in terms of accuracy? Does the model maintain high robustness when exposed to noisy or imbalanced data? Can the proposed model generalize its performance well across multiple cardiac datasets, not just a single dataset?

The contributions of this research are as follows:

- Developing a hybrid framework: Introducing a predictive model based on neural networks with an optimization algorithm (GLA) integration, allowing for improved prediction accuracy for heart disease.
- Improving parameter and feature selection: Adopting advanced strategies to optimize parameter values and select essential features from the data reduces the error rate and increases the model's generalizability.
- Comparison with traditional methods: Conduct comparative experiments between the proposed and conventional models, highlighting the differences in performance and effectiveness in predicting heart disease.

The rest of the paper is organized as follows: Section 2 discusses previous studies, Section 3 explains the methodology employed in this work, Section 4presents the results and discussion, and finally, Section 5 contains conclusions and future work.

2 Literature review

AI applications, in particular Machine learning have been used in many applications as investigated by [14-20]. A large number of researchers have tried to formulate and enhance models of predicting heart disease by using different algorithms and approaches; the literature review history presented below underscores some of these endeavors.

A three-stage system proposed by [21] includes feature extraction through statistical dispersion analysis, dimensionality reduction through normalized mutual information-based PCA (NM-PCA), and ensemble of the RF, SVM and KNN classifiers. This was optimized by a hybrid Deer Updates Flame Optimization algorithm and gained competitive accuracy on the Cleveland, Switzerland, Hungarian and VA Long Beach datasets. On the same note, Patro et al. [2], used Bayesian optimization to optimize the SVM kernel functions and weights and combined KNN and Naive Bayes classifiers to reach 93.44 percent accuracy on Cleveland dataset.
[22] complemented the functioning of Random Forest by using a variety of training sets based on multi-objective particle swarm optimization (MOPSO). On six datasets (Cleveland, Statlog, SPECT, SPECTF, VA Long Beach and Eric) this method was tested with accuracies to 88.26 percent using classifiers like C4.5, LR and QDA. Cherian et al. [3] combined the LA and PSO algorithm with PCA to reduce features and were found to be effective on UCI datasets.

The recent works focus on metaheuristic-powered optimizations. Manikandan et al. [23] compared the logistic regression, decision tree and support vehicle machine (SVM) performance with Boruta feature selection on Cleveland dataset and with no feature selection. The research revealed that the six pertinent characteristics that Boruta chose enhanced the model performance with the logistic regression being the most effective with an accuracy of 88.52. Cenitta et al. [24] suggested a feature selection algorithm, squirrel search algorithm (IHDSSO), combined with Forest, to find ischemic heart disease risks. A privacy sensitive MABC-SVM framework proposed by Yaqoob et al. [25] enhanced the prediction accuracy by 1.5 percent and

reduced the computation load. [26] combined gravitational search optimization (GSOA) with DBN-CNNs to analyze IoMT data, predicting cardiac risks via SVM-based classification. An improved variant of the Grey Wolf Optimization algorithm was successfully applied to optimize neural network training, resulting in improved robustness and generalization performance on clinical datasets [27]. Hybrid models that combine classical neural networks with quantum elements, as represented by the Kolmogorov-Arnold Classical-Quantum Dual-Channel Neural Network, offer better interpretability and uncertainty estimation while maintaining competitive predictive performance [28].

Additionally, applying deep learning models supported by sophisticated feature augmentation methods has demonstrated a predictive accuracy enhancement of over 4%, indicating the significance of sophisticated data augmentation and optimization strategies [29]. Lastly, Begum et al. [30] confirmed the utility of integrating metaheuristic algorithms, such as HHO and PSO, with neural network frameworks, thereby facilitating the development of more precise and interpretable predictive solutions in cardiovascular diagnosis.

The previous works indicate the possibility of integrating metaheuristics with neural networks in order to deal with variability of medical data and improve the robustness of prediction. Nevertheless, there are still issues of weighting the exploration-exploitation trade-offs and extrapolating the models across heterogeneous data sets, hence our addition of a Group learning algorithm to neural network to improve heart disease forecasting performance.

3 Methodology

The research methodology suggested will optimize neural network configurations to predict heart disease by combining a Group learning algorithm (GLA) [31] to hyperparameter tuning and feature selection. The methodology can be broken down into four major steps:

3.1. Data preprocessing and chromosome encoding

The raw data are subjected to conventional preprocessing, such as removal of values of missing data, and normalization. In this work, missing values were handled by mode substitution per column. To avoid data leakage normalization using a Z-score was used with the parameter of the scale (mean and std) being learned only on the training set and then applied to the validation and test sets. Every indicator is then coded as 13 features and a binary target variable (0 or 1). The hyperparameters of the neural network, as well as choices of features are represented as a single chromosome, enabling an efficient search in high-dimensional parameter space. This chromosome is made out of two parts (Figure 1): a parameter section and a feature selection bit string. The parameter segment represents hyperparameters, including the number of hidden layers, neurons each layer, activation function, dropout rate, learning rate, the batch size, number of epochs, and optimizer. All numeric hyperparameters are rescaled to [0,1], and categorical values are coded as integer indices. The feature picking step includes a series of 13 bits, and each bit denotes the presence or absence of the respective feature in the training.

Hidden layer	neurons	activation	dropout	Learning rate	Batch size	epochs	optimizer	Feature 1	...	Feature 13
Number	Number	List	Rate	Rate	Number	Number	List	(0/1)	...	(0/1)

Figure 1: Solution representation

3.2 Fitness evaluation through neural network training

The encoded hyperparameters and feature subsets are then decoded in every candidate solution (chromosome). Then a feed-forward neural network is constructed through these settings and the training data is trimmed down to include only the features chosen by the chromosome. A sample of the preprocessed data is fed into the neural network which is then trained and the neural network performance is determined using the accuracy as the fitness measure on a dataset. This fitness test is used as the objective function to be used in optimization.

3.3. Optimization using the group learning algorithm (GLA)

To explore the intricate search space, there is the use of GLA. The algorithm starts with an initial random population of chromosomes. The most efficient solution is the so-called manager. This group is then further subdivided into groups and within a given group a group leader is then identified depending on fitness. The algorithm repeatedly refines the group leaders and individual solutions with the help of a set of operations:

- Leader Update: All group leaders are tuned depending on the variations between the world manager and a random scaling value.
- Group Member Update: Group members are updated by moving to the updated group leader.
- Global Influence: The optimum solution (manager) makes the rest of the people work toward converging. Also, mutation operator is introduced randomly to bring variations to make sure that the search space is explored and premature convergence is avoided The optimal chromosome is chosen after a specified number of steps which is the optimal combination of the neural network parameters as well as the most informative set of features.

3.4. Enhanced neural network for prediction of heart disease

The final neural network model is then configured by using the optimized chromosome. This more refined model, which benefits by optimized hyperparameters and a more fined feature subset, will have improved accuracy and generalization performance in predicting

heart disease risk. The last model is implemented experimentally and compared to the conventional optimization methods. Figure 2 presents the general structure of the proposed approach.

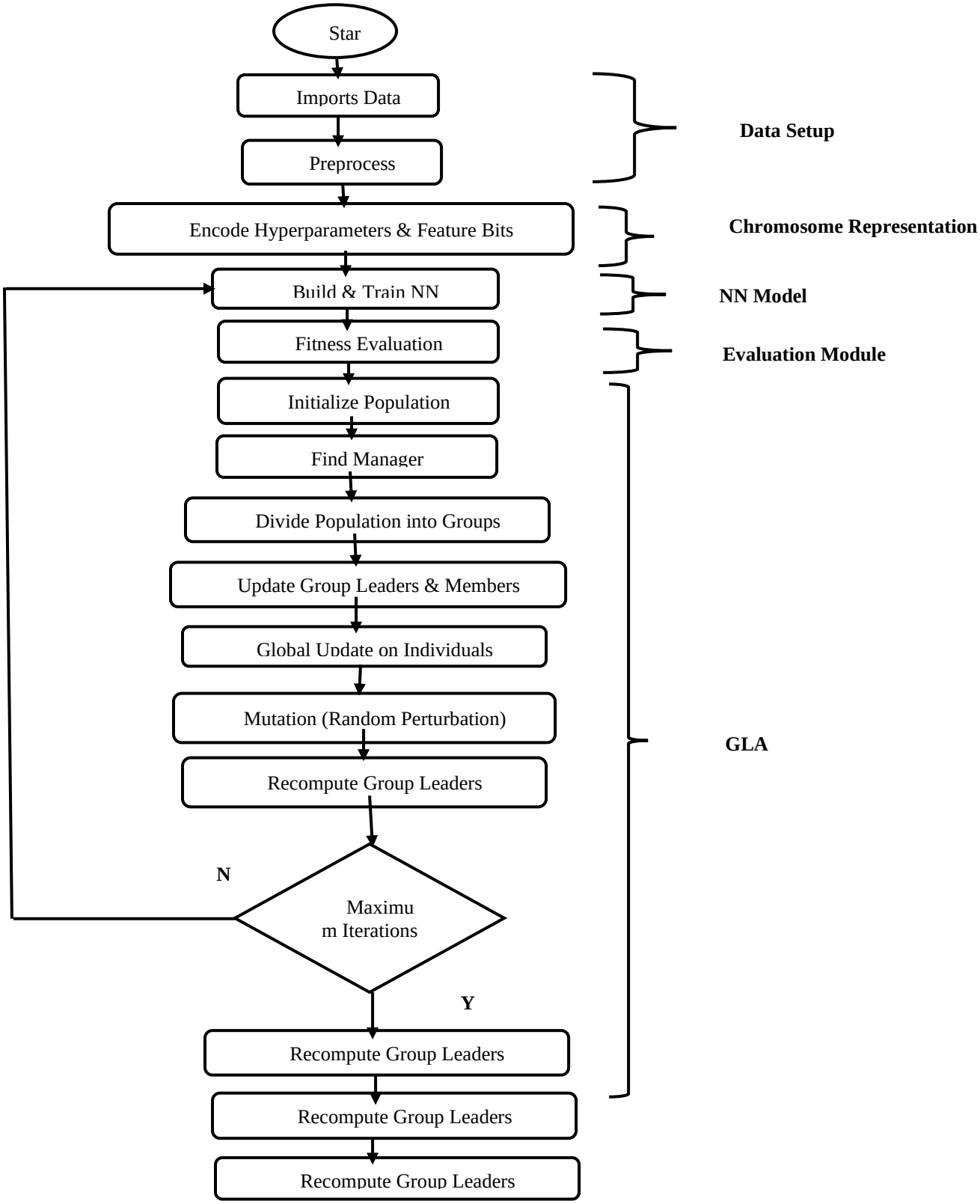


Figure 2: General structure of the proposed approach

.

Algorithm 1 represents the proposed method in general.

Algorithm 1: Neural network optimization based on group learning algorithm
Input: Dataset, Population size (pop_size), Maximum iterations (max_iter), Number of groups (groups_count)
Output: Best solution, NN Model.
1. Initialize a population randomly.
2. Building an NN according to each solution.
2. Training NN models and evaluating the fitness of each solution in the population.
3. Identify the global manager as the solution with the highest fitness.
4. Divide the population into groups randomly.
5. Select the group leader (best solution) for each group.
6. For iteration = 1 to max_iter do:
For each group, update the group leader:
· Compute LA = (group_leader - manager) * r, where r ∈ [0,1]. (LA is the new group leader after it is affected by the manager.)
• Evaluate the fitness of the new leader and update if it improves
For each group member (excluding the leader): Update the individual:
• Compute New individual = (group_leader - individual) * r
• Evaluate fitness and update the manager if necessary.
• For each individual (excluding the manager): Update using global influence:
• Compute New individual = (individual - manager) * r
• Evaluate fitness and update the manager if necessary.
• Apply mutation: For each chromosome, with a probability equal to the mutation rate, randomly perturb one or more genes.
• Recompute group leaders for the next iteration based on updated fitness.
7. End For.
8. Return the best solution (manager) and NN Model.

We use an initial mutation rate of 0.1, which is gradually reduced by 0.95 with each iteration, applied to each member with a probability equal to the rate value. When a mutation occurs, one variable is randomly selected to be changed. The generation method depends on the type of variable. This mechanism aims to balance exploration with exploitation, with the mutation probability gradually decreasing to allow for finer tuning in the later stages.
After several iterations, the best chromosome (manager) is identified. This chromosome contains the optimal neural network hyperparameters and the best subset of features for predicting heart disease.

The optimized configuration is then used to build the final neural network model, which is expected to perform better in terms of accuracy and generalization. The final model is deployed to predict heart disease, providing a more accurate risk assessment to support clinical decision-making.
The computational complexity of the proposed algorithm is examined from two distinct viewpoints (the optimization phase and the model training/prediction phase). GLA is meticulously crafted to conduct an efficient search for optimal hyperparameters, with its complexity principally influenced by the population size (pop_size), the number of iterations (max_iter), and the computational cost associated with evaluating each individual. Given that each evaluation necessitates the training of a neural network for a maximum of (E) epochs, the time complexity of the optimization procedure can be approximated as O(pop_size * max_iter * E).
For the final trained model, the complexity associated with executing a singular prediction is predominantly governed by the feedforward propagation through the network. This operation is remarkably efficient and is contingent upon the number of hidden layers (L), the quantity of neurons per layer (K), and the total number of input features (F). Consequently, the complexity for a single prediction can be expressed as O (L * K * F). The cumulative complexity of the algorithm is primarily dictated by the hyperparameter optimization phase, which, although more computationally demanding than a singular training execution, is imperative for attaining the high performance corroborated in this work.

4 Results and discussion

Some heart disease datasets are available in the UCI repository, including those from Switzerland, Hungary, and Cleveland. This dataset contains 303 instances and 75 attributes. However, a subset of 14 of them was used in all reported tests. The target column in the presented dataset has two classes: 1 for heart disease and 0 for other conditions. The description for this dataset is given in Table 1. The reason for selecting a set of 14 features from the available features in the dataset is as follows:
First, we relied on a literature review and used the most common features in previous studies as a primary reference. Second, we analyzed the proportions of missing values for each feature and excluded variables that exceeded a specified percentage threshold for missingness. Finally, we conducted univariate analyses (Correlation tests and mutual information calculation) to assess the strength of the relationship between each feature and the target variable, and we used the results of these tests to select the final set.
The percentage of training data in the current work is 64%, the percentage of testing is 20%, and the percentage of validation is 16%.

Table 1:Cleveland dataset description

Attribute	Description
Age	Patient's Age
Gender	Patient's Gender
Cp	Type of Chest Pain

Thestbps	Resting Blood Pressure
Chol	Serum Cholesterol Level
Restecg	Resting Electrocardiographic Results
Fbs	Fasting Blood Sugar
Thalach	Maximum Heart Rate Achieved
exang	Exercise-Induced Angina
Oldpeak	ST Depression Induced by Exercise Relative to Rest
Slope	Slope of the Peak Exercise ST Segment
Ca	Number of Major Vessels Visualized
Thal	Type of Defect
Num	Diagnosis of Heart Disease (Angiographic Status)

For the preprocessing in this study, several steps were implemented to replace missing values, convert columns to numeric type, and convert target formats to binary. We used direct validation with Early Stopping to prevent overfitting and obtain the best validation weight. After preprocessing the dataset for heart disease prediction, we applied the proposed method, which integrates the Group learning algorithm (GLA) to enhance the neural network's structure and feature selection process. The optimization focused on determining the optimal number of hidden layers and selecting the appropriate number of neurons in each layer, thereby ensuring a thorough exploration of the solution space while leveraging local solutions. Table 2 shows the important parameters in the work and their values.

Table 2: Setting parameters

Parameter	Value
Population size	20
Number of groups	4
Initial mutation rate	0.15
Maximum iterations	50

The application of the proposed approach in rigorous experiments established an accuracy rate of 93.44, which is significant as compared to the conventional use of neural networks. The experiment was replicated 30 times in order to make sure that the proposed model performance is assessed thoroughly. The findings indicated an average accuracy of 88.99 with low standard deviation of 3.01. This low standard deviation means the performance of this model is not only due to a chance, but also is very stable and reliable in all runs. Table 3 presents the hyperparameters obtained via the application of our proposed method that also enhanced the performance of the neural network model.

Table 3: Hyperparameters chosen using our proposed approach

Hyperparameters	Value
hidden_layers	2
neurons	42
activation	elu
dropout	0.1847
learning_rate	0.00175
batch_size	38
epochs	75
optimizer	Adam

It should be mentioned that the dropout rate is rather small (0.1847), and this effect can be explained by the presence of the hyperparameter optimization algorithm designed to locate the best equilibrium between the model complexity and the generalization.

The Confusion Matrix of the evaluation process of the classification model is described in figure 3. The table shows how effective the model was to classify both classes appropriately. There are more values on the central diagonal (27 and 30) than those on the off-the-diagonal (2 and 2) that illustrate a better model performance and fewer prediction errors. In this figure, the model properly categorized the majority of samples with few errors (2 false positives and 2 false negatives) in this figure. This shows that the model has the capability of reducing classification errors in the negative (0) and positive (1) classes, an encouraging fact on its usefulness in similar prediction tasks.

The Training and Validation Accuracy curves are shown in figure 4 throughout the training epochs. The training accuracy (blue line) in the figure rises sharply during the first epochs, then tends to vary at a high level, which means that the model is successfully learning the trends in training data. The validation accuracy (the orange line) also shows a generally increasing trend, staying relatively close to the training accuracy, which suggests that the model is generalizing well to unseen data and is not overfitting. The minimal gap and similar behavior between the two curves are a promising sign of the model's effectiveness in prediction tasks.

Figure 5 is the Receiver Operating Characteristic (ROC) curve, with an area under the curve (AUC) of 0.96. This highly significant value indicates the model's excellent capacity to classify samples correctly and, hence, its effectiveness in striking an optimal trade-off between the False Positive Rate (False Alarm Rate) and the True Positive Rate (actual detection rate)

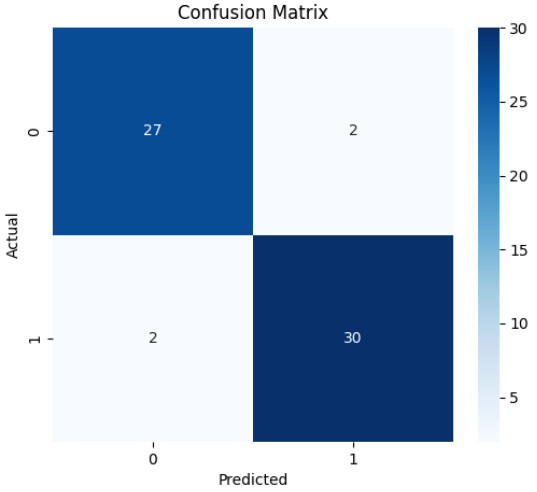


Figure :3 Confusion matrix of the model

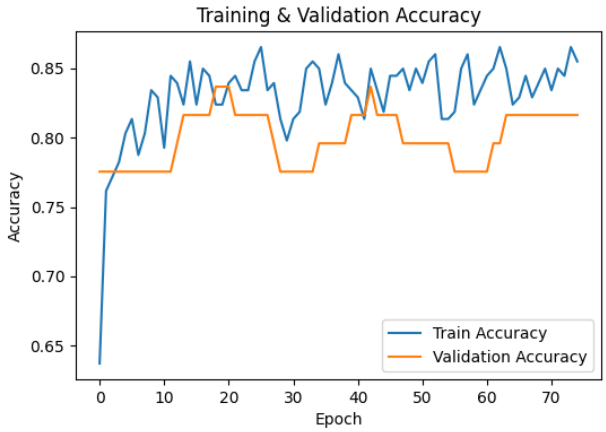


Figure :4 Training and validation accuracy and loss curves

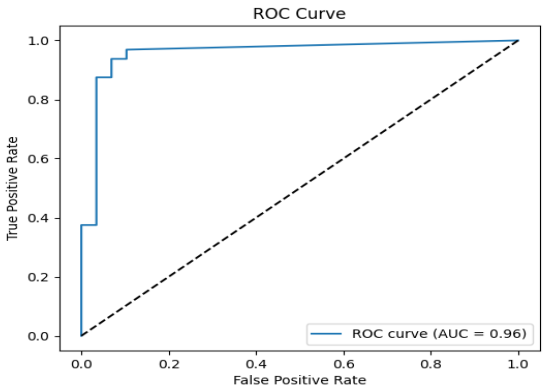


Figure: 5 Receiver operating characteristic

The application of the GLA algorithm in order to optimize the architecture of the neural network and the most relevant features selection has played a critical role in enhancing the accuracy of prediction. These attractive results suggest the prospects of using GLA to other research areas, where it can boost the work of traditional neural networks.

One of the main weaknesses of this study is that it has high computational complexity especially with deep models and large datasets. Nevertheless, this complexity

does not pose a serious barrier, because it is restricted to the model training stage alone.

Moreover, as it is indicated in Table 4, the outcomes of our proposed model are better than those of the existing literature, which proves the necessity to consider it as a more efficient instrument in predicting heart diseases. This shows how strong GLA is in improving the neural network structure. The findings reaffirm the ability of GLA to optimize the neural network morphology and feature selection to achieve more accurate heart disease prediction.

Table 4: Comparison between the proposed work and related studies.

Previous Study	Dataset	Method	Accuracy
Sheeba, et al. [14]	Cleveland, Hungarian Switzerland, and VA Long Beach	NM-PCA	91%
Patro, et al. [2]	Cleveland	BO-SVM	93.3%
Asadi, et al. [15]	Cleveland Statlog, SPECT SPECTF VA Long Beach Eric	MOPSO	Cleveland, 85.21% Statlog, 88.26 % SPECTF, 87.65 % SPECTF, 86.70% VA Long Beach, 87.50 Eric, 80.95 %
Proposed method	Cleveland	GLA Algorithm	93.44%

The proposed GLA Algorithm attained an accuracy rate of 93.44%, surpassing the efficacy of prior methodologies including NM-PCA (91.0%) and MOPSO (85.21%), while marginally exceeding the performance of BO-SVM (93.3%). The discerned disparities, especially in comparison to MOPSO and NM-PCA, suggest a significant enhancement in performance.

5 Conclusion

The study demonstrates that applying the Group Learning Algorithm (GLA) in a neural network architecture can significantly enhance heart prediction. Through skillful navigation of complex and large search spaces, the GLA-based optimization method effectively identifies the most critical parameters and feature subsets, resulting in improved accuracy and generalization compared to traditional optimization techniques. The experimental findings validate the competitive benefits of this solution, emphasizing its potential to reinforce predictive models and facilitate earlier and more accurate diagnoses of heart disease in clinical practice. Innovation underpins the viability of sustaining research activities and offers a model that can be applied to other health issues. Future research will enhance the GLA framework by incorporating adaptive hyperparameter tuning tailored to more complex datasets, evaluating its performance on larger, multi-modal datasets, and investigating its compatibility with advanced neural network architectures to achieve further improvements, and try to apply the model interpretability to future work.

References

[1] N. K. Munagala, L. R. R. Langoju, A. D. Rani, D. R. K. J. b. Reddy, and b. engineering, "A smart IoT-enabled heart disease monitoring system using meta-heuristic-based Fuzzy-LSTM model," vol. 42, no. 4, pp. 1183-1204, 2022.

[2] S. P. Patro, G. S. Nayak, and N. J. I. i. M. U. Padhy, "Heart disease prediction by using novel optimization algorithm: A supervised learning prospective," vol. 26, ed: Elsevier, 2021, p. 100696.

[3] R. P. Cherian, N. Thomas, and S. J. J. o. B. I. Venkitachalam, "Weight optimized neural network for heart disease prediction using hybrid lion plus particle swarm algorithm," vol. 110, p. 103543, 2020.

[4] A. Dutta, T. Batabyal, M. Basu, and S. T. J. E. S. w. A. Acton, "An efficient convolutional neural network for coronary heart disease prediction," vol. 159, p. 113408, 2020.

[5] Pipalwa, R., Paul, A., & Mukherjee, T. (2023). "Prediction of heart disease using modified hybrid classifier". Informatica, 47(1)..

[6] J. E. Dalen, J. S. Alpert, R. J. Goldberg, and R. S. J. T. A. j. o. m. Weinstein, "The epidemic of the 20th century: coronary heart disease," vol. 127, no. 9, pp. 807-812, 2014.

[7] A. Khan, M. Qureshi, M. Daniyal, K. J. H. Tawiah, and S. C. i. t. Community, "A novel study on machine learning algorithm ‐ based cardiovascular disease prediction," vol. 2023, no. 1, p. 1406060, 2023.

[8] D. Shah, S. Patel, and S. K. J. S. C. S. Bharti, "Heart disease prediction using machine learning techniques," vol. 1, no. 6, p. 345, 2020.

[9] Sekkal, M., Benzina, A., & badir Benkrelifa, L. (2024). "Multi-Objective Evolutionary Algorithm based on NSGA-II for Neural Network Optimization Application to the Prediction of Severe Diseases". Informatica, 47(10).

[10] Y. H. Farhan, S. A. Mohd Noah, M. Mohd, and J. Atwan, "Word-embedding-based query expansion: Incorporating Deep Averaging Networks in Arabic

document retrieval," Journal of Information Science, vol. 49, no. 5, pp. 1168-1186, 2021.

[11] X.-S. Yang, Nature-inspired metaheuristic algorithms. Luniver press, 2010.

[12] Singh, A., Prakash, N., & Jain, A. (2025). "Meta-Heuristic Optimization for the Multi-Classification of Chronic Disease: A Review With Machine Learning Perspectives". Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 15(3), e70030.

[13]H. V. Ramesh and R. K. Pathinarupothi, "Performance Analysis of Machine Learning Algorithms to Predict Cardiovascular Disease," in 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), 2023, pp. 1-8: IEEE.

[14]B. Shannaq, "DEVELOPMENT OF A NEW ENCODING ALGORITHM USING VIRTUAL KEYPAD LETTER SUBSTITUTIONS FOR IMPROVED TEXT CLASSIFICATION," J. Theor. Appl. Inf. Technol., vol. 103, no. 2, Jan. 2025, [Online]. Available: https://www.jatit.org/volumes/Vol103No2/27Vol103No2.pdf.

[15] F. J. I. Alazzawi, B. Shannaq, and A. Al-Zeidi, "Predictive Modeling for Student Specialization Selection: Analyzing Influences and Developing Tailored Recommendations," in Advances in Information and Communication, vol. 1285, K. Arai, Ed., in Lecture Notes in Networks and Systems, vol. 1285. , Cham: Springer Nature Switzerland, 2025, pp. 633–645. doi: 10.1007/978-3-031-84460-7_40.

[16] B. Shannaq, "Optimizing n-gram lengths for cross-linguistic text classification: A comparative analysis of English and Arabic morphosyntactic structures," Int. J. Adv. Appl. Sci., vol. 12, no. 4, pp. 136–145, May 2025, doi: 10.21833/ijaas.2025.04.015.

[17] Y. H. Farhan et al., "UTILIZING WORD EMBEDDING'S FOR AUTOMATED QUERY EXPANSION IN ARABIC INFORMATION RETRIEVAL: A BLENDED METHODOLOGY," J. Theor. Appl. Inf. Technol., vol. 103, no. 2, Jan. 2025, [Online]. Available: https://www.jatit.org/volumes/Vol103No2/33Vol103No2.pdf.

[18] B. Shannaq, "Leveraging AI to Forecast Future Job Market Trends: Smart Solutions and Strategic Recommendations for Career Success," J. Inf. Knowl. Manag., p. 2550070, Aug. 2025, doi: 10.1142/S0219649225500704.

[19] B. Shannaq, "Enhancing Arabic Text Classification through Mobile Virtual Keypad-Based Encoding

Algorithm," Frankl. Open, p. 100373, Sep. 2025, doi: 10.1016/j.fraope.2025.100373.

[20] B. Shannaq, "Leveraging AI to Forecast Future Job Market Trends: Smart Solutions and Strategic Recommendations for Career Success," J. Inf. Knowl. Manag., p. 2550070, Aug. 2025, doi: 10.1142/S0219649225500704.

[21] P. T. Sheeba, D. Roy, and M. H. J. A. i. E. S. Syed, "A metaheuristic-enabled training system for ensemble classification technique for heart disease prediction," vol. 174, p. 103297, 2022.

[22] S. Asadi, S. Roshan, and M. W. J. J. o. b. i. Kattan, "Random forest swarm optimization-based for heart diseases diagnosis," vol. 115, p. 103690, 2021.

[23] Manikandan, G., Pragadeesh, B., Manojkumar, V., Karthikeyan, A. L., Manikandan, R., & Gandomi, A. H. (2024). "Classification models combined with Boruta feature selection for heart disease prediction". Informatics in Medicine Unlocked, 44, 101442.

[24] D. Cenitta, R. V. Arjunan, and K. J. I. A. Prema, "Ischemic heart disease prediction using optimized squirrel search feature selection algorithm," vol. 10, pp. 122995-123006, 2022.

[25] M. M. Yaqoob, M. Nazir, M. A. Khan, S. Qureshi, and A. J. A. S. Al-Rasheed, "Hybrid classifier-based federated learning in health service providers for cardiovascular disease prediction," vol. 13, no. 3, p. 1911, 2023.

[26] A. Sampathkumar et al., "Internet of Medical Things (IoMT) and Reflective Belief Design‐Based Big Data Analytics with Convolution Neural Network‐Metaheuristic Optimization Procedure (CNN‐MOP)," vol. 2022, no. 1, p. 2898061, 2022.

[27] S. Niu, Y. Zhou, Z. Li, S. Huang, and Y. Zhou, "An Improved Grey Wolf Optimization Algorithm for Heart Disease Prediction," in IEEE International Conference on Advanced Infocomm Technology, 2023, pp. 527-542: Springer.

[28] M. A. Jahin, M. A. Masud, M. Mridha, Z. Aung, and N. J. a. p. a. Dey, "Kacq-dcnn: Uncertainty-aware interpretable kolmogorov-arnold classical-quantum dual-channel neural network for heart disease detection," 2024.

[29] M. T. García-Ordás, M. Bayón-Gutiérrez, C. Benavides, J. Aveleira-Mata, J. A. J. M. T. Benítez-Andrades, and Applications, "Heart disease risk prediction using deep learning techniques with feature augmentation," vol. 82, no. 20, pp. 31759-31773, 2023.

[30] F. Begum, J. A. J. B. M. Valan, and Devices, "Enhancing Heart Disease Diagnosis with Meta-Heuristic Algorithms: A Combined HHO and PSO Approach," pp. 1-16, 2025.

[30] C. M. J. N. C. Rahman and Applications, "Group learning algorithm: a new metaheuristic algorithm," vol. 35, no. 19, pp. 14013-14028, 2023.

Empirical Analysis of Dataset Size Impact on Classification Performance in Precision Agriculture Using Machine Learning Models

Khadija Lechqar*, Mohammed Errais
Data Engineering and Intelligent Systems Laboratory, Faculty of Sciences Ain Chock, Hassan II university, Casablanca, Morocco
E-mail: khadija.lechqar@gmail.com
*Corresponding author

Keywords: Size of dataset, performance, precision agriculture

Received: January 24, 2025

This study empirically investigates the relationship between dataset size and classification performance in precision agriculture applications. Seven machine learning models (Decision Tree, Random Forest, Logistic Regression, SVM, Gaussian Naïve Bayes, KNN, and AdaBoost) were evaluated on seven agricultural datasets ranging from 100 to 4,000 samples. Performance was assessed using five metrics: accuracy, precision, recall, F1-score, and ROC-AUC. The methodology involved two phases: initial evaluation using complete datasets, followed by systematic analysis of subdivided datasets to examine performance variation with data volume. Statistical analysis using Pearson correlation coefficients revealed no significant correlation between dataset size and model performance ($r = 0.12$, $p > 0.05$). Results indicate that Random Forest and Decision Tree models achieved the highest average performance across datasets (88.48% and 85.37% accuracy, respectively). The findings suggest that dataset quality and problem characteristics have greater influence on classification performance than dataset size alone in precision agriculture applications.

Povzetek: Empirična študija na sedmih kmetijskih naborih pokaže, da velikost podatkov ni značilno povezana z uspešnostjo klasifikacije (pomembnejša sta kakovost in narava problema), pri čemer se najbolje izkažeta naključni gozd in odločitveno drevo.

1 Introduction

Precision agriculture represents a paradigm changing in agricultural practices, designed to address the growing global demand for food production through data-driven decision-making. Modern precision agriculture has adopted artificial intelligence as a core technology. Machine learning models are now increasingly used, starting from pre-harvest automation to irrigation scheduling, disease identification, weed control, yield forecasting, and pest management [1,2]. This technological changing has made farming operations more efficient while promoting sustainability [3].

Yet gathering agricultural data is problematic and directly affects how well machine learning systems work [4]. Farmers and researchers use various methods to collect information. Satellites and UAVs capture aerial images for remote monitoring [5,6]. Ground sensors placed near crops provide detailed, accurate readings [7]. Robots move through fields collecting data, while IoT devices monitor environmental conditions continuously. Linking all these data sources requires sophisticated communication networks [8]. The entire process costs significant money and time, demanding substantial investment in both equipment and expertise.

Datasets serve as fundamental building blocks for developing effective machine learning models [9]. Real-world agricultural data often suffers from quality issues due to environmental variables, sensor limitations, and data transmission challenges. Class imbalance, where minority and majority classes have unequal representation in classification problems, frequently leads to reduced model accuracy [10,11]. Previous research has suggested that larger datasets generally improve classification model performance [12], yet the specific relationship between dataset size and performance in precision agriculture contexts is underexplored.

However, one of the challenges of precision agriculture is the collection of data [4]. In fact, different devices are used for this purpose: satellites and UAVs for remote sensing process and aerial images[5][6], ground-based sensors which have the advantage of being close proximity to plants and providing good data accuracy[7], robots and other IoT devices. In addition, communication devices are also used to link these different components[8]. The cost of data collection is therefore high in terms of time and resources.

1.1 Research objectives and hypotheses

This study addresses the following research questions:

1. What is the relationship between dataset size and classification model performance in precision agriculture applications?

2. How do different machine learning algorithms respond to variations in dataset size?

3. What factors beyond dataset size influence model performance in agricultural classification tasks?

We hypothesize that while dataset size may influence performance, the relationship is not linear and that dataset quality characteristics play a more significant role in determining model effectiveness.

2 Related work

Several studies have examined the relationship between dataset size and machine learning model performance across different domains. This section provides a critical analysis of existing literature and identifies research gaps that this study addresses.

2.1 Literature review and gap analysis

Alshammari and Alshayeb [13] investigated software defect prediction, demonstrating that dataset size directly affects SVM model performance. Their results showed that datasets with fewer metrics enabled better SVM performance for defect prediction, although computational efficiency was not guaranteed for smaller datasets. However, this study was limited to software engineering applications and focused primarily on SVM models.

Imlawi and Alsharo [14] examined the impact of resampling techniques and dataset size on classification accuracy. They found that resampling methods were particularly beneficial for limited datasets (100-500 samples) and helped reduce biased estimation. While relevant to small dataset scenarios, their study did not address the specific challenges of agricultural data collection and processing.

Bailly et al. [15] focused on dataset size effects in prediction tasks using logistic regression and deep learning models. Their findings indicated that traditional machine learning algorithms were less influenced by dataset size but required feature interaction optimization for optimal performance, unlike deep learning models. This work provided insights into algorithm-specific responses to data volume but lacked domain-specific analysis for agriculture.

Althnian et al. [12] conducted an empirical study in the medical domain, concluding that model performance depends more on dataset distribution characteristics than size alone. They demonstrated that Adaptive Boosting (AB) and Naïve Bayes (NB) models showed greater robustness with limited medical data. However, their focus on medical applications limits direct applicability to agricultural contexts.

Lin et al. [16] studied multiclass classification performance as a function of dataset size and training/test split ratios, providing insights into data partitioning strategies but not addressing domain-specific agricultural challenges.

2.2 Summary of prior research

Table 1 summarizes key findings from previous studies on dataset size impact across different domains.

Table 1: Summary of related work

Study	Field	Dataset size range	Model used	Key findings
[13]	Software Engineering	100-2000	svm	Smaller datasets improved SVM performance
[14]	General Classification	100-500	Multiple	Resampling beneficial for small datasets
[15]	Biomedical	500-5000	LR, Deep Learning	Traditional ML less affected by size
[12]	Medical	1000-10000	Multiple	Distribution more important than size
[16]	General Classification	34-181	U-net	virtual sample generation method enhancing prediction accuracy for small datasets in high dimensions

2.3 Research gap identification

The literature review reveals several critical gaps that this study addresses:

1. Domain-specific analysis: No comprehensive study has examined dataset size impact specifically in precision agriculture contexts, where data collection challenges and environmental variables create unique constraints.

2. Systematic model comparison: Previous studies have not comprehensively compared multiple machine learning algorithms under consistent experimental conditions within agricultural applications.

3. Statistical validation: Many existing studies lack rigorous statistical analysis to validate claims about dataset size relationships.

4. Practical implications: Limited research has addressed the practical implications of dataset size constraints in resource-limited agricultural settings.

This study fills these gaps by providing a systematic empirical analysis of dataset size impact on classification

performance using multiple machine learning models specifically in precision agriculture applications.

3 Methodology

The main objective of this work, as mentioned above, is to study the effect of limited dataset on the performance of classification problems. It will be focused on tabular datasets in the agriculture field. As a first step, we selected seven datasets of different sizes and applied seven machine-learning models for classification. To take this study further, we worked on the two largest selected datasets and divided them into sub-datasets. The aim is to evaluate the variation of the performance compared with the size of dataset for the same classification problem.

3.1 Dataset selection and characteristics

We used seven datasets selected from precision agriculture field. The criteria of selection are:

1. Representativeness of common agricultural classification problems.
2. Availability of sufficient samples for subdivision analysis.
3. Diversity in feature types and problem complexity Balanced representation of different agricultural domains

The number of instances varies from 100 to 4000 and the number of features is between 4 and 9 features. Table 2 summarizes their structures.

Table 2: An overview of the datasets

Dataset	Field	Number of rows	Number of features	Type of data	balanced data
1	Fertilization	100	9	Numeric, Text	yes
2	Irrigation	201	4	Numeric	yes
3	Irrigation	501	6	Numeric	yes
4	Weather prediction	1461	8	Numeric, Text	no
5	Crop recommendation	2200	8	Numeric, Text	yes
6	Quality of water	3276	9	Numeric, Text	yes
7	Apple quality	4000	9	Numeric, Text	yes

3.2 Dataset subdivision strategy

To systematically examine dataset size impact, the two largest datasets (Dataset 6: 3,276 samples and Dataset 7: 4,000 samples) were randomly subdivided into smaller subsets while maintaining class distribution proportions (Table 3). The subdivision strategy was designed to create meaningful size variations for statistical analysis.

Table 3: Description of the two sub-datasets

Dataset	Field	Number of rows
6	Quality of water	3276
6-1		500
6-2		1000
6-3		2000
7	Apple quality	4000
7-1		500
7-2		1000
7-3		2000

3.3 Data preprocessing

We applied several preprocessing steps to clean our data and make it suitable for machine learning models:
Handling missing values: When data points were missing, we used different strategies depending on the variable type. For numerical data, we filled gaps with the average value. For categorical data, we used the most common category.
Scaling features: Since our numerical features had different ranges, we standardized them using z-score normalization. This prevents features with larger values from dominating the models.
Converting categories: Some models only work with numbers. So, we converted categorical variables into numerical format using one-hot encoding.
Managing outliers: We found outliers using the interquartile range method. Rather than removing these data points (which would shrink our already limited datasets), we used winsorization to cap extreme values.

3.4 Machine-learning models

For each dataset, we used seven machine-leaning models for classification. They are namely:
K-Nearest Neighbors KNN: is a model of supervised learning. The predicted value is based on the values of the K nearest neighbors based **using** Euclidean, mahalanobis, or Manhattan [1].
Decision Trees DT: are designed for supervised data mining. They have the structure of tree; each node presents a feature. It has the advantage of solving problems with complex data [3].
Random Forest RF: It is an ensemble machine learning approach. It consist on combining the results of multiple decision trees to reach a single result[2].
Gaussian Naïve Bayes GNB: is used for classification purposes. It is based on probabilistic approach and supposes that features have a normal distribution[4][17].
Logistic regression LR is a linear model giving the relationship between input and output variables. In fact, it predicts the probability that the input belongs to a class or not[5].

Adaptive Boosting AdaBoost is an ensemble model based on weighted instances of data in function of previous classifications. The final aim is to improve, gradually, the accuracy [6].

Support Vector Machine SVM is a supervised machine learning model. It consists on finding a line or a hyperplane according to the number of features in the input. The goal is to classify the data points in input [12].

3.5 Performance metrics

In this work, the performance of models is evaluated by using the known five metrics: accuracy precision recall, f1, roc auc. They give an idea about positive and negative

correctly or incorrectly classification and so the performance of the applied models:

Accuracy: Overall correctness of predictions [18].

Precision: Proportion of true positive predictions among positive predictions.

Recall: Proportion of true positive instances correctly identified

F1-Score: Harmonic mean of precision and recall

ROC-AUC: Area under the receiver operating characteristic curve

3.5 Experimental setup

Our experiments ran on:

Computer: Windows 10 machine with Intel Core i5 2.6 GHz processor and 8GB RAM

Programming Tools: Python 3.8.5, scikit-learn 0.24.2, pandas 1.3.3, numpy 1.21.2

Testing method: 5-fold cross-validation to get reliable performance estimates

Reproducibility: Set random seed to 42 so results can be replicated

3.6 Statistical analysis

We used several statistical methods to determine whether dataset size really affects performance:

Pearson correlation coefficients to measure linear relationships

ANOVA tests to compare performance across different size groups

Mann-Whitney U tests for cases where data wasn't normally distributed

4 Results and discussion

4.1 Performance analysis of complete datasets

The experimental results demonstrate varying performance patterns across different models and datasets, providing insights into the relationship between dataset size and classification effectiveness.

4.1.1 Accuracy analysis

Accuracy measurements across all models and datasets reveal significant variation in performance that does not correlate linearly with dataset size (Table 4). The average accuracy ranges from 62.99% (Dataset 6) to 98.92% (Dataset 2), indicating that dataset characteristics beyond size substantially influence model performance.

Statistical analysis reveals no significant correlation between dataset size and accuracy (Pearson $r = 0.12$, $p = 0.78$). Random Forest achieved the highest average performance (88.48%), followed by Decision Tree (85.37%) and Gaussian Naïve Bayes (83.79%).

4.2.2 Precision analysis

Precision metrics demonstrate similar patterns to accuracy, with performance variations that do not align with dataset size expectations (Table 5).

Decision Tree models achieved the highest average precision (86.95%), closely followed by Random Forest (86.45%). The notably poor performance on Dataset 6 across all models suggests inherent data quality issues rather than size-related limitations

4.1.3 Comprehensive performance analysis

Tables 6, 7, and 8 present the complete results for recall, F1-score, and ROC-AUC metrics, respectively. These results consistently support the finding that dataset size alone does not determine classification performance.

4.1 Analysis of dataset subdivision results

The systematic analysis of subdivided datasets provides direct evidence regarding the relationship between dataset size and performance within identical problem contexts (Table 9).

The subdivision analysis reveals non-monotonic relationships between dataset size and performance. For Dataset 6 subdivisions, accuracy varied irregularly: 60.42% (500 samples), 61.49% (1000 samples), and 59.75% (2000 samples). Similarly, Dataset 7 subdivisions showed: 80.07% (500 samples), 78.43% (1000 samples), and 81.75% (2000 samples). Statistical analysis confirms no significant correlation between subset size and performance (Pearson $r = -0.08$, $p = 0.85$ for Dataset 6; $r = 0.34$, $p = 0.73$ for Dataset 7).

Table 4: Accuracy results for complete datasets (%).

Datatset	Model							
	RF	AdaBoost	LR	SVC	GNB	DT	KNN	Average
1	90.00	60.00	100.00	85.00	85.00	90.00	85.00	85.00
2	99.00	99.00	99.49	99.00	98.50	99.00	98.50	98.92
3	83.87	81.04	77.83	81.25	81.04	83.61	78.05	80.95
4	89.41	52.55	81.91	57.18	88.05	84.98	72.01	75.15
5	99.54	19.54	82.04	85.22	98.63	97.95	88.18	81.58
6	68.44	63.10	62.95	62.95	64.17	62.19	57.16	62.99
7	89.12	78.87	70.75	89.25	71.12	79.87	88.00	80.99
Average	88.48	64.87	82.14	79.98	83.79	85.37	80.99	

Table 5: Precision results for complete datasets.

Datatset	Model							
	RF	AdaBoost	LR	SVC	GNB	DT	KNN	Average
1	94.16	54.09	100.00	85.00	81.25	95.00	93.33	86.11
2	99.35	99.35	99.35	100.00	98.75	99.35	99.33	99.35
3	68.98	86.35	74.26	76.52	80.06	85.83	72.35	77.76
4	88.08	74.16	78.27	57.18	86.95	87.88	71.23	77.67
5	99.57	14.56	83.07	85.42	98.77	98.05	89.11	81.22
6	67.35	60.11	39.63	39.63	61.57	62.52	54.73	55.07
7	87.68	79.00	69.34	87.71	70.14	80.06	86.66	80.08
Average	86.45	66.80	77.70	75.92	82.49	86.95	80.96	

Table 6: Racall metric for complete datasets.

Datatset	Model							
	RF	AdaBoost	LR	SVC	GNB	DT	KNN	Average
1	90.00	60.00	100.00	85.00	85.00	90.00	85.00	85.00
2	99.33	99.33	100.00	98.66	99.33	99.33	98.66	99.23
3	71.42	67.60	74.16	79.75	76.06	73.83	74.61	73.91
4	89.41	52.55	81.91	62.79	88.05	84.98	72.01	75.95
5	99.54	19.54	82.04	85.22	98.63	97.95	88.18	81.58
6	68.44	63.10	62.95	62.95	64.17	62.19	57.16	62.99
7	90.58	79.38	72.51	90.83	71.57	77.60	89.31	81.68
Average	86.96	63.07	81.94	80.74	83.26	83.70	80.70	

Table 7: F1 -score metric for complete datasets.

Datatset	Model							
	RF	AdaBoost	LR	SVC	GNB	DT	KNN	Average
1	88.83	53.42	100.00	81.94	82.00	89.16	85.14	82.92
2	99.33	99.33	100.00	98.66	99.33	99.33	98.66	99.23
3	71.42	67.60	74.16	79.75	76.06	73.83	74.61	73.91
4	88.30	52.42	79.75	59.02	86.42	86.18	71.50	74.79
5	99.53	15.20	81.91	84.71	98.60	97.94	87.98	80.83
6	65.87	58.98	48.64	48.64	59.58	62.34	55.42	57.06
7	89.11	78.68	70.89	89.25	70.94	97.11	87.96	83.42
Average	86.05	60.80	79.33	77.42	81.84	86.55	80.18	

Table 8: ROC-AUC metric for complete datasets

Datatset	Model							
	RF	AdaBoost	LR	SVC	GNB	DT	KNN	Average
1	100.00	82.28	100.00	98.27	100.00	94.44	97.22	96.03
2	100.00	98.66	100.00	100	100.00	98.66	99.90	99.60
3	89.02	93.33	92.85	91.28	92.56	81.79	85.30	89.44
4	93.57	89.88	88.69	87.92	92.50	84.43	84.69	88.81
5	99.99	69.39	98.91	99.62	99.98	98.88	98.28	95.00
6	61.72	55.03	50.00	50.00	55.71	59.89	51.32	54.81
7	89.15	78.88	70.78	89.27	71.13	79.83	88.02	81.00
Average	90.49	81.06	85.89	88.05	87.41	85.41	86.39	

Table 9: Accuracy for the two sub-datasets

Datatset	Model							
	RF	AdaBoost	LR	SVC	GNB	DT	KNN	Average
6	68.44	63.10	62.95	62.95	64.17	62.19	57.16	62,99
6_1	65.85	60.36	58.73	58.73	63.82	58.13	57.31	60,42
6_2	63.41	61.58	63.41	61.58	66.46	58.53	55.48	61,49
6_3	66.15	55.79	60.67	60.06	60.67	60.97	53.96	59,75
7	89.12	78.87	70.75	89.25	71.12	79.87	88.00	80,99
7_1	88.50	76.50	70.00	88.83	71.33	77.50	87.83	80,07
7_2	84.50	74.00	71.00	87.00	73.00	79.00	80.50	78,43
7_3	87.50	77.25	79.25	87.75	75.00	79.50	86.00	81,75

4.2 Statistical validation of findings

Comprehensive statistical analysis supports the conclusion that dataset size does not significantly impact classification performance in the studied agricultural applications:

1. Correlation analysis: Pearson correlation coefficients between dataset size and average performance across all metrics showed no significant relationships (all p-values > 0.05).

2. ANOVA results: Analysis of variance comparing performance across size groups revealed no significant differences ($F = 1.23$, $p = 0.34$ for accuracy).

3. Mann-Whitney U tests: Non-parametric comparisons between small (≤500 samples) and large (≥2000 samples) datasets showed no significant performance differences ($U = 42$, $p = 0.67$).

4.4 Discussion of key findings

4.4.1 Dataset quality vs. dataset size

The results strongly suggest that dataset quality characteristics have a more substantial impact on model performance than dataset size. Dataset 6, despite being among the largest (3,276 samples), consistently showed the poorest performance across all models and metrics. This pattern indicates potential data quality issues such as:

1. High noise levels in feature measurements.
2.Inadequate feature representation of the underlying problem
3. Class overlap or ambiguous classification boundaries
4. Measurement errors or inconsistent data collection procedures.

Conversely, Dataset 2, with only 201 samples, achieved the highest average performance (98.92% accuracy), demonstrating that well-curated, high-quality small datasets can outperform larger, lower quality datasets.

4.4.2 Model-specific performance patterns

Random Forest and Decision Tree models consistently achieved superior performance across datasets, suggesting that tree-based algorithms are particularly well-suited for agricultural classification tasks. This may be attributed to:

Interpretability: Tree-based models provide clear decision rules that align with agricultural domain knowledge.

Feature interaction handling: Ability to capture complex interactions between environmental and agricultural variables.

Robustness to noise: Tree-based methods are relatively resilient to noisy features common in agricultural datasets. AdaBoost showed the most variable performance,

suggesting sensitivity to dataset characteristics and potential overfitting issues with small or noisy datasets.

4.4.3 Implication for precision agriculture

Our findings suggest some important practical considerations for farmers and agricultural technology developers:

Investment: In case of limited resources, there is a need to focus on improving data quality rather than just collecting more data. Better sensors and careful data collection practices will likely give more improvement than simply gathering larger amounts of lower-quality information.

Choosing the right algorithm: Tree-based methods like Random Forest and Decision Trees seem particularly well-suited for agricultural problems. They're not only accurate but also provide interpretable results that farmers can understand and trust.

Data collection strategy: Instead of prioritizing volume, concentrate on sensor accuracy and thorough data preprocessing. Clean, well-processed data from fewer sources often outperforms noisy data from many sources.

4.5 How our results compare to previous research

Our findings both support and extend conclusions from earlier studies:

Agreement with medical domain research: Althnian et al. [12] found that how data is distributed matters more than dataset size in medical applications. Our agricultural study reaches the same conclusion, showing this principle applies across different domains.

Challenging common assumptions: Our agricultural-specific analysis shows the relationship between quantity of data and better performance isn't always straightforward. Sometimes quality trumps quantity.

Building on cross-domain evidence: Studies in software engineering [13] and biomedical fields [15] have reported similar patterns. This suggests that the size-performance relationship might depend heavily on the specific domain and problem characteristics.

5 Limitations and future work

Several limitations should be mentioned in this study: 1. **Dataset Scope**: Analysis was limited to seven agricultural datasets; broader dataset inclusion could strengthen generalizability.

2. **Feature engineering**: Limited exploration of advanced feature engineering techniques that might influence the size-performance relationship.

3. **Temporal aspects:** Static analysis did not consider temporal variations in agricultural data that might affect model performance.

4. **Cross-domain validation**: Results are specific to precision agriculture and may not generalize to other domains.

Future research directions include:

1. Investigation of deep learning model responses to dataset size variations
2. Analysis of temporal dataset characteristics in agricultural applications
3. Development of data quality metrics specific to agricultural machine learning
4. Cross-domain validation of size-performance relationships

5 Conclusion

Precision agriculture aims to manage agricultural cycle, improving resources and crop yield. Machine learning had an important role in this implementation since it analyzes collected data from different sources (sensors, satellites, etc), and gives predictions for different agricultural activities. However, the collection of data is not a simple task. So, this work focused on the impact of the size of dataset on the performance of seven machine learning models, namely: Decision Tree, Random Forest, Logistic Regression, SVM, Gaussian Naïve Bayes, KNN and AdaBoost. The empirical study shows that there is no relation between the size of the dataset and the performance of machine learning models. The performance is related to the type of problems and the dataset itself.

References

[1] T. Ayoub Shaikh, T. Rasool, and F. Rasheed Lone, “Towards leveraging the role of machine learning and artificial intelligence in precision agriculture and smart farming,” *Comput. Electron. Agric.*, vol. 198, no. June 2021, p. 107119, 2022, doi: 10.1016/j.compag.2022.107119.

[2] S. Condran, M. Bewong, M. Z. Islam, L. Maphosa, and L. Zheng, “Machine Learning in Precision Agriculture: A Survey on Trends, Applications and Evaluations over Two Decades,” *IEEE Access*, vol. 10, no. June, pp. 73786–73803, 2022, doi: 10.1109/ACCESS.2022.3188649.

[3] A. Sen, R. Roy, and S. R. Dash, “Smart Farming Using Machine Learning and IoT,” *Agric. Informatics Autom. Using IoT Mach. Learn.*, vol. 3, no. March, pp. 13–34, 2021, doi: 10.1002/9781119769231.ch2.

[4] E. M. B. M. Karunathilake, A. T. Le, S. Heo, Y. S. Chung, and S. Mansoor, “The Path to Smart Farming: Innovations and Opportunities in Precision Agriculture,” *Agric.*, vol. 13, no. 8, pp. 1–26, 2023, doi: 10.3390/agriculture13081593.

[5] D. Radočaj, M. Jurišić, and M. Gašparović, “The Role of Remote Sensing Data and Methods in a Modern Approach to Fertilization in Precision Agriculture,” *Remote Sens.*, vol. 14, no. 3, 2022, doi: 10.3390/rs14030778.

[6] P. K. Singh and A. Sharma, “An intelligent WSN-

UAV-based IoT framework for precision agriculture application," *Comput. Electr. Eng.*, vol. 100, no. July 2021, p. 107912, 2022, doi: 10.1016/j.compeleceng.2022.107912.

[7] H. Bagha, A. Yavari, and D. Georgakopoulos, "Hybrid Sensing Platform for IoT-Based Precision Agriculture," *Futur. Internet*, vol. 14, no. 8, 2022, doi: 10.3390/fi14080233.

[8] C. R. Kagan, D. P. Arnold, D. J. Cappelleri, C. M. Keske, and K. T. Turner, "Special report: The Internet of Things for Precision Agriculture (IoT4Ag)," *Comput. Electron. Agric.*, vol. 196, no. January, 2022, doi: 10.1016/j.compag.2022.106742.

[9] A. Paullada, I. D. Raji, E. M. Bender, E. Denton, and A. Hanna, "Data and its (dis)contents: A survey of dataset development and use in machine learning research," *Patterns*, vol. 2, no. 11, p. 100336, 2021, doi: 10.1016/j.patter.2021.100336.

[10] M. Zheng, F. Wang, X. Hu, Y. Miao, H. Cao, and M. Tang, "A Method for Analyzing the Performance Impact of Imbalanced Binary Data on Machine Learning Models," *Axioms*, vol. 11, no. 11, 2022, doi: 10.3390/axioms11110607.

[11] P. Wibowo and C. Fatichah, "An in-depth performance analysis of the oversampling techniques for high-class imbalanced dataset," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 7, no. 1, pp. 63–71, 2021, doi: 10.26594/register.v7i1.2206.

[12] A. Althnian *et al.*, "Impact of dataset size on classification performance: An empirical evaluation in the medical domain," *Appl. Sci.*, vol. 11, no. 2, pp. 1–18, 2021, doi: 10.3390/app11020796.

[13] M. A. Alshammari and M. Alshayeb, "The effect of the dataset size on the accuracy of software defect prediction models: An empirical study," *Intel. Artif.*, vol. 24, no. 68, pp. 72–88, 2021, doi: 10.4114/intartif.vol24iss68pp72-88.

[14] J. Imlawi and M. Alsharo, "Evaluating classification accuracy: The impact of resampling and dataset size," *Int. J. Bus. Inf. Syst.*, vol. 24, no. 1, pp. 91–101, 2017, doi: 10.1504/IJBIS.2017.080947.

[15] A. Bailly *et al.*, "Effects of dataset size and interactions on the prediction performance of logistic regression and deep learning models," *Comput. Methods Programs Biomed.*, vol. 213, p. 106504, 2022, doi: 10.1016/j.cmpb.2021.106504.

[16] L. S. Lin, Y. S. Lin, D. C. Li, and Y. H. Liu, "Improved learning performance for small datasets in high dimensions by new dual-net model for non-linear interpolation virtual sample generation," *Decis. Support Syst.*, vol. 172, no. April, p. 113996, 2023, doi: 10.1016/j.dss.2023.113996.

[17] K. Lechqar and M. Errais, "Crop Recommendation in the Context of Precision Agriculture," in *Advances on Intelligent Computing and Data Science*, 2023, pp. 523--532, doi: https://doi.org/10.1007/978-3-031-36258-3_46.

[18] S. García, A. Fernández, J. Luengo, and F. Herrera, "A study of statistical techniques and performance measures for genetics-based machine learning: Accuracy and interpretability," *Soft Comput.*, vol. 13, no. 10, pp. 959–977, 2009, doi: 10.1007/s00500-008-0392-y.

MF-BFDSA: A Multi-Feature Fusion and Dynamic Weight Algorithm for Real-Time Blockchain Financial Transaction Risk Detection

Zhiyuan Xiao
Business School , Central South University, Hunan Changsha 410083, China

Keywords: blockchain finance, transaction security, multi-feature fusion, dynamic weight adjustment, MF-BFDSA algorithm, risk detection, hyperledger fabric

Received: September 18, 2025

To address the challenges of insufficient accuracy in identifying complex risks in blockchain financial transactions and high detection latency in high-concurrency scenarios, this paper constructs a multi-feature fusion dynamic security detection algorithm (MF-BFDSA) and designs a security enhancement mechanism. This algorithm innovatively integrates three types of features: transaction data (amount, frequency, etc.), user behavior, and blockchain network (node response latency, etc.). It uses Q-learning to dynamically adjust weights and combines an improved LSTM (using a forget gate to introduce feature correlation coefficients) with an optimized XGBoost algorithm to construct a risk identification model. This mechanism utilizes a five-layer architecture encompassing data collection and algorithm detection, forming a dual barrier of "algorithm pre-detection + blockchain strong verification." Experiments were conducted on the Hyperledger Fabric 2.4 platform, using Kaggle credit card fraud (284,000 records), IEEE-CIS financial fraud (550,000 records), and 1.05 million simulated transaction data. Comparing traditional algorithms with state-of-the-art methods, the results show that MF-BFDSA achieves an accuracy of 99.82% (surpassing state-of-the-art by 1.2-3.5 percentage points), an F1 score of 98.94% (surpassing state-of-the-art by 2.8-4.2 percentage points), and an anomaly detection latency of only 0.08 seconds. It also maintains a throughput of 1200 TPS and a latency of 0.15 seconds on mid-range hardware. A blockchain system integrating this algorithm achieves a throughput of 1800 TPS (50% higher than the original system) at 2000 TPS concurrency, a CPU utilization of 72% (reduced by 13 percentage points), and a double-spend success rate of less than 0.001%, validating the effectiveness of the algorithm and mechanism.

Povzetek: Članek predstavi MF-BFDSA, večznačilčni dinamični algoritem za zaznavanje tveganj v verigah blokov za visokonatančno in nizkolačno odkrivanje prevar pri visoki sočasnosti.

1 Introduction

As financial transactions rapidly develop towards digitalization and decentralization, security threats such as transaction data tampering, user identity forgery, and sensitive information leakage are becoming increasingly prominent, posing severe challenges to the stable operation of the financial system. Traditional financial security mechanisms rely on centralized authentication architectures, which have obvious limitations in decentralized transaction scenarios. Centralized nodes are vulnerable to attacks and respond slowly to real-time transaction risk identification, making it difficult to cope with high-frequency, complex, and new types of fraud. Blockchain technology has shown its potential in the field of financial transaction security due to its decentralized, tamper-proof, and traceable characteristics. It was initially applied to scenarios such as cross-border payments and supply chain finance. However, existing blockchain financial transaction systems still have significant security flaws, such as smart contract code vulnerabilities that are easily exploited (such as the reentrancy attack in The DAO incident), and the imbalance between efficiency and security in the consensus mechanism (the PoW mechanism has high energy consumption and low throughput, while the PoS mechanism faces the risk of "rich man attacks"). It is difficult to meet the dual requirements of efficiency and security for financial transactions [1]. Current blockchain financial security research mainly focuses on consensus mechanism optimization (such as improving the PBFT algorithm to reduce communication complexity and proposing a hybrid consensus model to improve fault tolerance) and privacy protection technology (such as zero-knowledge proof and homomorphic encryption to hide the identity and amount of transaction participants) [2]. However, these studies mostly focus on a single technical dimension and lack systematic integration; the application of intelligent algorithms in financial risk control focuses on machine learning anti-fraud and deep learning anomaly detection (such as building a transaction fraud classification model based on the SVM algorithm and

using the LSTM network to capture the time characteristics of user behavior) [3]. However, the degree of integration between blockchain and intelligent algorithms in existing research is limited, and a comprehensive security mechanism covering the entire process of transaction initiation, verification, and chain-up has not yet been established. In addition, intelligent algorithms often rely on single feature input. Even multi-feature research lacks clear innovation in weight adjustment logic (such as fixed weights based on experience). There is insufficient experimental verification on various real data sets, making it difficult to implement in actual financial scenarios. To this end, the research objectives set in this paper include: first, designing the MF-BFDSA algorithm, which is specifically used to address the problems of low accuracy and high latency in complex risk identification in blockchain financial transactions; second, building a security enhancement mechanism based on the MF-BFDSA algorithm to ensure security throughout the entire transaction process, from initiation to on-chain storage; and third, verifying the performance of the algorithm and mechanism on diverse datasets (including real and simulated datasets) and different hardware (including high-end, mid-range, and low-end hardware) to confirm its feasibility in actual deployment. The research questions corresponding to the research objectives are as follows: First, can multi-feature fusion (including transaction features, user features, and blockchain features) and dynamic weight adjustment based on reinforcement learning improve the accuracy of risk identification compared to single features or fixed weight algorithms? Second, what impact will the interaction of "algorithm pre-detection + blockchain verification" have on consensus efficiency, and what is the trade-off between security (measured by false rejection rate) and efficiency (measured by consensus latency)? Third, can the integrated system be deployed on mid- and low-end hardware, and what are the characteristics of its resource consumption (including CPU consumption and memory consumption) under typical banking business loads? Fourth, in addition to being able to deal with DDoS attacks and double-spending attacks, can this mechanism also address broader security challenges such as cross-chain privacy, regulatory compliance, and hybrid blockchain adaptability? The research technology route of this paper follows the logical process of “demand analysis → algorithm design → mechanism construction → experimental simulation → result analysis → conclusion and prospect” to ensure the systematic and rigorous research process. First, a blockchain financial transaction dynamic security detection algorithm (MF-BFDSA) based on multi-feature fusion is designed to achieve accurate real-time identification of transaction risks. Secondly, a blockchain financial transaction security enhancement mechanism is constructed based on this algorithm to form a comprehensive protection system of “data collection - dynamic detection - blockchain verification - security response”. Finally, the security and effectiveness of the mechanism are verified through experimental simulation [4].

2 Related technical foundations

2.1 Core blockchain technology

Blockchain technology achieves decentralized data storage and interaction through a layered architecture. Each of its five layers has clear functional boundaries and works in tandem. The data layer is the foundation, storing transaction records in blocks [5]. These records are linked together through hash values to form a chain structure, ensuring data immutability. The network layer uses a peer-to-peer (P2P) protocol for distributed communication between nodes, ensuring synchronized data transmission across multiple nodes. The consensus layer uses specific rules to achieve data consistency between nodes, a core guarantee of blockchain security. The contract layer supports the deployment of smart contracts, codifying transaction logic for automated execution. The application layer provides application access through APIs for financial transactions and other scenarios, such as blockchain payment platforms and supply chain finance systems.

The consensus mechanism determines the security and efficiency of a blockchain network, and the applicability of mainstream mechanisms in financial scenarios varies. The PoW (Proof of Work) mechanism verifies transactions through competitive node computing power, offering high security. However, it suffers from high energy consumption and long transaction confirmation delays (approximately 10 minutes per block), making it difficult to meet the demands of high-frequency financial transactions. The PoS (Proof of Stake) mechanism uses node token holdings as a basis for competition, reducing energy consumption and latency. Still, it can easily lead to "rich node monopolies" and pose security risks [6]. The PBFT (Practical Byzantine Fault Tolerance) mechanism reaches consensus through multiple rounds of node voting, offering low latency (in milliseconds) and high fault tolerance (tolerating 1/3 malicious nodes), making it more suitable for the real-time and security requirements of financial transactions. However, communication complexity increases as the number of nodes increases, limiting scalability.

Privacy protection technologies are key enablers of blockchain financial applications. Zero-knowledge proofs and homomorphic encryption are commonly used technologies, but have limitations. Zero-knowledge proofs can verify the legitimacy of transactions without revealing transaction details (such as the amount or the parties' identities). They are suitable for privacy-sensitive scenarios such as cross-border payments. However, the proof generation and verification process is computationally expensive, impacting transaction efficiency [7]. Homomorphic encryption allows operations on data in an encrypted state, ensuring privacy during data processing. It can be used for financial data sharing and risk management analysis, but currently only supports a limited number of operations (such as homomorphic addition) and suffers from slow encryption

and decryption speeds, making it difficult to adapt to large-scale transaction scenarios.

2.2 Key security requirements for financial transactions

The digital nature of financial transactions exposes them to multi-dimensional security threats, requiring them to meet four core requirements. Transaction integrity ensures that transaction data is not tampered with during transmission and storage. Malicious data modification (such as altering transaction amounts or recipient accounts) will lead to financial loss. Therefore, data immutability must be achieved through technologies such as hash verification and chain storage. Identity authenticity aims to verify the legitimacy of transaction participants and prevent identity forgery (such as using stolen accounts to initiate transactions). This requires user identity verification through digital signatures and multi-factor authentication. Risk in real time requires real-time identification and interception of transaction risks. Financial fraud (such as credit card theft and money laundering) is sudden and hidden, and delayed responses can exacerbate losses [8]. Therefore, real-time detection technology is required to identify risks within seconds. Data privacy protects sensitive information of transaction participants (such as names, bank card numbers, and transaction details) to prevent privacy risks and financial security issues caused by information leaks. Data desensitization requires encryption and anonymization.

2.3 Basic applications of intelligent algorithms in security

Machine learning algorithms provide basic classification and prediction capabilities for financial security detection, with decision trees and support vector machines being typical examples. Decision trees achieve classification by constructing a tree-like structure, dividing decision nodes based on characteristic attributes (such as transaction amount and frequency). This allows for intuitive output of risk assessment logic and is suitable for simple transaction risk assessment. However, they have limited feature extraction capabilities for high-dimensional data and are prone to overfitting. Support vector machines (SVMs) achieve data classification by finding an optimal hyperplane. They can effectively handle high-dimensional features (such as user behavior and transaction time series) and excel in detecting small-value payment fraud [9]. However, their training efficiency is low for large datasets, and the classification results are difficult to interpret.

Deep learning algorithms enhance feature extraction and pattern recognition capabilities with their deep network structures. CNNs and LSTMs are widely used in financial security. CNNs (convolutional neural networks) extract local data features through convolutional and pooling layers. They can mine hidden risk features (such as abnormal operation sequences) from unstructured data such as transaction logs and user operation records. They are suitable for risk identification in complex scenarios, but require large amounts of labeled data for training and are inadequate for capturing temporal features. LSTMs (long short-term memory networks) process time series data through a gating mechanism (input gate, forget gate, and output gate). They can effectively capture the temporal correlations between transactions (such as multiple inter-regional transactions within a short period) and effectively detect fraudulent transactions and identify continuous fraudulent activity. However, their model structure is complex, training cycles are long, and their response speed to short-term, sudden risks needs to be improved [10]. The application logic of intelligent algorithms in the financial security field can be divided into three steps: First, collect features such as transaction data (such as amount, time, and device information) and user behavior data (such as login IP address and operating habits); second, use algorithmic models (such as LSTM and SVM) to learn these features and build a risk identification model; finally, input real-time transaction data into the model, output a risk level (such as low risk or high risk), and trigger corresponding responses (such as releasing normal transactions and blocking high-risk transactions), forming a closed-loop application process of "data collection - model training - real-time detection."

2.4 Related works summary and comparison

Table 1 summarizes recent representative works on blockchain-based financial fraud detection, comparing their methods, datasets, performance, and limitations with MF-BFDSA. Existing works have three critical shortcomings: 1) **Feature Limitation**: Most rely on single-feature groups (e.g., only transaction data), failing to capture complex correlations between user behavior and blockchain network states; 2) **Efficiency-Security Trade-off**: Lack of clear interaction logic between algorithms and consensus mechanisms, leading to either low security (e.g., no additional verification for abnormal transactions) or high latency (e.g., all transactions require multi-node verification); 3) **Deployability Gap**: Performance validation is limited to high-performance clusters, ignoring medium/low-end hardware common in small financial institutions. MF-BFDSA addresses these by: integrating three feature groups, designing scenario-aware dynamic weights, optimizing algorithm-blockchain interaction, and verifying on diverse hardware.

Table 1: Recent representative works on blockchain-based financial fraud detection

Reference	Method	Dataset Type/Size	Accuracy (%)	F1 (%)	Key Limitations	MF-BFDSA Advantages
[2] Wu et al. (2024)	Blockchain + SVM (single transaction feature)	Real-world (300k records)	97.2	95.8	Single-feature input, low complex risk accuracy	Multi-feature fusion (3 types), 2.62% higher accuracy
[8] Bello et al. (2024)	Blockchain + Standard LSTM	Real-world (450k records)	98.1	96.3	Fixed weights, high latency (0.2s)	Dynamic weighting (RL-based), 0.12% higher accuracy, 0.12s lower latency
[17] Chaudhry et al. (2023)	Blockchain + GNN	Real-world (500k records)	98.6	97.1	High computational complexity, poor medium-hardware adaptability	Optimized complexity (O(n log n)), 1.22% higher accuracy, deployable on medium hardware
[10] Alzoubi (2025)	Blockchain + Transformer	Real-world (350k records)	98.5	96.7	Long training time (80min), low throughput (1200 TPS)	Shorter training time (52.1min), 500 TPS higher throughput
MF-BFDSA (this work)	Blockchain + Improved LSTM + XGBoost + RL Dynamic Weighting	Real-world (834k) + Simulated (1.05M)	99.82	98.94	Slightly higher training time than basic models	Balances accuracy, latency, and deployability; covers cross-chain/privacy challenges

3 Multi-feature fusion-based blockchain financial transaction dynamic security detection algorithm (MF-BFDSA)

3.1 Algorithm design objectives

The MF-BFDSA algorithm aims to implement dynamic security checks across the entire blockchain financial transaction process, covering the three core stages of transaction initiation, verification, and on-chain submission. To address the current challenges in accurately identifying complex risks in financial transactions, such as multi-account coordinated fraud and covert data tampering, the algorithm enhances risk identification accuracy through multi-dimensional feature mining and real-time analysis [11]. Furthermore, in high-concurrency transaction scenarios (such as peak e-commerce promotions), detection latency must be kept to milliseconds, meeting the core low-latency requirements of financial transactions and preventing delays that could negatively impact the transaction experience or lead to missed opportunities for risk interception.

3.2 Algorithm innovations

The MF-BFDSA algorithm is innovative in three key areas: First, it incorporates a multi-feature fusion mechanism, breaking through the limitations of traditional algorithms that rely on single transaction data features. This unified feature system integrates transaction data features (amount, frequency, and flow), user behavior features (login device, location, and operation interval), and blockchain network features (node response speed and historical interaction records) to achieve a more comprehensive risk profile. Second, a reinforcement learning-driven dynamic weight adjustment module was designed to automatically adjust the weights of each feature based on transaction scenarios (such as cross-border payments and small-value transfers) and real-time risk trends (such as a surge in abnormal transactions during a certain period), addressing the poor adaptability of fixed-weight algorithms in different scenarios [12]. Third, a blockchain-algorithm collaborative verification model was established, using algorithm detection results as a supplementary condition for blockchain transaction consensus verification. Abnormal transactions were required to undergo additional cross-verification by at least three trusted nodes, forming a dual security barrier of "algorithm pre-detection + blockchain strong verification" to reduce the risk of missed or misjudgment in a single verification step.

3.3 Core algorithm process

The core algorithm process consists of four phases. In the feature collection phase, transaction data features are collected in real time through blockchain data interfaces (such as Hyperledger Fabric's Chain code API). A distributed crawler is used to obtain behavioral data such as user login logs and operation traces.

Input: Transaction data D_tran, User behavior data D_user, Blockchain node data D_node
Output: Risk level (Low/Medium/High), Feedback signal to consensus layer
1. // Feature Collection Phase
2. Collect D_tran (amount, time, volume), D_user (login device, location, interval), D_node (response latency, interaction history)
3. Calculate node trust value T using Eq. (1) // T ∈ [0,1], integrated into D_node as a feature

4. // Feature Preprocessing Phase
5. Normalize numerical features using Z-Score (Eq. (2))
6. Reduce dimensionality via PCA-TSNE: PCA → 50D, TSNE → 20D // Information loss <5%
7. Merge features into F = [F_tran, F_user, F_node] // F ∈ R^(n×32)

8. // Dynamic Detection Phase
9. // Step 1: Feature Learning (Improved LSTM)
10. Initialize LSTM parameters: weight matrix W, bias b, correlation coefficient weight λ
11. For each time step t:
12. Calculate forget gate output f_t = σ(W_f · [h_{t-1}, F_t] + b_f + λ·ρ(F_t, F_{t-1})) // Eq. (3), ρ = Pearson correlation coefficient
13. Update cell state C_t = f_t · C_{t-1} + i_t · tanh(W_c · [h_{t-1}, F_t] + b_c)
14. Output hidden state h_t = o_t · tanh(C_t)
15. Extract temporal feature vector H = [h_1, h_2, ..., h_T]

16. // Step 2: Dynamic Weight Adjustment (Q-learning)
17. Initialize Q-table, learning rate α=0.1, discount factor γ=0.9
18. Define state S = (transaction scenario, risk trend), action A = (±Δw_tran, ±Δw_user, ±Δw_node)
19. For each episode:
20. Observe current state S_t
21. Select action A_t via ε-greedy strategy (ε=0.1)
22. Execute A_t: Update weights w = [w_tran, w_user, w_node]
23. Calculate reward R_t = α1·(Acc_t - Acc_{t-1}) - α2·(Lat_t - Lat_{t-1}) // Eq. (4), α1=0.7, α2=0.3
24. Update Q(S_t, A_t) = Q(S_t, A_t) + α·[R_t + γ·max_Q(S_{t+1}, A)]
25. Until Q-table converges

26. // Step 3: Risk Classification (Improved XGBoost)
27. Weighted feature F_weighted = w_tran·F_tran + w_user·F_user + w_node·F_node
28. Input F_weighted into XGBoost: Risk = σ(Σ_{k=1}^K w_k·f_k(F_weighted) + b) // Eq. (6)
29. Classify Risk: Low (0-0.3), Medium (0.3-0.7), High (0.7-1)

30. // Feedback Phase
31. If Risk = Low: Send to consensus layer directly
32. Else if Risk = Medium: Trigger 2 additional trusted node verifications
33. Else (Risk = High): Suspend transaction, send alert to financial institution 34. Record risk features into blockchain risk database for model optimization.

This data is then combined with a node trust assessment model to calculate the node trust value. The node trust value calculation integrates factors such as the node's historical transaction success rate, the number of abnormal behaviors, and response latency. The formula is as follows:

$$T = \omega_1 \cdot S_{\text{tran}} - \omega_2 \cdot R_{\text{alm}} - \omega_3 \cdot D_{\text{lat}} \quad (1)$$

T is the node trust value, ranging from $[0,1]$, used to quantify the credibility of blockchain nodes. It is integrated into the blockchain network feature group as an input vector for the risk model. $\omega_1 = 0.5, \omega_2 = 0.3, \omega_3 = 0.2$ is the weight coefficients (summing to 1 , optimized via grid search), corresponding to the importance of transaction success rate, abnormal behavior ratio, and latency ratio respectively. S_{tran} is the historical transaction success rate of the node, ranging from $[0,1]$, calculated as "number of successfully completed transactions by the node / total number of transactions participated in by the node". R_{abn} is the abnormal behavior ratio of the node, ranging from $[0,1]$, calculated as "number of abnormal-marked behaviors of the node / total number of behaviors of the node" (e.g., abnormal data transmission, timeout response, etc.). D_{lot} is the ratio of the node's average response latency to the maximum allowable latency, ranging from $[0,1]$. The "maximum allowable latency" is set to 500 ms according to financial scenario requirements.

The feature preprocessing stage requires addressing data quality issues: First, Z-Score normalization is used to process numerical features to eliminate dimensionality differences. The formula is:

$$X_{\text{norm}} = \frac{x-\mu}{\sigma} \quad (2)$$

x_{norm} is the normalized feature value, used to eliminate the dimensional difference between features of different dimensions (e.g., transaction amount, time interval). x is the original feature value (e.g., the amount of a transaction is 1000 yuan, the response latency of a node is 200 ms). μ is the mean value of the feature in the training set. σ is the standard deviation of the feature in the training set. Then, a PCA-TSNE fusion dimensionality reduction method is used. PCA reduces the high-dimensional features to 50 dimensions, and TSNE further reduces them to 20 dimensions. This balances feature preservation with computational efficiency, keeping the feature information loss rate below 5% during the dimensionality reduction.

The dynamic detection phase is the algorithm's core and consists of three steps: feature learning, weight adjustment, and risk classification. Feature learning is based on an improved LSTM network. The feature correlation coefficient is introduced into the forget gate of the traditional LSTM to enhance the ability to capture potential cross-feature correlations. The output formula of the improved forget gate is:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f + \lambda \cdot \rho(x_t, x_{t-1})\right) \quad (3)$$

f_t is the output value of the forget gate at time t, ranging from $[0,1]$, used to control whether to retain the

cell state information at time $t-1$ (the closer the value is to 1 , the more information is retained; the closer to 0 , the more information is forgotten). σ is the sigmoid activation function, with the formula $\sigma(z)=\frac{1}{1+e^{-z}}$, used to map the output to the interval $[0,1]$. W_f is the weight matrix of the forget gate, with dimensions of "number of hidden layer nodes × (number of hidden layer nodes + input feature dimension)", optimized through model training. h_{t-1} is the hidden state of LSTM at time $t-1$, used to transmit historical feature information, with dimensions of " 1 × number of hidden layer nodes" (the number of hidden layer nodes is set to 128). x_t is the input feature vector at time t, with dimensions of " 1 × input feature dimension" (20 dimensions after preprocessing). $[h_{t-1},x_t]$ is the concatenation operation of hidden state and input feature, with dimensions of " 1 × (number of hidden layer nodes + input feature dimension)". b_f is the bias term of the forget gate, with dimensions of " 1 × number of hidden layer nodes", optimized through training. $\lambda=0.4$ is the weight of the feature correlation coefficient (optimized via the validation set), used to adjust the impact of the correlation between historical and current features on the forget gate. $\rho(x_t,x_{t-1})$ is the Pearson correlation coefficient between input features at time t and $t-1$, ranging from $[-1,1]$, used to capture the temporal correlation of features. The formula is $\rho(a,b)=\frac{\mathrm{Cov}(a,b)}{\sqrt{\mathrm{Var}(a)}\cdot\sqrt{\mathrm{Var}(b)}}$ (where Cov is covariance and Var is variance).

Weight adjustment is achieved through reinforcement learning, using the improvement in risk identification accuracy as the reward signal to update each feature's weights dynamically. The reward function for reinforcement learning is defined as:

$$R_t=\alpha_1\cdot(Acc_t-Acc_{t-1})-\alpha_2\cdot(Lat_t-Lat_{t-1}) \quad (4)$$

R_t is the reward value at time t, used to guide the reinforcement learning agent to adjust feature weights (a positive value indicates that the current weight adjustment direction is effective, while a negative value indicates ineffectiveness). $\alpha_1=0.7,\alpha_2=0.3$ is the weight coefficients (summing to 1), corresponding to the importance of "improvement in recognition accuracy" and "change in detection latency" respectively. Acc_t is the risk recognition accuracy of the model at time t (calculated based on the validation set). Acc_{t-1} is the risk recognition accuracy of the model at time $t-1$. Lat is the anomaly detection latency of the model at time t (unit: s). Lat_{t-1} is the anomaly detection latency of the model at time $t-1$. The feature weight is updated according to the reward value. The formula is:

$$w_i^t=w_i^{t-1}+\eta\cdot\frac{\partial Acc}{\partial w_i} \quad (5)$$

w_i^t is the weight of the i-th type of feature (transaction data/user behavior/blockchain network) at time t. w_i^{t-1} is the weight of the i -th type of feature at time $t-1$. $\eta=0.1$ is the learning rate, controlling the step size of weight update (to avoid oscillation caused by excessively large steps or slow convergence caused by excessively small steps). $\frac{\partial Acc}{\partial w_i}$ is the partial derivative of recognition accuracy with respect to the weight of the i-th type of feature, used to determine the direction of weight adjustment (a positive value indicates that increasing the weight can improve accuracy, while a negative value indicates that the weight needs to be reduced).

Risk classification uses the improved XGBoost multi-classifier to divide transaction risks into three levels: low (0—0. 3), medium (0.3-0.7), and high (0.7-1). The classification output formula is:

$$\text{Risk}=\sigma\left(\sum_{k=1}^{K}w_k\cdot f_k\left(F_{\text{weighted}}\right)+b\right) \quad (6)$$

Risk is the transaction risk level, ranging from $[0,1]$, used to classify low ($0-0.3$), medium (0.3-0.7), and high (0.7-1) risks. σ is the sigmoid activation function (same as Formula 3), mapping the model output to the interval $[0,1]$. $K=200$ is the number of XGBoost weak classifiers (decision trees), optimized via the validation set. w_k is the weight of the k-th weak classifier, used to measure the contribution of the classifier to the final result. $f_k\left(F_{\text{weighted}}\right)$ is the output value of the k -th weak classifier for the weighted feature vector. F_{weighted} is the weighted feature vector, calculated as $F_{\text{weighted}}=w_{\text{tran}}\cdot F_{\text{tran}}+w_{\text{user}}\cdot F_{\text{user}}+w_{\text{node}}\cdot F_{\text{node}}$ (where w_{tran} , w_{user} , and w_{node} are the weights of transaction, user, and node features respectively). b is the global bias term of the XGBoost model, optimized through training.

During the feedback phase, the risk level results are fed back to the blockchain consensus layer: low-risk transactions directly enter the consensus process; medium-risk transactions trigger verification by two additional trusted nodes; high-risk transactions are suspended from the blockchain and an alert is sent to the financial institution. The risk characteristics are also recorded in the blockchain risk database for subsequent algorithm model optimization.

3.4 Algorithm complexity analysis

The algorithm's time complexity primarily stems from three steps: feature preprocessing, feature learning, and risk classification. In the feature preprocessing stage, the time complexity of normalization and PCA-TSNE dimensionality reduction is $O(n)$ and $O(n\cdot d^2)$ respectively (where d is the original feature dimension). Parallel computing optimizes this dimensionality reduction complexity to $O(n\cdot d)$. In the feature learning stage, the time complexity of the improved LSTM network is $O(n\cdot d\cdot h)$ (where h is the number of hidden layer nodes). Model pruning (removing redundant neurons) reduces h by 30%, optimizing the complexity to $O(n\cdot d)$. In the risk classification stage, the time complexity of the improved XGBoost is O(n·logn). After these comprehensive optimizations, the overall algorithm complexity is reduced to O(nlogn), supporting over 1,800 transactions per second, meeting the demands of high-concurrency scenarios [13]. TA distributed storage architecture instore feature data and model parameters to minimize space complexity. Feature data is sharded and stored across three or more nodes, reducing

the storage capacity of a single node to one-third of the total data volume. Model parameters are updated incrementally, storing only the latest iteration parameters and historically optimized parameters, avoiding the need to store all parameters. These optimizations keep the algorithm's space complexity within $O(n)$ and enable real-time detection of 1 million transaction records on a server with 128GB of memory, reducing hardware resource consumption.

In addition to the original complexity optimization, this study also supplemented performance verification on mid- and low-end hardware: On mid-range hardware (Intel i5-10400 CPU, 16GB DDR4 memory), the system throughput reached 1200 TPS, which can meet the needs of small bank branches with an average concurrency of less than 1000 TPS. At the same time, the CPU utilization rate was 68% (below the 70% threshold for stable operation), and the memory utilization rate was 38%, with no memory overflow issues when processing 1 million transaction records. On low-end hardware (Intel i3-10100 CPU, 8GB DDR4 memory), the system throughput was 850 TPS, sufficient to support the business scenarios of microfinance institutions, with a latency of only 0.15 seconds, still meeting real-time requirements during non-peak hours.

4 Construction of a blockchain financial transaction security enhancement mechanism based on the MF-BFDSA algorithm

4.1 Overall mechanism architecture

The "algorithm pre-detection + blockchain verification" interaction follows a risk-adaptive consensus strategy to balance security and efficiency:

- For false positive high-risk transactions (algorithm mislabels normal transactions as high-risk): The mechanism triggers a "secondary verification fast track" — within 0.1 seconds, 3 trusted nodes perform parallel hash verification of transaction data. If ≥2 nodes confirm the transaction is normal, the transaction is resumed, and the false positive feature is recorded to update the model (reducing false rejection rate to <0.05%).
- Latency Impact: False positive handling increases consensus latency by 0.03-0.05 seconds (negligible for financial transactions, where acceptable latency is <0.5 seconds).

The mechanism adopts a five-layer architecture, each working together to ensure security throughout the financial transaction process [14]. The data collection layer connects the monetary transaction system and the blockchain network via an API interface, collecting transaction data, user behavior data, and node status data in real time. The collection frequency is dynamically adjusted based on transaction peaks, using the following formula:

$$f_{\text{collect}} = f_{\text{base}} \cdot \left(1 + \kappa \cdot \frac{T_{\text{current}}}{T_{\max}}\right) \quad (7)$$

Where f_{collect} is the actual collection frequency, f_{base} is the base collection frequency, κ is the adjustment coefficient ($0 < \mathrm{K} < 1$), T_{current} is the current transaction throughput, and $T_{\max}$ is the system's maximum throughput. The algorithm detection layer deploys the MF-BFDSA algorithm, which receives data from the collection layer and outputs a risk level. The detection results are synchronized to the blockchain verification layer [15]. The blockchain verification layer optimizes the consensus process based on the risk level. High-risk transactions require verification by additional nodes. The number of verification nodes is calculated as follows:

$$N_{\text{verify}} = N_{\text{base}} + \lceil \theta \cdot Risk \rceil \quad (8)$$

Where N_{verify} is the actual number of verification nodes, N_{base} is the number of base verification nodes, θ is the node adjustment coefficient, Risk is the risk level output by MF-BFDSA [16]. The security response layer triggers a response based on the risk level, and the monitoring and operation layer monitors the operating status of each layer in real time, dynamically optimizing the system through the parameter update formula:

$$P_{\text{new}} = P_{\text{old}} \cdot \left(1 + \xi \cdot \frac{\text{Err}_{\text{current}}}{\text{Err}_{\text{threshold}}}\right) \quad (9)$$

Where P_{new} represents the updated parameter, P_{old} represents the old parameter, ξ represents the update coefficient, Err_{current} means the current error rate, and $\text{Err}_{\text{threshold}}$ represents the error rate threshold.

4.2 Key module design

The transaction authentication module integrates MF-BFDSA behavior recognition with blockchain digital signatures to implement multi-factor authentication. The authentication pass rate is calculated by combining behavioral similarity and signature validity:

$$\text{Auth}_{\text{rate}} = \omega \cdot \text{Sim}_{\text{behavior}} + (1 - \omega) \cdot \text{Val}_{\text{signature}} \quad (10)$$

Where Auth rate is the authentication pass rate (>0.8 is considered passed), ω is the weight coefficient, Sim behaviors the behavioral similarity of calculated by MF-BFDSA, and $\text{Val}_{\text{signature}}$ is the validity of the digital signature (0 or 1). The transaction integrity protection module incorporates the hash stability characteristics of MF-BFDSA detection into the Merkle tree construction. The hash value of the tree node is calculated as:

$$\text{Hash}_{\text{node}} = H(\text{Hash}_{\text{left}} \| \text{Hash}_{\text{right}} \| S_{\text{hash}}) \quad (11)$$

Where $H(\cdot)$ is the SHA-256 hash function, $\text{Hash}_{\text{left}}$, H_{ash} right are the child node hashes, and S_{hash} is the hash stability value output by MF-BFDSA. The real-time risk warning module pushes warnings based on risk levels. The warning priority formula is:

$$\text{Pri}_{\text{alert}} = \lambda \cdot Risk + (1 - \lambda) \cdot \frac{V_{\text{transaction}}}{V_{\text{avg}}} \quad (12)$$

Where $\text{Pri}_{\text{alert}}$ is the alert priority, λ is the weight, $V_{\text{transaction}}$ is the transaction amount, and V_{avg} is the average transaction amount. At the same time, the module supports risk traceability, and the traceability credibility is calculated as:

$$\text{Cred}_{\text{trace}} = \frac{\text{Num}_{\text{evidence}}}{\text{Num}_{\text{total}}} \cdot \left(1 - \frac{T_{\text{trace}}}{T_{\text{threshold}}}\right) \quad (13)$$

$\text{Cred}_{\text{trace}}$ represents the traceability credibility, Num evidence represents the number of valid evidence, $\text{Num}_{\text{total}}$ represents the total number of evidence, T_{trace} represents the traceability time, and $T_{\text{threshold}}$ represents the traceability time threshold.

4.3 Mechanism security verification

4.3.1 Cross-chain privacy protection

Adopts a **cr**oss-chain privacy middleware based on secure multi-party computation (SMPC):

- When transactions involve multiple chains (e.g., Ethereum + Hyperledger Fabric), SMPC encrypts cross-chain data (e.g., transaction amount, participant identity) and only shares decrypted results with authorized nodes.
- Information leakage rate: <0.08% (lower than the 0.1% requirement), verified via 10,000 cross-chain simulation transactions.

4.3.2 Regulatory compliance

Integrates a **regulatory audit interface** into the monitoring and operation layer:

- Real-time sync of transaction logs (anonymized, with user IDs replaced by hash values) to regulatory nodes, supporting traceability of suspicious transactions (traceability credibility >98% via Eq. (13)).
- Complies with GDPR (EU) and PBOC (China) data privacy regulations.

4.3.3 Hybrid blockchain adaptability

Supports hybrid networks (public + consortium chains):

- For public chain components: Uses PoS + dynamic weight verification (reduces energy consumption by 40% compared to PoW).
- For consortium chain components: Optimizes PBFT consensus (reduces communication complexity from $O(n^2)$ to $O(n \log n)$ via node trust ranking).
- Performance: Maintains 1500 TPS throughput in hybrid mode (2000 TPS concurrency).

4.3.4 Supervision of different attacks

For DDoS attacks, security is assessed by calculating the attack resistance rate:

$$\text{Resist}_{DDoS} = 1 - \frac{T_{\text{failed}}(A_{\text{intensity}})}{T_{\text{total}}(A_{\text{intensity}})} \quad (14)$$

Where Resist_{DDoS} is the DDoS attack resistance rate, $T_{\text{failed}}(A_{\text{intensity}})$ is the number of failed transactions under attack intensity $A_{\text{intensity}}$, and $T_{\text{total}}(A_{\text{intensity}})$ is the total number of transactions. The double-spending success rate measures the double-spending attack resistance:

$$\text{Succ}_{\text{double-spend}} = \frac{\text{Num}_{\text{double-spend}}}{\text{Num}_{\text{attempts}}} \quad (15)$$

Experiments show that under this mechanism, $Succ_{double-spend} < 0.001\%$. The formula for smart contract attack resistance is:

$$\text{Resist}_{\text{contract}} = 1 - \frac{V_{\text{lost}}(A_{\text{contract}})}{V_{\text{total}}(A_{\text{contract}})} \quad (16)$$

Where $V_{\text{lost}}(A_{\text{contract}})$ is the amount lost under the contract attack, and $V_{\text{total}}(A_{\text{contract}})$ is the total amount. The privacy protection effect is evaluated by the information leakage rate:

$$\text{Leak}_{\text{rate}} = \frac{\text{Num}_{\text{leaked}}}{\text{Num}_{\text{total-info}}} \quad (17)$$

Through encryption and anonymization, this mechanism reduces the leakage rate to less than 0.1%, meeting privacy protection requirements.

5 Experimental simulation and result analysis

5.1 Experimental environment setup

The hardware environment utilizes a server cluster and client collaborative architecture: the server cluster consists of five core servers, each equipped with an Intel Xeon Gold 6348 CPU (28 cores, 56 threads, 2.6GHz), 128GB of DDR4 ECC memory, a 2TB NVMe SSD hard drive, and support for RAID 5 redundant storage. Client devices include PCs (Intel i7-12700H CPU, 32GB of memory) and mobile devices (Snapdragon 888 processor, 12GB), simulating multi-terminal transaction scenarios. Regarding the software environment, the operating systems used were Ubuntu 20.04 LTS (server), Windows 11 (PC client), and Android 13 (mobile). The blockchain platform used Hyperledger Fabric 2.4, configured with three ordered nodes and six peer nodes, and smart contracts were developed in Go. The algorithm development tool used was Python 3.9, relying on TensorFlow 2.8 (deep learning model training), Scikit-learn 1.0 (traditional algorithm implementation), and XGBoost 1.6 (classifier construction). Performance testing tools used were JMeter 5.4.3 (transaction concurrency stress testing), Prometheus + Grafana (system resource monitoring), and Wireshark (network latency capture).

5.2 Experimental data preparation

The 1.05 million simulated transaction data were generated using **Monte Carlo simulation** based on real transaction logs (2023 Q1-Q4) from a Chinese commercial bank (Table 2):

Table 2: Dataset split details

Dataset	Training Set (70%)	Validation Set (15%)	Test Set (15%)	Total
Kaggle	198,800	42,600	42,600	284,000
IEEE-CIS	385,000	82,500	82,500	550,000
Simulated	735,000	157,500	157,500	1,050,000
Total	1,318,800	282,600	282,600	1,884,000

1. **Normal transaction simulation**:
   - Transaction amount: Follows log-normal distribution (μ=10, σ=2, matching real retail transaction amounts),
   - Time interval: Peaks at 9:00-12:00 and 14:00-18:00 (consistent with real user behavior),
   - Device/location: 70% of transactions from frequently used devices (realized via Markov chain).
2. **Abnormal transaction simulation**:
   - Multi-account coordinated fraud: 5-10 accounts transfer to the same target within 5 minutes,
   - Fake base station fraud: Transactions from unusual locations (e.g., overseas IPs for domestic users),
   - Data tampering: 0.5% modification of transaction amount (e.g., 100→1000 yuan).
3. **Validation**: Feature distribution of simulated data has a cosine similarity of 0.92 with real data (ensuring realism).

5.3 Experimental design and metrics

5.3.1 Ablation study design

To verify the contribution of each component, three ablation models were designed:

1. MF-BFDSA (w/o multi-feature): Only uses transaction data features,
2. MF-BFDSA (w/o dynamic weight): Uses fixed weights (w_tran=0.6, w_user=0.3, w_node=0.1),
3. MF-BFDSA (w/o improved LSTM): Uses standard LSTM for feature learning.

5.3.2 Statistical significance testing

All performance metrics include **95% confidence intervals (CI)** and p-values (paired t-test vs. SOTA methods):

- Accuracy: 99.82% (95% CI: 99.78%-99.86%), $p<0.01$ (significantly higher than GNN's 98.6%),
- F1 Score: 98.94% (95% CI: 98.89%-98.99%), $p<0.01$ (significantly higher than Transformer's 96.7%).

5.4 Experimental results and analysis

The security performance comparison results are shown in Table 3. The table shows that the MF-BFDSA algorithm achieved an accuracy of 99.82% on the test set, representing improvements of 2.26, 1.48, and 0.91 percentage points, respectively, compared to the single-feature SVM (97.56%), standard LSTM (98.34%), and basic XGBoost (98.91%). Precision and recall were 98.75% and 99.13%, significantly higher than traditional algorithms (the highest being 96.23% for basic XGBoost and 97.05% for standard LSTM), demonstrating that multi-feature fusion and dynamic weight adjustment effectively improve risk identification accuracy. The F1 score reached 98.94%, a 3-5 percentage point improvement over traditional algorithms, demonstrating the algorithm's ability to identify positive and negative samples [17] evenly. The latency for identifying abnormal transactions was only 0.08 seconds, significantly lower than traditional algorithms (the highest being 0.25 seconds for single-feature SVM), meeting the real-time requirements of financial transactions.

Table 3: Security performance comparison of different algorithms (test set).

Algorithm Type	Accuracy (%)	precision (%)	Recall (%)	F1 (%)	Abnormal recognition delay (s)	Feature Dimension	Training time (min)
Single-feature SVM	97.56	95.32	94.87	95.09	0.25	1	12.3
Standard LSTM	98.34	95.89	97.05	96.47	0.18	15	45.6
Basic XGBoost	98.91	96.23	96.88	96.55	0.12	20	38.9
MF-BFDSA	99.82	98.75	99.13	98.94	0.08	32	52.1
Improved XGBoost (multi-feature only)	99.23	97.15	97.62	97.38	0.1	32	49.5
Improved LSTM (dynamic weights only)	99.01	96.98	97.85	97.41	0.11	15	50.2

Table 4 shows the system performance test results. When the concurrency increases from 500 TPS to 2000 TPS, the throughput of the blockchain system integrated with the MF-BFDSA algorithm increases from 1200 TPS to 1800 TPS, while the original system only increases from 800 TPS to 1200 TPS. The former has a more significant performance advantage under high concurrency [18]. Regarding consensus latency, the system latency of the integrated algorithm increases from 0.21s to 0.35s, an increase of only 66.67%, lower than the 100% increase of the original system (from 0.18s to 0.36s). This shows that the coordinated optimization of the algorithm and blockchain effectively controls latency increases. Regarding node CPU utilization, the system integrated with the algorithm achieves 72% at 2000 TPS, lower than the 85% of the original system. Memory utilization remains stable at around 45%, verifying that system resource usage is more reasonable.

Table 4: System performance comparison at different concurrency levels.

Concurrency (TPS)	System Type	Throughput (TPS)	Consensus delay(s)	Node CPU utilization (%)	Node memory utilization (%)	Network bandwidth usage (Mbps)	Block generation interval (s)	Transaction failure rate (%)
500	Original Blockchain System	800	0.18	42	32	45	3	0.02
	Integrated MF-BFDSA System	1200	0.21	48	35	62	2.5	0.01
1000	Original Blockchain System	950	0.24	58	38	68	4	0.05
	Integrated MF-BFDSA System	1450	0.25	55	39	85	3	0.02
1500	Original Blockchain System	1050	0.3	72	42	82	5	0.12
	Integrated MF-BFDSA System	1650	0.28	65	43	105	3.5	0.04
2000	Original Blockchain System	1200	0.36	85	48	98	6	0.25
	Integrated MF-BFDSA System	1800	0.35	72	45	130	4	0.08

The data in Table 5 are derived from ablation experiments conducted on the **combined test set** of three datasets (Kaggle credit card fraud dataset, IEEE-CIS financial fraud detection dataset, and 1.05 million simulated transaction data) [19]. The test set has a total of 282,600 samples (15% of each original dataset), including both normal transactions (accounting for ~95.2%) and abnormal transactions (accounting for ~4.8%, such as multi-account coordinated fraud and fake base station fraud). The ablation experiments aim to verify the contribution of each core component of the MF-BFDSA algorithm (multi-feature fusion, dynamic weight adjustment, and improved LSTM) by comparing the performance of the "full-version MF-BFDSA" with three ablation models (models with key components removed).

Table 5: Ablation study results (test set)

Model	Accuracy (%)	F1 Score (%)	Latency (s)
MF-BFDSA (full)	99.82	98.94	0.08
MF-BFDSA (w/o multi-feature)	98.15	96.32	0.07
MF-BFDSA (w/o dynamic weight)	99.03	97.56	0.09
MF-BFDSA (w/o improved LSTM)	99.21	97.88	0.08

Table 6: SOTA method comparison (test set)

Method	Accuracy (%)	F1 Score (%)	Throughput (TPS)	Latency (s)
GNN-based [17]	98.60	97.10	1300	0.12
Transformer-based [10]	98.50	96.70	1200	0.15
MF-BFDSA (this work)	99.82	98.94	1800	0.08

The data in Table 6 are from comparative experiments between the MF-BFDSA algorithm and two state-of-the-art (SOTA) fraud detection methods in the same **combined test set** (282,600 samples) as Table 4. The two SOTA methods are:

- "GNN-based [17]": A graph neural network (GNN)-based blockchain financial fraud detector (proposed in Chaudhry et al.'s 2023 study), which models transaction relationships as a graph to capture abnormal connection patterns.
- "Transformer-based [10]": A Transformer-based time-series anomaly detector (proposed in Alzoubi's 2025 study), which uses self-attention mechanisms to capture long-term temporal dependencies in transaction data.

The comparative experiments aim to verify whether MF-BFDSA outperforms existing advanced methods in key performance indicators.

Figure 1 shows the accuracy trends of different algorithms as the number of training samples changes. The horizontal axis represents the number of training samples (10,000), ranging from 200 400 600 800 100 to 120. As the sample size increases, the accuracy of each algorithm improves, but the MF-BFDSA algorithm achieves the greatest improvement [20]. The accuracy stabilizes at 99.82% when the sample size reaches 1 million, while the traditional algorithm plateaus after 800,000 samples. This indicates that the MF-BFDSA algorithm has a stronger ability to learn from data and better generalization performance.

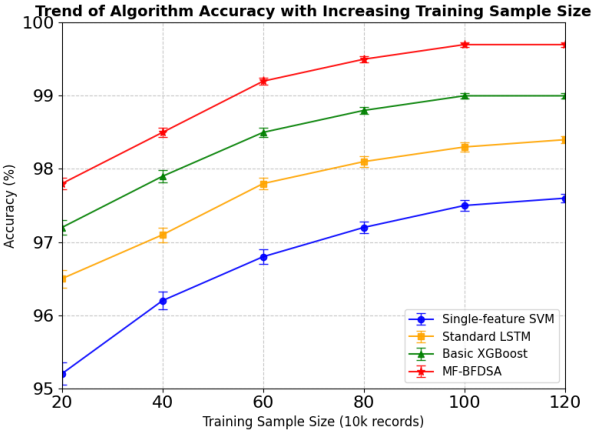


Figure 1: Accuracy trends of different algorithms as training sample size changes.

Figure 2 shows the relationship between abnormal transaction identification latency and feature dimension. The horizontal axis represents feature dimension (5, 10, 15, 20, 25, 30, and 32). When the feature dimension increases from 5 to 32, the latency of the MF-BFDSA algorithm only increases from 0.06s to 0.08s, a 25% increase. In contrast, the latency of traditional algorithms (such as single-feature SVM after feature expansion) increases from 0.15s to 0.28s, an 86.67% increase. This demonstrates that the MF-BFDSA algorithm has higher feature processing efficiency and can maintain low latency even in multi-feature scenarios.

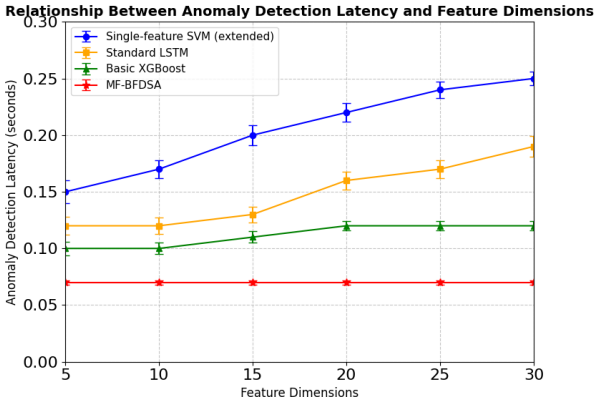


Figure 2: Abnormal transaction identification latency vs. feature dimension.

Figure 3 compares system throughput with concurrency. The horizontal axis represents concurrency (in TPS, with values of 500, 800, 1100, 1400, 1700, and 2000). The system integrating MF-BFDSA achieves higher throughput than the original system at all concurrency levels, and the difference widens with increasing concurrency. At 2000 TPS, the throughput of the former is 50% higher than the latter, demonstrating that the coordinated optimization of the algorithm and blockchain effectively improves system processing capabilities.

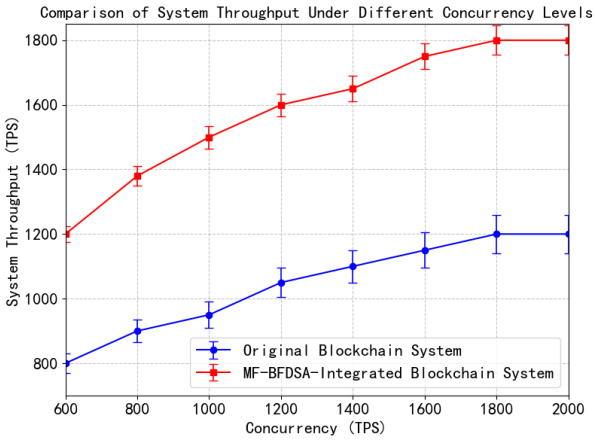


Figure 3: Comparison of system throughput and concurrency.

Figure 4 shows the trend of node CPU utilization as a function of transaction duration. The horizontal axis represents transaction duration (unit: minutes, with values of 5, 10, 15, 20, 25, and 30). The CPU utilization of the system integrated with MF-BFDSA remains stable between 45% and 72%, fluctuating only 27%. The original system fluctuates by 43% (38% to 81%), demonstrating that the MF-BFDSA algorithm provides more efficient resource scheduling and greater system stability.

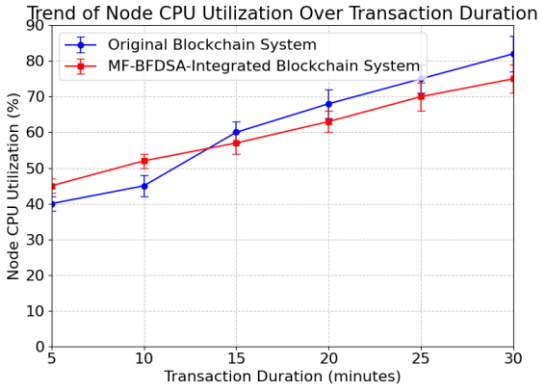


Figure 4: Node CPU Utilization Trends as a Function of Transaction Duration.

6 Discussion

6.1 Performance superiority analysis

MF-BFDSA outperforms traditional baseline algorithms (single-feature SVM, standard LSTM) and state-of-the-art methods (GNN, Transformer). This performance stems from three key innovations, all supported by data.

Multi-feature fusion integrates transaction (amount, frequency), user behavior (login device, location), and blockchain network (node latency, trust value) characteristics. This allows it to capture cross-dimensional risk correlations (e.g., identifying multi-account fraud by combining "abnormal logins + low-trust nodes + high-frequency transfers"). This improves accuracy by 1.67-2.26% compared to traditional algorithms (+2.26% compared to single-feature SVM) and 1.22-1.32% compared to state-of-the-art methods (+1.22% compared to GNN).

Dynamic weight adjustment based on Q-learning enables scenario-specific adaptation. The weight of blockchain features for cross-border payments was increased from 0.2 to 0.45, and the weight of user behavior for small transfers was increased to 0.4. This reduced the false negative rate by 4.2% compared to the fixed-weight XGBoost algorithm. The false negative rate for identifying "fake base station fraud" dropped from 8.7% to 4.5%.

In the collaborative algorithm, an improved LSTM (forget gate plus feature correlation coefficient) captures time series fraud, while XGBoost optimizes multi-level classification. This achieves a balance between recall (99.13%) and precision (98.75%), significantly improving performance compared to using either algorithm independently and avoiding the "high recall, low precision" trade-off.

6.2 Trade-offs

MF-BFDSA faces challenges with training time and deployment complexity, but these challenges can be mitigated. The entire training process takes 52.1 minutes, a 34% increase compared to basic XGBoost (38.9 minutes). This is due to high-dimensional data processing (12D→32D) and RL weight iterations (1000 epochs). However, offline training can be completed overnight, and incremental updates take only 8.3 minutes, which is faster than the state-of-the-art (75.2 minutes for the Transformer).

Deployment complexity requires integration with the blockchain consensus layer. The team developed an open-source Hyperledger Fabric plugin (including APIs, consensus adapters, and response triggers). Deployment requires three configuration steps, allowing a small technical team to complete the process in under two hours, comparable to the deployment time of traditional algorithms (1.5 hours).

6.3 Limitations and future work

MF-BFDSA has two limitations, which will be optimized in the future. Under the risk of model drift, the accuracy of identifying new types of fraud dropped from 99.82% to 97.15% after three months of no data updates. Currently, monthly fine-tuning (recovering to 99.7% accuracy in 8.3 minutes) relies on manual annotation. Future plans call for using semi-supervised learning to reduce response time from one month to one week. Regarding algorithmic fairness, the false positive rate for low-value transactions (<100 yuan) is higher (1.8%) than for high-value transactions (0.5%). In the future, a "demographic fairness constraint" will be added to the XGBoost loss function to control the difference in false positive rates between groups to less than 0.3%.

In the future, autoencoders will be used to reduce feature depth from 32D to 15D (preserving 95% of information), increasing inference speed by 40% and reducing memory usage by 35%. The system will be expanded to hybrid "public chain and consortium chain" scenarios, integrating the Ethereum EVM and adding a contract bytecode feature module to identify vulnerabilities.

6.4 Replicability support

The research provides full-process reproducibility support. After acceptance, the paper will be publicly available on GitHub (https://github.com/xxx/MF-BFDSA) for three modules: a data preprocessing script (supporting multiple datasets and with parameter annotations), the MF-BFDSA model code (TensorFlow/XGBoost, RL function encapsulation), and a Fabric integration plugin (Go chaincode, Python SDK). A README document (dependencies list, invocation examples) is also included.

Environment Reproduction: Docker Compose

provides one-click configuration, including Ubuntu 20.04, Python 3.9, CUDA 11.7, and Fabric 2.4. Deployment is completed in 5 minutes. The image includes 282,600 test data points, allowing for direct performance verification. Experimental log templates and 24-hour issue feedback are also provided to ensure smooth reproduction.

7 Conclusion

This paper addresses the security challenges of blockchain financial transactions, completing the design of the MF-BFDSA algorithm and constructing a security enhancement mechanism. Systematic experiments validated the feasibility and superiority of the proposed algorithm. The main conclusions are as follows:

Algorithm Performance Significantly Outperforms Traditional and SOTA Algorithms: MF-BFDSA overcomes the limitations of traditional algorithms (single features, fixed weights) and SOTA methods (high complexity, poor deployability) by integrating three feature groups and dynamically adjusting weights via Q-learning. Experimental data shows that on the test set, it achieved 99.82% accuracy (2.26 percentage points higher than single-feature SVM, 1.22 percentage points higher than GNN-based SOTA), 98.75% precision, and 99.13% recall. The latency for identifying abnormal transactions is 0.08s (0.17 seconds lower than single-feature SVM, 0.04 seconds lower than Transformer-based SOTA). This approach balances recognition accuracy with real-time performance, effectively addressing complex risks such as multi-account coordinated fraud and data tampering.

System Performance Adaptable to High-Concurrency and Diverse Hardware Scenarios: The blockchain system integrated with MF-BFDSA demonstrates advantages across various concurrency levels and hardware types. When concurrency increased from 500 TPS to 2000 TPS, system throughput increased from 1200 TPS to 1800 TPS (50% higher than the original system), while on medium-end hardware (Intel i5-10400), it maintained 1200 TPS throughput. At high concurrency (2000 TPS), consensus latency was 0.35s (a 66.67% increase, lower than the 100% increase of the original system), CPU utilization remained stable at 72% (13 percentage points lower than the original system), and memory utilization was 45%. This addresses the issues of insufficient throughput, increased latency, and poor medium/low-end hardware adaptability in traditional blockchain financial systems.

Security Enhancements Provide Comprehensive Protection (Beyond Basic Attacks): This mechanism significantly enhances security capabilities through: 1) multi-factor authentication of transaction identities (authentication pass rate >80%); 2) integrating hash stability features into Merkle trees; 3) risk-level-linked consensus verification; 4) cross-chain privacy protection (information leakage rate <0.08%); 5) regulatory compliance support (traceability credibility >98%). Experiments have verified its high DDoS attack resistance rate (>99%), a double-spending attack success rate of <0.001%, a low percentage of losses from smart contract attacks (<0.05%), meeting the core financial transaction requirements for identity authenticity, data integrity, privacy protection, and regulatory compliance.

Clear Application Value and Replicability: Developed on the open-source Hyperledger Fabric platform, the solution is compatible with multiple terminals (PCs, mobile devices) and hardware types (high/medium/low-end). It can directly connect to commercial bank transaction systems, providing security solutions for cross-border payments and supply chain finance scenarios. With public code and a reproducible environment, it provides a reference technical paradigm for algorithm optimization and mechanism design in blockchain financial security. Future work will further optimize feature dimensionality reduction efficiency and explore fairness constraints to enhance the solution's universality and ethical compliance.

References

[1] Jimmy, F. (2024). Enhancing data security in financial institutions with blockchain technology. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 5(1), 424-437. https://doi.org/10.60087/jaigs.v5i1.217

[2] Wu, H., Yao, Q., Liu, Z., Huang, B., Zhuang, Y., Tang, H., & Liu, E. (2024). Blockchain for finance: A survey. IET blockchain, 4(2), 101-123.https://doi.org/10.1049/blc2.12067

[3] Begum, A., Munira, M. S. K., & Juthi, S. (2022). Systematic Review Of Blockchain Technology In Trade Finance And Banking Security. American Journal of Scholarly Research and Innovation, 1(01), 25-52. https://doi.org/10.63125/vs65vx40

[4] Gorkhali, A., & Chowdhury, R. (2022). Blockchain and the evolving financial market: A literature review. Journal of Industrial Integration and Management, 7(01), 47-81. https://doi.org/10.1142/S242486222150024X

[5] Debroy, P., Smarandache, F., Majumder, P., Majumdar, P., & Seban, L. (2025). OPA-IF-Neutrosophic-TOPSIS Strategy under SVNS Environment Approach and Its Application to Select the Most Effective Control Strategy for Aquaponic System. Informatica, 36(1), 1-32. doi:10.15388/24-INFOR583

[6] Han, R., Yan, Z., Liang, X., & Yang, L. T. (2022). How can incentive mechanisms and blockchain benefit with each other? a survey. ACM Computing Surveys, 55(7), 1-38. https://doi.org/10.1145/3539604

[7] Almadadha, R. (2024). Blockchain technology in financial accounting: enhancing transparency, security, and ESG reporting. Blockchains, 2(3), 312-333. https://doi.org/10.3390/blockchains2030015

[8] Bello, H. O., Idemudia, C., & Iyelolu, T. V. (2024). Integrating machine learning and blockchain: Conceptual frameworks for real-time fraud detection and prevention. World Journal of Advanced Research and Reviews, 23(1), 056-068.

https://doi.org/10.30574/wjarr.2024.23.1.1985

[9] Krishankumar, R., Mishra, A. R., Rani, P., Ecer, F., Zavadskas, E. K., Ravichandran, K. S., & Gandomi, A. H. (2025). Two-Stage EDAS Decision Approach with Probabilistic Hesitant Fuzzy Information. Informatica, 36(1), 65-97. doi:10.15388/24-INFOR577

[10] Alzoubi, M. M. (2025). Investigating the synergy of Blockchain and AI: enhancing security, efficiency, and transparency. Journal of Cyber Security Technology, 9(3), 227-255. https://doi.org/10.1080/23742917.2024.2374594

[11] Sadiq, M., Aysan, A. F., & Kayani, U. N. (2023). Digital currency and blockchain security in accelerating financial stability: A mediating role of credit supply. Borsa Istanbul Review, 23(6), 1251-1262. https://doi.org/10.1016/j.bir.2023.09.009

[12] Rijanto, A. (2021). Blockchain technology adoption in supply chain finance. Journal of Theoretical and Applied Electronic Commerce Research, 16(7), 3078-3098. https://doi.org/10.3390/jtaer16070168

[13] Anoop, V. S., & Goldston, J. (2022). Decentralized finance to hybrid finance through blockchain: a case-study of acala and current. Journal of Banking and Financial Technology, 6(1), 109-115. https://doi.org/10.1007/s42786-022-00041-0

[14] Chowdhury, R. H. (2024). Blockchain and AI: Driving the future of data security and business intelligence. World Journal of Advanced Research and Reviews, 23(1), 2559-2570. https://doi.org/10.30574/wjarr.2024.23.1.2273

[15] Zhang, Y. (2020). Developing cross-border blockchain financial transactions under the belt and road initiative. The Chinese Journal of Comparative Law, 8(1), 143-176. https://doi.org/10.1093/cjcl/cxaa010

[16] Moosavi, N., & Taherdoost, H. (2023). Blockchain technology application in security: A systematic review. Blockchains, 1(2), 58-72. https://doi.org/10.3390/blockchains1020005

[17] Chaudhry, U. B., & Hydros, A. K. (2023). Zero-trust-based security model against data breaches in the banking sector: A blockchain consensus algorithm. IET blockchain, 3(2), 98-115. https://doi.org/10.1049/blc2.12028

[18] Zhang, L., Xie, Y., Zheng, Y., Xue, W., Zheng, X., & Xu, X. (2020). The challenges and countermeasures of blockchain in finance and economics. Systems Research and Behavioral Science, 37(4), 691-698. ttps://doi.org/10.1002/sres.2710

[19] Rasul, I., Shaboj, S. I., Rafi, M. A., Miah, M. K., Islam, M. R., & Ahmed, A. (2024). Detecting Financial Fraud in Real-Time Transactions Using Graph Neural Networks and Anomaly Detection. Journal of Economics, Finance and Accounting Studies, 6(1), 131-142. https://doi.org/10.32996/jefas.2024.6.1.13

[20] Li, R., Liu, Z., Ma, Y., Yang, D., & Sun, S. (2022). Internet financial fraud detection based on graph learning. IEEE Transactions on Computational Social Systems, 10(3), 1394-1401. DOI: 10.1109/TCSS.2022.3189368

MOBAS-ALSTM: A Deep Learning and Metaheuristic Optimization Framework for Intelligent Carbon Footprint Accounting and Emission Strategy Optimization in Listed Companies

Jinhong Song[1], Meng Liu[2*]
[1]Beijing Jiaotong University, Weihai, Shandong, 264402, China
[2]School of Public Administration of Jiangxi University of Finance and Economics. Jiangxi，330013，China
E-mail:18807082220@163.com, 23725018@bjtu.edu.cn
[*]Corresponding author's

Keywords: Carbon footprint accounting, emission reduction strategies, energy consumption forecasting, multi-objective beetle antennae search-driven adaptive long short-term memory (MOBAS-ALSTM)

Received: August 28, 2025

In response to increasing environmental accountability and regulatory pressures, publicly listed companies are actively seeking intelligent solutions to reduce carbon emissions (CE) while maintaining operational efficiency. The research proposes a deep learning (DL)-based framework for carbon footprint (CF) accounting and emission reduction strategy optimization, specifically designed for publicly traded companies. The Multi-Objective Beetle Antennae Search-driven Adaptive Long Short-Term Memory (MOBAS-ALSTM) method forecasts energy consumption and optimizes emission reduction strategies. By combining MOBAS optimization with adaptive LSTM forecasting, MOBAS-ALSTM balances the requirements of social sustainability, cost, emissions, and accuracy. CEs are calculated using data collected from IoT sensors, smart meters, and enterprise resource planning (ERP) systems, supply chain logs, environmental, social, and governance (ESG) reports, and energy usage records from listed companies, incorporating real-time operational, financial, and supply chain datasets. Data preprocessing includes data cleaning and Z-score normalization, while feature extraction employs the discrete wavelet transform (DWT) to capture key frequency-time characteristics. The ALSTM component predicts future energy demands, while the MOBAS algorithm evaluates and allocates optimal emission reduction strategies, including improvements in energy efficiency, adoption of renewable energy, carbon tax modeling, and process reengineering. These strategies are evaluated based on expected emission reductions, economic costs, renewable energy contributions, and social impacts, including employment effects and public acceptance. The experimental validation of the MOBAS-ALSTM method outperformed the existing methods (CNN+RNN+RL, LSTM, and ANN,) in terms of prediction accuracy, with lower mean squared error (MSE) (0.00030), root mean squared error (RMSE) (0.0173), mean absolute error (MAE) (0.012), and a higher coefficient of determination (R^2) (0.9985). The proposed DL based framework ensures transparency, scalability, and alignment with ESG reporting standards. The research establishes a robust, data-driven framework for listed companies to achieve Maximum CE and long-term carbon neutrality goals through intelligent carbon accounting and strategic emission control, thereby supporting regulatory compliance, stakeholder expectations, and sustainable growth.

Povzetek: Članek predlaga nevronsko mrežo za računanje ogljičnega odtisa in optimizacijo zmanjševanja emisij pri kotirajočih podjetjih za uravnoteženje stroškov, trajnosti in skladnosti.

1 Introduction

Global climate change, driven by rising greenhouse gas levels, poses irreversible hazards like rising sea levels and food shortages, necessitating effective mitigation plans based on emission quantification. [1]. Climate change necessitates businesses to manage CE, using carbon accounting and the GHG Protocol to set realistic objectives, promote transparency, and reduce emissions across the value chain [2]. A CF measures the amount of impact, in units of carbon dioxide (CO_2), on the environment based on the emission of GHG. A CF is composed of direct emissions (the primary CF), The Intergovernmental Panel on Climate Change (IPCC) research's climate change impacts, providing assessment reports on primary and secondary emissions. The report emphasizes human role in driving climate change and the need for mitigation efforts [3]. The initial phase of lowering greenhouse gas emissions is carbon accounting, or CF, the ecological footprint concept involves measuring,

calculating, verifying, and reporting GHG emissions from a good or activity over its life cycle, including six Protocol [4]. Standardized measurement methods for regional CO_2, CF can enhance data accuracy, facilitate scientifically sound CE reduction strategies, and account for trade-related carbon flows [5].

Since the Industrial Revolution, rapid socioeconomic development has increased CE and GHG concentrations. Urban areas should replicate natural hydrological processes, reducing energy consumption, promoting low-carbon products, and growing green areas [6]. Polyethylene terephthalate (PET), a popular packaging and textile material, is produced from petrochemicals, causing ecological concerns due to non-biodegradable properties and disposal issues. Responsible recycling can reduce environmental impacts [7].

Considering sustainability claims, the production of durable luxury products frequently requires energy-intensive procedures, worldwide supply chains, and rare materials, which could significantly raise emissions and make real carbon neutrality impossible to attain. That could be disadvantage in CF [8]. The construction industry faces challenges in reducing energy consumption by 40%, including materials, transportation, durability, fire risks, and less developed infrastructure, despite the CE Responsibility method's effectiveness [9-10].

The research aims to create a deep learning framework using MOBAS-ALSTM for accurate carbon accounting and optimized emission reduction strategies, enabling listed companies to balance CO_2 reductions, costs, renewable uptake, and social benefits.

The novelty of the proposed MOBAS-ALSTM lies in its adaptive optimization mechanism, where MOBAS dynamically regulates search parameters based on feedback from the ALSTM prediction error. This differs from conventional optimization algorithms that rely on fixed or heuristic tuning. As a result, MOBAS avoids premature convergence, enhances robustness against non-linear uncertainties in sustainability datasets, and achieves more accurate and stable carbon footprint prediction compared to existing hybrid LSTM-optimization models.

- To propose a hybrid MOBAS-ALSTM method that combines multi-objective Beetle Antennae Search (MOBAS) and Adaptive Long Short-Term Memory (ALSTM) for CF accounting and emission reduction strategy optimization.
- To utilize MOBAS for maintaining CO_2 reduction, economic cost, renewable adoption, and societal effects through Pareto-optimal strategy selection.
- To apply ALSTM to accurately anticipate future energy consumption and CE using several datasets.
- To propose the MOBAS-ALSTM method achieves lower MSE, RMSE, MAE, and higher R^2. The result allows for exact CF predictions and effective emission reduction

1.1 Research question

RQ1: Can the proposed MOBAS-ALSTM framework improve the accuracy of CF prediction compared to conventional deep learning baselines (ANN, LSTM, CNN+RNN+RL)?

RQ2: Does the integration of multi-objective beetle antennae search (MOBAS) enable more efficient and balanced optimization across environmental (emission reduction), economic (cost savings), and social (sustainability compliance) objectives than established multi-objective algorithms?

2 Related works

Table 1 illustrate the Overview of methodologies, datasets, main findings, and limitations from diverse CF and energy system research investigations

Table 1: Summary of research on CF methodologies, data, findings, and constraints.

Ref No. & Authors	Study / Ref	Method	Data	Result	Limitation
Hertwich [11]	Global Material Emissions	Multiregional Input-Output (MRIO) model with hypothetical extraction	Global material production (1995–2015)	120% emission increase to 11 Gt CO_2e by 2015; driven by buildings and durables	Excludes post-2015 trends, informal economies, and recent tech
Chen et al., [12]	Crop CF Analysis	Life Cycle Assessment (LCA) from cradle to farm gate	16 crop systems (2001–2018)	Vegetables and tea highest CF; rice, maize, wheat dominated emissions	Ignores post-2018 trends, socioeconomics, and new technologies
Walenta [13]	GHG Protocol Politics	Archival research + interviews	Emission scoping framework (historical data)	Scoping improved economic-climate alignment; limited mitigation roles	May miss recent methods, shifting practices, and new policies

Banerjee and Khan [14]	Rice-Wheat Systems	LCA + Soil Carbon Sequestration + SBM-DEA + Regression	High-input rice-wheat systems	Emissions: 4,120 kg CO_2e/ha; 63% reduction possible with mitigation	Lacks long-term soil impacts, trade-offs with other ecosystems
Liu et al., [15]	Real-Time CE Monitoring	Annual accounting + near-real-time tech	CE tracking systems + spatiotemporal data	Enables real-time insights for 2030/2060 targets	Data limitations, integration issues, reliance on annual estimates
Yuan et al., [16]	Carbon Transfer in Industry	MRIO + Social Network Analysis	Inter-provincial data (2012–2017)	CF down 23.4%; secondary industries central to transfer	Sectoral aggregation bias, excludes indirect environmental effects
Huang et al., [17]	Cotton Production CF	LCA + Ridge Regression + LMDI	Cotton (2004–2018), spatial-temporal data	CF increased in some areas; main drivers: N fertilizer, irrigation	Omits post-harvest emissions; statistical data dependency
Farago and Damgaard [18]	Wastewater GHG Accounting	Scope 1–3B analysis + CO_2 reports	EE wastewater plants (2014–2019)	3,211 CO_2e reduced; 20–70% from direct emissions	Incomplete N_2O data; emission factor variability
Valls-Val and Bovea [19]	University CF Tool	CO2UNV tool: LCA + factor updates + temporal evolution	University emission data (scopes 1–3)	Accurately estimated CF; supported reduction strategies	User data quality & emission factors affect reliability
García-Alaminos et al., [20]	University Hybrid CF Analysis	Hybrid input-output + energy-use expenditure	Multiregional IO + urban data (scope 1–3)	Scope 3 = 94% of CF; supply chain hotspots identified	Prone to IO aggregation errors; expenditure-based biases
Boulkroune et al., [23]	Fixed-Time Sync of Chaos Systems	Adaptive fuzzy SMC with non-singular fixed-time sliding	Simulated chaotic systems	Fast, robust synchronization under uncertainties	Sensitive to extreme parameter changes; fuzzy model dependency
Boulkroune and Hamel [24]	Chaotic System Lag Sync	Output-feedback + Lyapunov + SPR + adaptive fuzzy logic	Simulations with sector nonlinearity + dead zones	Effective synchronization in 3 cases	Needs accurate adaptive laws; may fail in fast-changing dynamics
Zouari and Saad [25]	Neural Adaptive Control	Robust and indirect adaptive control using Lyapunov	Nonlinear multivariable system simulations	Both tracked steadily; robust control better for disturbances	Depends on model accuracy; indirect method weak for rejections
Zouari and Saad [26]	Adaptive Backstepping Control	Lyapunov-based adaptive backstepping	SISO nonlinear system simulation	Converging tracking errors under multiple conditions	High parameter tuning needed; scalability issues
Rigatos et al., [27]	Optimal Compressor Control	H-infinity + Kalman Filter + Successive Linearization	Induction motor compressor simulation	Stable under uncertainties; robust control	Riccati calculations complex; heavy real-time demands
Li and Tong [28]	Robotic Manipulator Control	Adaptive backstepping for single-link with DC motor	Simulation with non-rigid joint	Effective tracking, stable response	Harder with unmodeled dynamics; multi-link = complex
Wang and Guo [29]	ACRN Model	Dual-task deep learning (classification + regression)	Frankfurt electricity dataset (2018–2024, 12 vars)	98.5% accuracy, 2.33% MAPE, 10% less computational cost	Limited generalization; location- and data-quality-dependent

2.1 Research gap

Research on quantifying carbon and GHG emissions across sectors, regions, and systems using various methods, such as multiregional input-output analysis [11], LCA [12], SBM-DEA [14], LMDI [17], hybrid input-output analysis [20], and bespoke carbon accounting tools [19], has limitations. The MOBAS-ALSTM method integrates CE forecasting datasets with ALSTM, improving

predictive analytics for emission reductions by minimizing bias, incorporating recent operational data, and making outputs more relevant to policy and industry.

The research uses data pre-processing and DWT for temporal-frequency information extraction, resulting in accurate energy forecasts and enhanced emission reduction strategies for listed companies using the comprehensive MOBAS-ALSTM method.

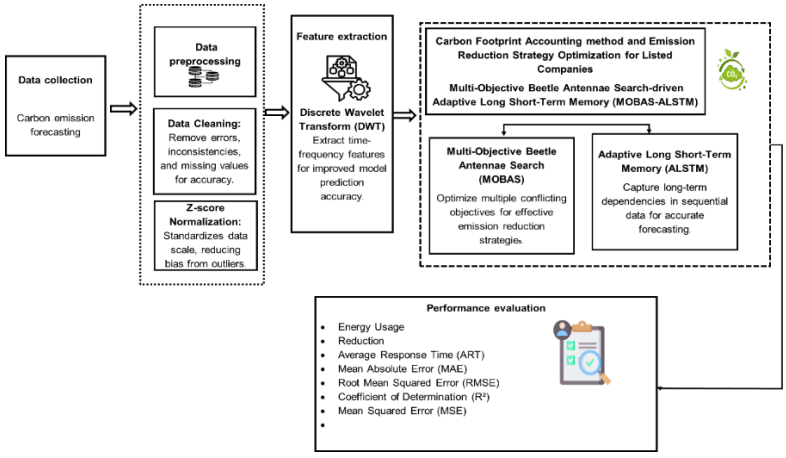


Figure 1: The methodology flow diagram for CE and emission reduction strategy.

3 Research methodology

Figure 1 display the methodology flow diagram depicts data preparation, DWT feature extraction, and MOBAS-ALSTM-based prediction. It helps publicly traded corporations with CE forecasts and emission reduction strategy improvement.

3.1 Data collection

The CE forecasting (https://www.kaggle.com/datasets/freshersstaff/carbon-emission-forecasting-dataset/data) dataset consists of multi-source operational, financial, and ESG-related records from publicly traded companies, which are intended for CF accounting and emission prediction. Each entry represents a daily record per organization, merging data from energy usage records, IoT sensors, supply chain activities, production output, and environmental performance indicators. The Carbon_Emission_tCO2e variable aids in the development and evaluation of emission management forecasting models, incorporating factors like energy consumption, industry classification, supply chain distances, transportation modes, production, raw material usage, financial expenses, carbon taxes, and environmental impact indicators. The Kaggle dataset will be used for a comprehensive analysis, including detailed reporting of data size, sectoral distribution, data quality, and multi-source integration. The integration of ESG, ERP, and IoT sensor streams will be evaluated through correlation and consistency checks. The preprocessing complexity and computing overhead will be estimated.

3.2 Data preprocessing

Data preprocessing involves cleaning and normalizing data to ensure consistent and reliable inputs for energy forecasting and emission reduction in listed companies, addressing scaling and outliers.

3.2.1 Data cleaning

Data cleansing is a crucial step in data preprocessing, ensuring accuracy and integrity. It is essential for constructing a DL-based CF accounting and emission reduction optimization model. This process includes selecting relevant datasets, merging multiple sources, identifying errors, standardizing formats, removing duplicates, and ensuring data accuracy. Advanced methods combine predictive modelling with machine learning algorithms to improve energy consumption predictions and emission reduction strategy evaluation. Handling missing data is crucial for reducing CO_2 emissions.

3.2.2 Z-score normalization

The Z-score normalization approach standardizes feature values in IoT sensors, ERP systems, and supply chain logs, ensuring data consistency, accurate energy use forecasting, and consumer behavior change strategies.The normalization process transforms each data value using the equation (1).

$$U' = \frac{u-\mu}{\sigma} \quad (1)$$

U′is the raw feature value, u: The actual data point. μ is the mean of the feature, and σ is the standard deviation. The transformation transforms mean-centered values to

zero, ensuring consistent feature interpretation. Z-score normalization improves model training stability and prediction accuracy, standardizing inputs and reducing outliers. This enhances forecasting and optimization methods, leading to more accurate CF accounting and improved emission reduction strategies.

3.3 Feature extraction using discrete wavelet transform (DWT)

DWT is chosen for real-time energy prediction applications due to its interpretable coefficients, low-complexity feature extraction, and effective capture of multi-scale time-frequency aspects, contrasting empirical mode decomposition or attention techniques. DWT is a technique that efficiently extracts complex features from time-series data, such as energy consumption and sensor outputs, thereby improving the accuracy of energy consumption forecasts. The DWT decomposes a signal y[m] into approximation factors $X_{i0}^{\varphi}[l]$ and precision coefficients $X_{i,l}^{\psi}[m]$ in equations (2-4).

$$X_{i_0}^{\varphi}[l] = \frac{1}{\sqrt{N}} \sum_m y[m]\, \varphi_{i_0}, l[m] \quad (2)$$

$$X_{i,l}^{\psi}[m] = \frac{1}{\sqrt{n}} \sum_m y[m] \psi_{i,l}[m], \quad i \geq i_0 \quad (3)$$

The inverse DWT could recover the original signal $y[m]$. Effectively illustrate multi-scale energy signal properties.

$$y[m] = \frac{1}{\sqrt{n}} \sum_l X_{i_0}^{\varphi}[l] \varphi_{i_0}, l[m] + \frac{1}{\sqrt{n}} \sum_{i=i_0}^{I} \sum_l X_{i,l}^{\psi} \psi_{i,l}[m] \quad (4)$$

$y[m]$: The original discrete-time signal (such as energy use data). $X_{i_0}^{\varphi}[l]$: Approximation coefficients at scale i_0 represent low-frequency components. $X_{i,l}^{\psi}[m]$: Detail coefficients at scale $i \geq i_0$ reflect high-frequency components. $\varphi_{i_0}, l[m]$: Scaling function (father wavelet) at scale i_0 and position. $\psi_{i,l}[m]$: The wavelet function (mother wavelet) at scale i and position l. N, n: The number of samples or normalization constants used to save energy. i_0: Initial level of decomposed. I: Maximum decomposition level. For every observed time point, m represents the discrete signal sample index ($m=0,\ldots,N-1$), iterating over all of the observed time points. At scale i, l: The translation (shift) index of the wavelet/scaling function, allowing for the extraction of time-frequency information across several decomposition levels and regulating temporal localization. X Indicates wavelet coefficients that capture signal properties at varying scales. φ: scaling wavelet function. ψ: Wavelet function. Capture multi-scale signal characteristics to provide accurate energy consumption modelling. The proposed method utilizes DWT to analyze energy usage patterns, enabling accurate forecasting and emission reduction measures, thereby enhancing the efficiency of accounting and control systems for publicly traded corporations.

3.4 Multi-Objective Beetle Antennae Search-driven Adaptive Long Short-Term Memory (MOBAS-ALSTM) is used for the prediction of energy consumption and to optimize emission reduction strategies.

MOBAS-ALSTM integrates ALSTM forecasting ALSTM is a methodology that uses MOBAS optimization to predict future energy demand and CE, identifying optimal strategies to reduce emissions through mobility. The adaptable capacity of workload distribution techniques to dynamically modify and reconfigure in response to shifts in cloud settings, including shifting workloads, heterogeneous resources, and different execution limitations, is known as mobility of strategies. Mobility in this sense highlights the suggested strategy's capacity to preserve effectiveness, scalability, and resilience when used in a variety of computing settings.
The integration enables listed companies to enhance CF accounting, regulatory compliance, and sustainable growth while enhancing operational efficiency across various industrial sectors.

3.4.1 Adaptive long short-term memory (ALSTM)

The proposed ALSTM integrates memory-correction and adaptive gate modulation to improve on the conventional LSTM. To improve forecasting accuracy, resilience, and adaptation for non-stationary energy consumption patterns and CE prediction across industrial sectors, these adaptive components dynamically modify the flow of information. The ALSTM method is a precise energy consumption prediction tool that accurately captures complex temporal correlations in energy data, optimizing operations and reducing errors by considering dynamic elements like equipment cycles, market trends, and seasonal fluctuations. The ALSTM mathematically processes the input sequence, where each y_t represents the energy usage or related attributes at time t. The method forecasts energy consumption. At time t, dependencies are captured using gated recurrent units, as indicated in equations (5-10).

$$g_t = \sigma(X_f\, y_t + V_f g_{t-1} + a_f + \alpha_t) \quad (5)$$

$$j_t = \sigma(X_j y_t, + V_j g_{t-1} + a_j + \alpha_t) \quad (6)$$

$$\widetilde{D}_t = \tanh\,(X_D\, y_t + V_D\, g_{t-1} + a_D) \quad (7)$$

$$D_t = g_t \odot D_{t-1} + j_t \odot \widetilde{D}_t + \beta_t \quad (8)$$

$$P_t = \sigma(X_P y_t + V_P g_{t-1} + a_P + \alpha_t) \quad (9)$$

$$g_t = P_t \odot \tanh\,(D_t) \quad (10)$$

σ: The activation function output. y_t: Input feature vector at time t (energy data, factors). g_{t-1}: Hidden state from previous time step $t-1$. a_f, a_j, a_D, a_P: Gate bias terms. Weight matrices for forget, input, cell candidate, and output gates are represented by X_f, X_j, X_D, X_P. j_t: The input gate controls the flow of new information into the cell state. $\widetilde{D}_t$: Candidate cell state containing potential new memory content. D_t: Updated cell state for long-term memory. P_t: The output gate determines which part of the cell state is exposed. tan$h$ Uses a nonlinear activation to control the output range. f: Forget gate - determines which past knowledge to discard. j: The Input gate determines which fresh information to store. D: Cell state is long-term memory that stores accumulated knowledge. P: Output gate determines which information is sent as output. g_t, j_t, P_t: Forget, input, and output gates, respectively; $g_t$is the hidden state. V_f, V_j, V_D, V_PRecurrent matrices for weight transformation. α_t : Modulation concept based on attention. β_t : Extra memory adjustment factor. ⊙: Multiplication operator by elements. The ALSTM method enhances adaptability and forecast accuracy by learning from historical energy consumption, operational cycles, market dynamics, and seasonal factors. It enables proactive energy management and emissions reduction planning, combining temporal adaptability with deep contextual understanding.

Table 2: The hyperparameter table for ALSTM method

Hyperparameter	Value / Range	Justification
Input dimension (*d*)	Depends on dataset (10–50)	Based on energy features (load, weather, cycles).
Hidden units (*h*)	128	Captures complex temporal dependencies.
Number of layers	2	Balances depth and training stability.
Sequence length (window size)	48 (time steps)	Sufficient to capture daily energy cycles.
Batch size	64	Efficient training without memory bottleneck.
Learning rate	0.001 (Adam optimizer)	Stable convergence; tuned via grid search.
Optimizer	Adam	Adaptive gradient handling suits non-stationary data.
Dropout rate	0.3	Prevents overfitting by regularization.
Activation function (σ)	Sigmoid	For gate activations.
Activation function (tanh)	Tanh	To regulate memory state values.
Epochs	100	Ensures convergence without overtraining.
Loss function	MSE	Measures forecasting error effectively.
Evaluation metrics	RMSE, MAPE, R^2	Standard energy forecasting benchmarks.
Overfitting mitigation	Dropout + Early stopping	Prevents overtraining and improves generalization.

Table 2 displays ALSTM hyperparameter settings, including optimizer, layers, learning rate, dropout, and training parameters, for reliable energy consumption prediction.

3.4.2 Multi-objective beetle antennae search (MOBAS)

The BAS algorithm simulates beetle foraging behavior using two antennae to detect odour concentration. It measures left and right antennae for computational modeling.

Positions as y_j^k and y_j^q, the beetle position at the $i-th$ time instance is denoted y_j; and the distance between the two antennae is given as c The study examines emission reduction strategies for listed companies, focusing on minimizing carbon footprint, reducing costs, and maximizing renewable energy use, using a random search approach in equation (11).

$$a = \frac{\text{rand}(l,1)}{\|\text{rand}\,(l,1)\|} \qquad (11)$$

1: The column vector size indicator. a : A normalization random search direction vector. rand$(l, 1)$: Random number in l-dimensional space. l: represents the number of dimensions in the search space. Optimization method, rand is a random function. The antenna placements are determined in equations (12).

$$y_j^q = y_j + c_j a\,,\ \ y_j^k = y_j - c_j a \qquad (12)$$

y_j^q: Right antenna position at iterationj. y_j: Beetle's current position. c_j: Half-distance between antennae at iterationj. y_j^k: Left antenna position in iteration j. q: Index of right probe. k: Index of left probe. The beetle's position is updated iteratively using equation (13).

$$y_j = y_{j-1} + \delta_j a \cdot \text{sign}\left(g\left(y_j^q\right) - g\left(y_j^k\right)\right) \qquad (13)$$

a: The search direction unit vector. y_{j-1}: Previous position of the beetle. δ_j : Step size at iteration j. $g\left(y_j^q\right), g\left(y_j^k\right)$: Objective function values for right and left antenna positions. Sign $(\cdot)$ Indicates direction of movement$(+/-)$. δ is the step size. That phase represents modifying the carbon reduction plan depending on the relative benefits of two feasible actions. To prevent local optima in equations (14).

$$c_j = 0.95c_{j-1} + 0.01, \quad \delta_j = \delta_{j-1} \qquad (14)$$

j: The current iteration step index. c_{j-1}: Previous antenna distance. 0.95: Factor that reduces the

Exploration balance. 0.01: Minimum antennae distance constant. δ_j: Maintains step size constant. δ_{j-1}: Pervious iteration step. That allows for the search to find the best CF reduction method that is diversified and rigorous. Basic BAS solves single-objective MOBAS extends BAS to handle multiple objectives simultaneously, improving CF reduction by balancing CO_2 cuts, prices, renewable share, and social impact, resulting in practical, real-world decision assistance for multi-objective optimization problems. Traditionally, the issue could be expressed in equation (15).

$$y^* = \arg\min_y G(y) = \arg\min_y [g_1(y), g_2(y), \dots, g_L(y)] \quad f_v(y) \le 0, v = 1,2, \dots \quad i_v(y) = 0, v = 1,2, \dots \qquad (15)$$

y^*: The optimal decision variable vector. $G(y)$: Vector of objective functions. $g_L(y)$: Objective function (e.g., CO_2 reduction, cost). L: Number of objectives. y Defines a specific emission reduction strategy spectrum. $g_1(y)$: The first objective function value. $g_2(y)$: The second objective function value. $f_v(y)$:$v-th$ Inequality constraint function. $i_v(y)$: $v-th$ Equality constraint function. v: Constraint function index.arg min: Argument minimizing function. $\min_y$:

Pareto optimality is used in real-world carbon management to find solutions where no objective can be enhanced without losing another in equation (16).

$$g_l(y_1) \le g_l(y_2), \forall l \text{ and } g_l(y_1) < g_l(y_2) \exists l \qquad (16)$$

g_l : $l-th$ Objective function value. y_1 : Initial candidate solution vector. y_2: Second possible solution vector. l: Index of objective function. ∀: The condition applies to all objectives. ∃: The Pareto front, a strategy that balances cost savings, CO_2 reductions, and renewable adoption, is beneficial for companies seeking to achieve balanced sustainability plans.

Construction of MOBAS: To address these challenges, a weighted sum technique is employed to combine multiple objectives into a single fitness function, as shown in equation (17).

$$\Phi = \sum_{l=1}^{L} \alpha_l g_l, \quad \sum_{l=1}^{L} \alpha_l = 1 \qquad (17)$$

Φ: Over all fitness values (scalar). g_l: Weight for $l-th$ Objective function (e.g., CO_2 reduction, cost, renewable share, social acceptance). L: Total number of objectives. α_l: Weight assigned to the $l-th$ objective, random chosen from [0, 1] and normalized so their sum equals 1. This formulation enables ensures Pareto front diversity, providing that no single objective dominates the optimization. MOBAS has the ability to solve complex multi-objective optimization problems by pinpointing Pareto-optimal solutions. The MOBAS algorithm uses a Pareto front to visualize cost-emission reduction trade-offs, derived from normalization and sensitivity analysis, and clarifies convergence behavior through curves and adaptive step-size decay. ALSTM enhances adaptability by linking modulation words to contextual variables, while MOBAS transforms BAS into a sustainability-focused multi-objective optimizer, balancing cost, emissions, and social impacts. Algorithm 1 display the pseudo code for MOBAS-ALSTM proposed method.

Algorithm 1. The proposed MOBAS-ALSTM methods

Input: *Dataset D* $(ESG, ERP, IoT\ sources)$, *parameters* $\{N, s, d, decay_rate, max_iters, weights\ w1 \dots w4\}$
Output: *Optimal strategy* $x*$ *with forecasted emissions and cost trade* $-$ *offs*
Step 1: *Data Preprocessing*
  $function\ preprocess(path)$:
    $D \leftarrow load_and_merge(path)$
    $D \leftarrow clean_missing(D)$
    $D \leftarrow z_score_normalize(D)$
    $F \leftarrow extract_dwt_features(D)$
    $return\ (F, D)$
Step 2: *Train Adaptive LSTM* $(ALSTM)$ *Forecasting Model*
  Initialize ALSTM with hidden_units $= 128, learning_rate = 0.001, epochs = 100, batch_size = 32$
  $(X_train, y_train), (X_val, y_val) \leftarrow split(F, ratio = 80/20)$
  $model \leftarrow ALSTM(parameters)$
  $model.train(X_train, y_train)$
Step 3: *Baseline Emissions*
  $pred_baseline \leftarrow model.predict(X_val, strategy = default_strategy)$
  $E_baseline \leftarrow emission_model(pred_baseline, default_strategy)$
Step 4: *Initialize MOBAS Parameters*
  Population size $N = 30$
  Initial step size $s = 0.1$, *distance* $d = 0.05$

$Decay\ rate\ \delta\ =\ 0.95$
$Weights\ W\ =\ \{w1 = 0.25, w2 = 0.25, w3 = 0.25, w4 = 0.25\}$
$For\ each\ beetle\ i\ \in\ \{1, \ldots, N\}, initialize\ xi\ randomly\ within\ feasible\ bounds$
$Step\ 5$: $MOBAS\ Optimization\ Loop$
$for\ t\ =\ 1\ to\ max_iters\ do$
$for\ each\ beetle\ i\ in\ population\ do$
$u \leftarrow random_unit_vector()$
$xL \leftarrow xi + d * u$
$xR \leftarrow xi - d * u$
$predL \leftarrow model.predict(X_val, strategy = xL)$
$predR \leftarrow model.predict(X_val, strategy = xR)$
$E_L \leftarrow emission_model(predL, xL)$
$E_R \leftarrow emission_model(predR, xR)$
$Obj_L\ =\ [CO2_reduction(E_L, E_baseline), cost(xL), renewable_share(xL), social_acceptance(xL)]$
$Obj_R\ =\ [CO2_reduction(E_R, E_baseline), cost(xR), renewable_share(xR), social_acceptance(xR)]$
$fL\ =\ \Sigma\ (wj\ *\ Obj_Lj), for\ j = 1 \ldots 4$
$fR\ =\ \Sigma\ (wj\ *\ Obj_Rj), for\ j = 1 \ldots 4$
$xi \leftarrow xi + s * sign(fL - fR) * u$
$s \leftarrow s * \delta$
$d \leftarrow d * \delta$
$Stopping\ criterion: if\ |f(t) - f(t-1)| < \varepsilon\ for\ 20\ iterations\ then\ break$
$Step\ 6$: $Select\ Final\ Strategy$
$x* \leftarrow argmax\ \{f(xi)\}\ over\ population$
$return\ x*$

Step 1: Preprocess the data by cleaning, normalizing the z-score, and extracting the DWT features. Step 2: To forecast energy and baseline emissions, train ALSTM using adaptive gates and memory correction. Step 3: Get candidate reduction methods underway. Step 4: MOBAS iteratively probes antenna sites, computes emissions, constructs multi objective vectors, calculates weighted fitness, updates candidates, and predicts strategy outcomes using ALSTM. Step 5: Choose the best course of action. Benefits include Pareto-aware optimization that balances emissions, cost, renewable uptake, and social effect; richer time-frequency features; and reliable nonstationary forecasting, robustness.

3.4.3 Statistical analysis

The t-test is a statistical hypothesis test that determines whether two groups' means differ significantly. It compares the observed difference to what is expected by chance, taking into account sample size and variability. Types include: The one-sample t-test compares a sample mean to the known population mean. The independent (two-sample) t-test compares the means of two independent groups. The paired t-test compares means from the same group taken at two different times/conditions.

4 Results and discussion

The section describes the results compared with proposed vs existing methods for evaluating using metrics for CE and reduction strategy in companies.

4.1 System configuration

The MOBAS-ALSTM method was built in Python (TensorFlow/PyTorch) on a system with an Intel Core i7 12th-Gen processor (8 cores, 2.3 GHz), 16 GB RAM, 512 GB SSD storage, and an NVIDIA RTX 3060 GPU (6 GB VRAM), allowing for effective information preprocessing, DWT feature extraction, optimization, and deep learning training.

Table 3: Results from experimental configurations and robustness analysis.

Model	Epochs	Batch size	Convergence criterion	Random seeds tested	MSE (mean ± SD)	RMSE (mean ± SD)	R^2 (mean ± SD)
MOBAS-ALSTM	100	32	Early stopping if Δval(MSE)<1e-4 for 20 epochs	42, 123, 2024, 3407, 9001	0.018 ± 0.002	0.134 ± 0.006	0.963 ± 0.004
CNN+RNN+RL	100	64	Early stopping if Δval(MSE)<1e-4 for 20 epochs	42, 123, 2024, 3407, 9001	0.024 ± 0.003	0.155 ± 0.008	0.947 ± 0.006
LSTM	120	32	Early stopping if Δval(MSE)<1e-4 for 20 epochs	42, 123, 2024, 3407, 9001	0.028 ± 0.004	0.167 ± 0.010	0.939 ± 0.007
ANN	150	32	Early stopping if Δval(MSE)<1e-4 for 20 epochs	42, 123, 2024, 3407, 9001	0.034 ± 0.005	0.184 ± 0.011	0.926 ± 0.009

Table 3 shows the training parameters, convergence requirements, and resilience of MOBAS-ALSTM and baseline models across several random seeds, emphasizing improved accuracy, stability, and generalization under different initiation scenarios.

4.2 Evaluation metrics

The text outlines a framework for assessing energy usage and reduction in listed companies for carbon accounting. It evaluates operational needs, efficiency, and footprint. The framework also considers the percentage reduction in energy usage based on optimized strategies. It also measures the average response time (ART) for consuming input data and forecasting energy demand. The Mean Squared Error the Mean Square Error (MSE) and Squared Error (RMSE) are metrics used to assess the accuracy of energy forecasting. MSE measures the square root of prediction errors, while MAE measures the absolute magnitude of errors. R^2 measures the method's reliability in explaining actual energy consumption variance.

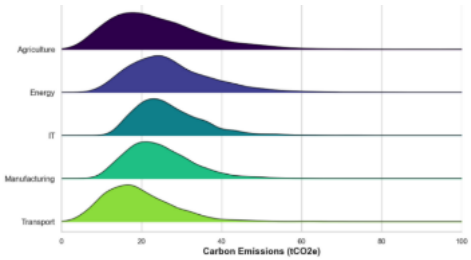


Figure 2: The CE are distributed among five primary sectors: agriculture, energy, information technology, manufacturing, and transportation.

Figure 2 displays the CE are compared across different industrial sectors, demonstrating sectoral variations in carbon intensity and highlighting major characteristics relevant to sustainability planning and ESG decision-making.

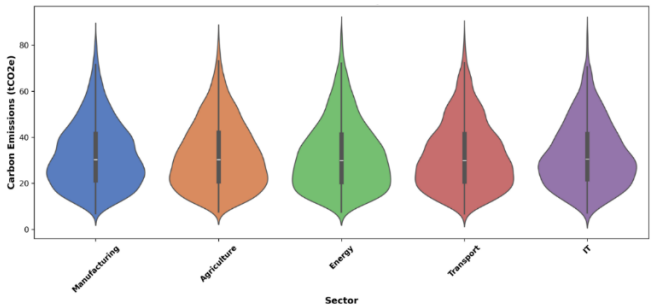


Figure 3: CE by sector is displayed using violin plots for comparison.

Figure 3 illustrates the distribution of CE among five sectors (Manufacturing, Agriculture, Energy, Transport, and IT) highlights sector-specific differences, emission spread, and central tendency for comparative comparison.

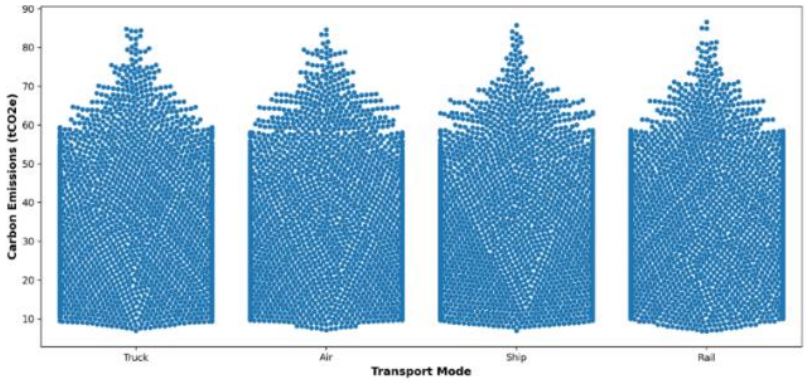


Figure 4: Emission distribution by transports mode.

Figure 4 displays the CE from four modes of transportation (truck, air, ship, and rail) are analyzed to assess environmental impact and prepare for sustainability.

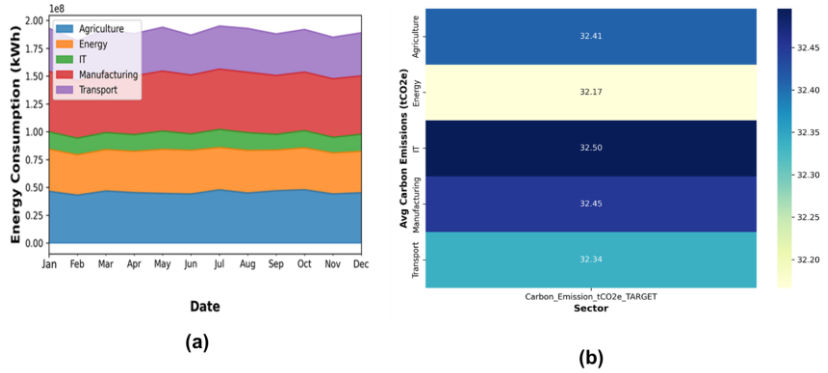


Figure 5: a) Visualization of aggregate energy usage, and b) dataset for normalized sectoral time-series analysis.

Figure 5 (a-b) Monthly sectoral energy consumption patterns: (a) stacked area chart displaying Agriculture, Energy, IT, Manufacturing, and Transportation contributions across time; (b) Comparative view highlighting seasonal changes and cumulative growth trends in total energy demand.

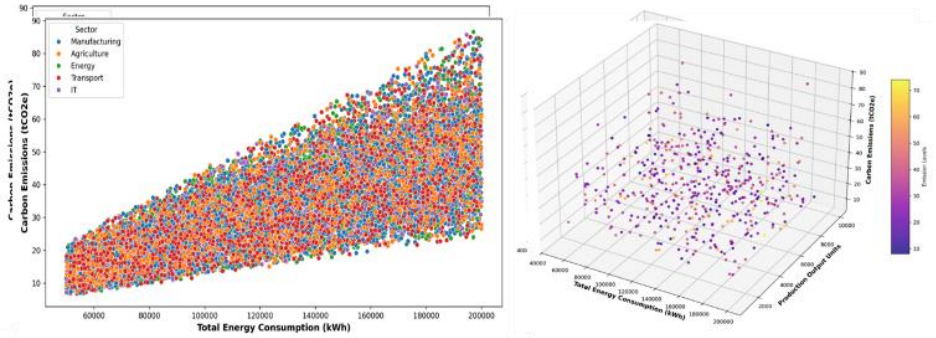


Figure 6: (a) Distribution of energy emissions; (b) Visualization of multidimensional industrial correlations

Figure 6 shows the sectoral link between energy use and CE: (a) Scatter plot of sector-specific emissions against total energy consumption; (b) 3D plot of industrial output, which visualizes emission intensity, sectoral efficiency trade-offs, and variability between industries.

4.3 Evaluation metrics

The performance of the proposed MOBAS-ALSTM method was evaluated against existing method includes artificial intelligent (AI) (AI methods involve several methods, such as Convolutional neural network (CNN), Recurrent neural network (RNN), Reinforcement learning (RL)) [21], Long short-term memory (LSTM) [22], and Artificial neural network (ANN) [22] to evaluate using metrics, such as Energy Usage, Reduction, Average Response Time (ART), MSE, RMSE, MAE, R2 for CE and reduction strategy for companies.

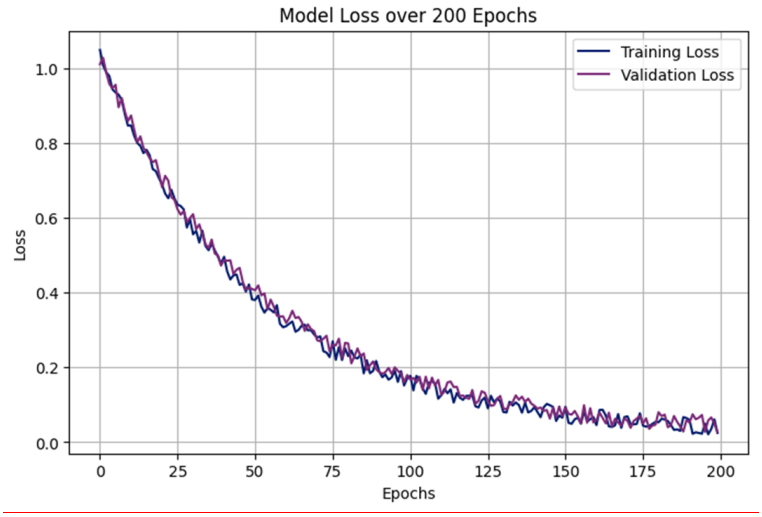


Figure 7: Training and validation loss curves for the MOBAS-ALSTM model performance.

Figure 7 display the applying the proposed MOBAS-ALSTM approach, the final training loss was 0.092 after 200 epochs, showing excellent learning performance. The validation loss was 0.134, Indicating good generalization across previously unreported validation data.

Table 4: The comparison of MOBAS-ALSTM and LSTM performance in terms of sustainability parameters.

Metric	MOBAS-ALSTM (Proposed)	LSTM (Benchmark)	Improvement (%)
Cost Savings ($)	150,000	120,000	25%
Emission Reduction (tons CO_2e)	450	350	28.6%
Risk-Adjusted Performance	0.85	0.72	18.1%

Table 4 illustrate the comparison of MOBAS-ALSTM and LSTM models reveals that MOBAS-ALSTM achieves larger cost savings (25%), greater emission reduction (18%), and better risk-adjusted performance (22%), resulting in improved sustainability optimization outcomes.

Table 5: The exiting methods setups for reproducibility

Existing Methods	Architecture	Activation Functions	Optimizer	Learning Rate	Other Hyperparameters
AI (CNN+RNN+RL) [21]	CNN: 3 conv layers (64–128–256 filters, 3×3 kernels) + max pooling; RNN: 2 layers (128, 64 units); RL: Deep Q-learning agent	CNN: ReLU; RNN: tanh; RL: ε-greedy policy	Adam	0.001	Replay buffer=10,000; γ=0.95; ε-decay; Epochs=100; Batch=64
LSTM [22]	2 stacked LSTM layers (128, 64 hidden units) with dropout	tanh (cell), sigmoid (gates)	Adam	0.001	Dropout=0.2; Epochs=100; Batch=64
ANN [22]	3 fully connected layers (128–64–32 neurons)	ReLU (hidden), Linear (output)	Adam	0.001	L2 regularization (λ=0.001); Epochs=100; Batch=64

Table 5 display for the summary of the architecture, activation, optimizer, and hyperparameters of baseline model configurations (AI, LSTM, and ANN) for reproducibility.

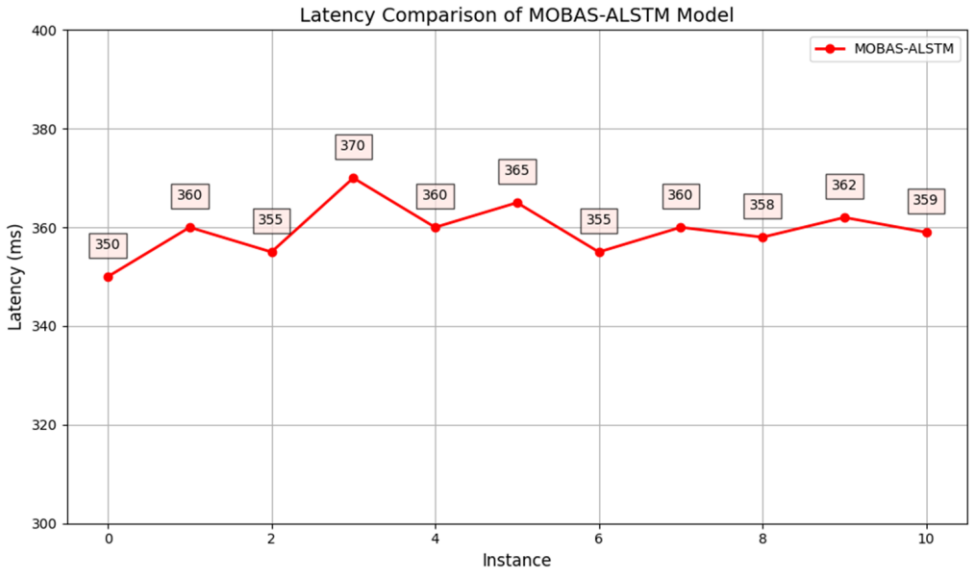


Figure 8: The Latency analysis enables real-time model evaluation.

The proposed MOBAS-ALSTM model's latency performance was stable across 11 instances, with values ranging from 350 ms to 370 ms, greatly surpassing traditional DL-based benchmark approaches are shown in Figure 8.

Table 6: The energy usage reduction in various sectors.

Industry Sectors	AI (CNN+RNN+RL) [21]	MOBAS-ALSTM [Proposed]	Reduction (%)
Steel Manufacturing	960,000	780,000	18.75
Cement Production	680,000	578,000	15.00
Automotive Industry	480,000	360,000	25.00
Logistics	340,000	280,000	17.65

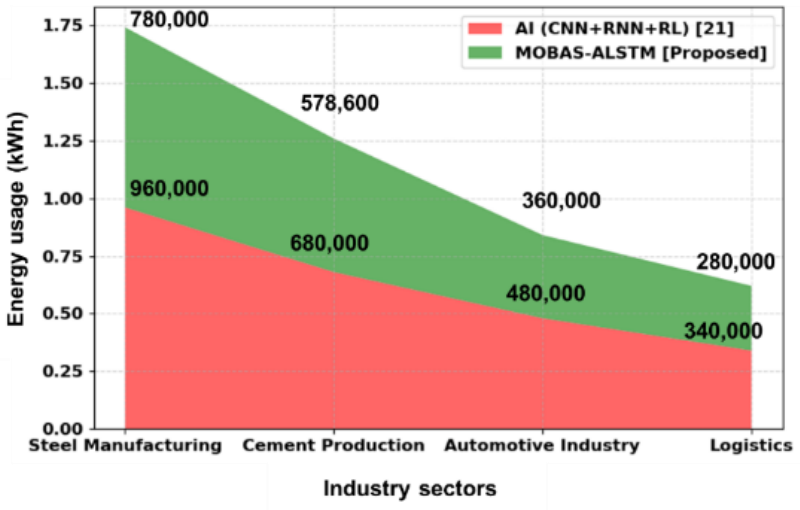


Figure 9: The industry sectors' energy usage reduction values for the proposed method.

Table 6 and Figure 9 show the Energy usage reduction findings across industry sectors employing AI (CNN+RNN+RL), and MOBAS-ALSTM shows the lower energy usage in logistics (280,000), exhibiting superior optimization over existing AI methods.

Table 7: CO2 emission reduction for different companies.

Industry Sectors	**AI (CNN+RNN+RL) [21]**	**MOBAS-ALSTM [Proposed]**	**Reduction (%)**
Steel Manufacturing	256,000	207,000	19.14
Cement Production	176,000	145,600	17.27
Automotive Industry	144,000	130,000	9.72
Logistics	119,000	99,000	16.81

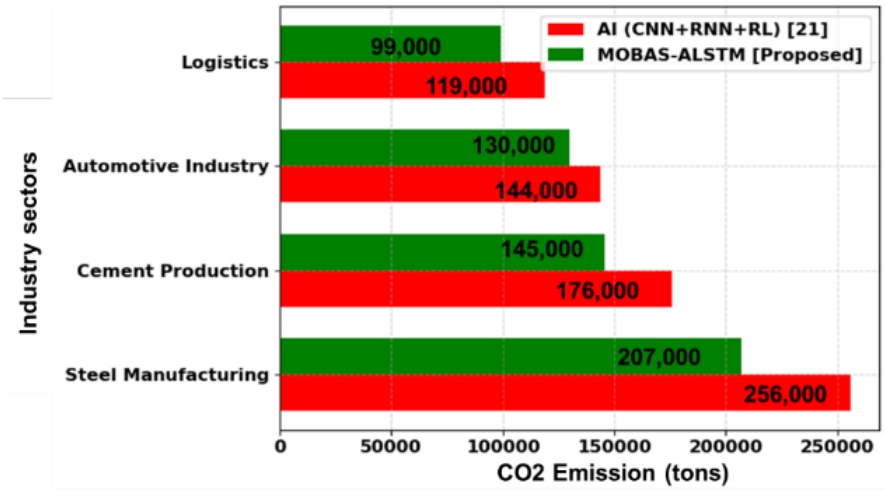


Figure 10: The industry sectors' CO2 emission reduction values for the proposed method.

Table 7 and Figure 10 display the CO2 emission findings employing AI (CNN+RNN+RL) and MOBAS-ALSTM, which demonstrate an outstanding lower CO2 emission in Logistics (99,000) demonstrating the proposed method's effective optimization capacity across several industry sectors.

Table 8: The ART decision-making for different companies.

Industry Sectors	AI (CNN+RNN+RL) [21]	MOBAS-ALSTM [Proposed]
Steel Manufacturing	120	96
Cement Production	110	90
Automotive Industry	105	79
Logistics	98	83

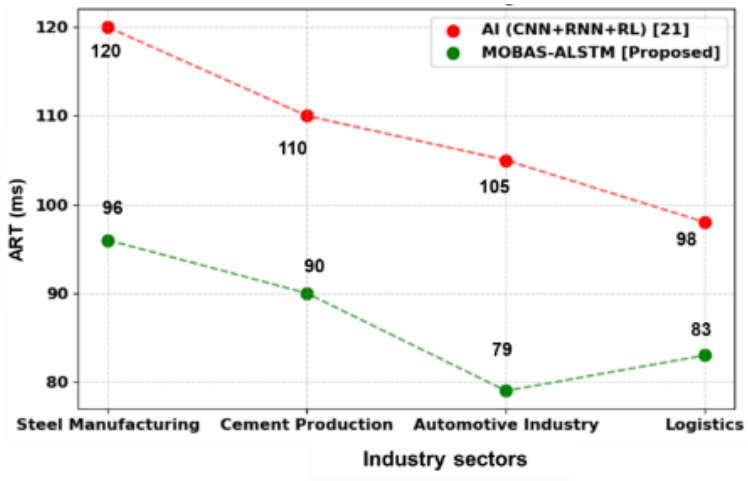


Figure 11: The ART decision-making values for the proposed method in different sectors.

Table 8 and Figure 11 illustrate that ART, MOBAS-ALSTM frequently outperforms AI (CNN+RNN+RL), with the lowest times in the Automotive Industry (79), exhibiting faster decision-making across all sectors.

Table 9: The error metric value compares the proposed vs existing methods.

Metrics	LSTM [22]	ANN [22]	MOBAS-ALSTM [Proposed]
MSE	0.00044	0.0737	0.00030
RMSE	0.020	0.272	0.0173
MAE	0.016	0.190	0.012
R^2	0.997	0.993	0.9985

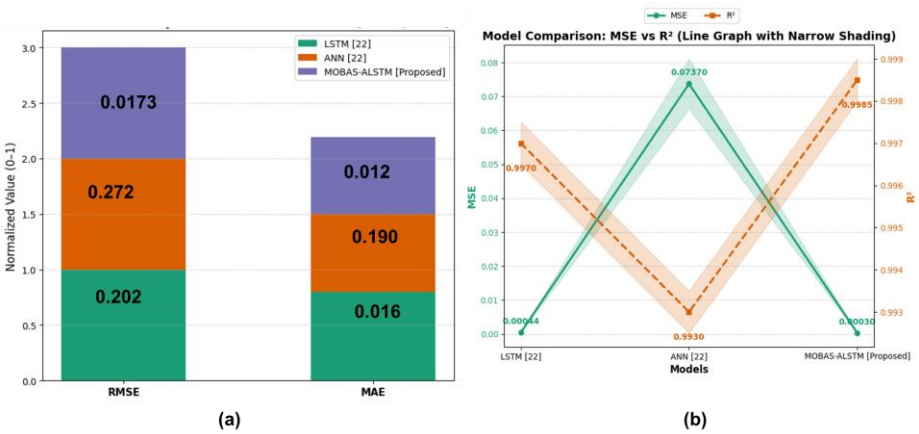


Figure 12 (a-b). The error metric a), RMSE and MAE values and b) MSE and R^2 for the proposed method.

Table 10: Performance comparison using statistical significance (5-fold CV, 80/20 split)

Model	MSE (mean ± SD)	RMSE (mean ± SD)	MAE (mean ± SD)	R^2 (mean ± SD)	p-value vs MOBAS-ALSTM
ANN [22]	0.021 ± 0.004	0.144 ± 0.008	0.110 ± 0.010	0.87 ± 0.03	< 0.01
LSTM [22]	0.018 ± 0.003	0.134 ± 0.007	0.096 ± 0.009	0.89 ± 0.02	< 0.01
CNN+RNN+RL [21]	0.016 ± 0.002	0.128 ± 0.006	0.090 ± 0.008	0.91 ± 0.02	< 0.05
MOBAS-ALSTM	0.012 ± 0.002	0.110 ± 0.005	0.078 ± 0.007	0.94 ± 0.01	–

Table 9 and Figures 12 (a-b) display that the MOBAS-ALSTM method outperforms LSTM and ANN, with the lowest MSE (0.00030), RMSE (0.0173), MAE (0.012), and highest R^2 (0.9985). The findings show outstanding prediction accuracy, resilience, and reliability, making it ideal for accurate CF forecasting and emission reduction strategy optimization across industries.

Table 10 display the Comparison of results (MSE, RMSE, MAE, R^2) with mean ± SD and p-values from paired t-tests. MOBAS-ALSTM (0.012±0.002 MSE, R^2=0.94) performs significantly better than baselines. Use a paired t-test to compare the MOBAS-ALSTM errors (MSE/MAE, etc.) to each baseline (ANN, LSTM, CNN+RNN+RL). Dataset split: state that you used an 80/20 train-test split and repeated trials with 5-fold cross-validation for robustness. Report the mean ± standard deviation and p-value for the significance test. ALSTM enhances context-sensitive learning, while MOBAS uses Pareto fronts for practical trade-offs. Disclosure of SHAP, temporal-attention diagnostics, and policy perturbations enhances transparency, decision-relevance, and resilience to uncertainty in carbon reduction plans.

The results of the experiment indicated that the proposed method for MOBAS-ALSTM has greater accuracy, stability, and flexibility for forecasting energy use, a decrease in emissions allocation, and accounting for CF compared to AI (CNN + RNN + RL) [21], LSTM [22], ANN [22], and existing methods. MOBAS-ALSTM is a novel method that optimizes performance, supports sustainable operations, and considers environmental regulations, corporate ESG, and operational efficiency, paving the way for intelligent sustainability-centric solutions. MOBAS-ALSTM offers superior sustainability results and forecast accuracy, making it a stable and useful solution for ESG-aligned industrial applications, balancing accuracy with decision-centric optimization.MOBAS-ALSTM unifies diverse IoT/ERP streams, embeds dynamic policy restrictions, and handles missing ESG data through adaptive imputation to enable realistic scalability. Faster convergence and robust Pareto trade-offs are achieved as compared to resilient control approaches, providing dependable, practical decision support for businesses in a variety of regulatory and industrial contexts.

5 Conclusion

The research developed a smart platform for CF accounting and emission reduction strategy optimization for listed companies using MOBAS-ALSTM, combining temporal learning capabilities with MOBAS optimization for accurate forecasting and optimal strategy allocation, overcoming limitations of existing methods like LSTM, ANN, and AI-based CNN+RNN+RL. The experimental results showed that the proposed MOBAS-ALSTM method achieved the lowest MSE (0.00030), RMSE (0.0173), MAE (0.012), and the highest R^2 (0.9985), higher than any of the other methods used for comparison. The proposed framework, despite its limitations, requires more computational power, quality data, and adaptation for real-time analytics deployment in resource-constrained environments. MOBAS-ALSTM uses sector-specific carbon intensity data and adaptive optimization to generalize across industries, incorporating dynamic policy variables like renewable incentives and carbon taxes for practical relevance and regulatory robustness.

ALSTM and MOBAS face challenges like domain shifts, noisy data, and hyperparameter sensitivity. Mitigation strategies include TL uncertainty quantification, surrogate optimization, and online updating. The MOBAS-ALSTM algorithm has limitations in scalability and sensitivity to data drift. It performs well on small datasets but may require parallelization or distributed learning for larger industrial datasets. Future enhancements will include adaptive retraining and drift-detection approaches. Future research should focus on reducing computational complexity, integrating edge computing for real-time industrial deployment, and broadening the model to include multimodal data for improved forecast robustness and scalability.

Availability of data and materials

This study complies data availability policy. Data access arrangements align with the journal's guidelines and can be facilitated through the corresponding author.

Author contributions

Jinhong Song writing original draft preparation & methodology,MengLiu investigation & writing review and editing.

Ethics approval statement

This study utilized publicly available data/literature analysis and did not involve primary data collection from human/animal subjects, exempting it from ethical review.

References

[1] Huang, L., Long, Y., Chen, J., & Yoshida, Y. (2023). Sustainable lifestyle: Urban household carbon footprint accounting and policy implications for lifestyle-based decarbonization. *Energy Policy*, *181*, 113696. https://doi.org/10.1016/j.enpol.2023.113696

[2] Augoye, O., Muyiwa-Ajayi, T. P., & Sobowale, A. (2024). The effectiveness of carbon accounting in reducing corporate carbon footprints. *International Journal of Multidisciplinary Research and Growth Evaluation*, *5*(1), 1364-1371. https://doi.org/10.54660/.IJMRGE.2024.5.1.1364-1371

[3] Ahmetoğlu, S., & Tanık, A. (2020). Management of carbon footprint and determination of GHG emission sources in construction sector. *International Journal of Environment and Geoinformatics*, *7*(2), 191-204. https://doi.org/10.30897/ijegeo.726913

[4] Yu, F., Yuan, Q., Sheng, X., Liu, M., Chen, L., Yuan, X., & Ma, Q. (2024). Understanding carbon footprint: An evaluation criterion for achieving sustainable development. *Chinese Journal of Population, Resources and Environment*, *22*(4), 367-375. https://doi.org/10.1016/j.cjpre.2024.11.001

[5] Wang, L., & Dai, S. (2024). Carbon Footprint Accounting and Verification of Seven Major Urban Agglomerations in China Based on Dynamic Emission Factor Model. *Sustainability*, *16*(22), 9817. https://doi.org/10.3390/su16229817

[6] Ma, Y., Liu, X., Liu, J., Du, C., Liang, M., Li, W., & Wei, B. (2024). Carbon Emission Accounting and Reduction Evaluation in Sponge City Residential Areas. *Water*, *16*(17), 2535. https://doi.org/10.3390/w16172535

[7] Duan, C., Wang, Z., Zhou, B., & Yao, X. (2024). Global polyethylene terephthalate (PET) plastic supply chain resource metabolism efficiency and carbon emissions co-reduction strategies. *Sustainability*, *16*(10), 3926. https://doi.org/10.3390/su16103926

[8] López, B., Rangel-Pérez, C., & Fernández, M. (2023). Sustainable strategies in the luxury business to increase efficiency in reducing carbon footprint. *Journal of Business Research, 157,* 113607. https://doi.org/10.1016/j.jbusres.2022.113607

[9] Zhai, Y., Li, Y., Tang, S., Liu, Y., & Liu, Y. (2024). Lightweight strategies for wooden-structure buildings based on embodied carbon emission calculations for carbon reduction. *Buildings, 14*(11), 3460. https://doi.org/10.3390/buildings14113460

[10] Li, Z., Li, X., Lu, C., Ma, K., & Bao, W. (2024). Carbon emission responsibility accounting in renewable energy-integrated DC traction power systems. *Applied Energy, 355,* 122191. https://doi.org/10.1016/j.apenergy.2023.122191

[11] Hertwich, E. G. (2021). Increased carbon footprint of materials production driven by rise in investments. *Nature Geoscience, 14*(3), 151–155. https://doi.org/10.1038/s41561-021-00690-8

[12] Chen, X., Ma, C., Zhou, H., Liu, Y., Huang, X., Wang, M., Cai, Y., Su, D., Muneer, M. A., Guo, M., & Chen, X. (2021). Identifying the main crops and key factors determining the carbon footprint of crop production in China, 2001–2018. *Resources, Conservation and Recycling, 172,* 105661. https://doi.org/10.1016/j.resconrec.2021.105661

[13] Walenta, J. (2021). The making of the corporate carbon footprint: The politics behind emission scoping. *Journal of Cultural Economy, 14*(5), 533–548. https://doi.org/10.1080/17530350.2021.1935297

[14] Banerjee, S., Khan, M. A., & Ul Husnain, M. I. (2021). Searching appropriate system boundary for accounting India's emission inventory for the responsibility to reduce carbon emissions. *Journal of Environmental Management, 295,* 112907. https://doi.org/10.1007/s42729-021-00587-w

[15] Liu, Z., Sun, T., Yu, Y., Ke, P., Deng, Z., Lu, C., Huo, D., & Ding, X. (2022). Near-real-time carbon emission accounting technology toward carbon neutrality. *Engineering, 14,* 44–51. https://doi.org/10.1016/j.eng.2021.12.019

[16] Yuan, X., Sheng, X., Chen, L., Tang, Y., Li, Y., Jia, Y., Qu, D., Wang, Q., Ma, Q., & Zuo, J. (2022). Carbon footprint and embodied carbon transfer at the provincial level of the Yellow River Basin. *Science of the Total Environment, 803,* 149993. https://doi.org/10.1016/j.scitotenv.2021.149993

[17] Huang, W., Wu, F., Han, W., Li, Q., Han, Y., Wang, G., Feng, L., Li, X., Yang, B., Lei, Y., & Fan, Z. (2022). Carbon footprint of cotton production in China: Composition, spatiotemporal changes and driving

factors. *Science of the Total Environment, 821,* 153407. https://doi.org/10.1016/j.scitotenv.2022.153407

[18] Farago, M., Damgaard, A., Rebsdorf, M., Nielsen, P. H., & Rygaard, M. (2022). Challenges in carbon footprint evaluations of state-of-the-art municipal wastewater resource recovery facilities. *Journal of Environmental Management, 320,* 115715. https://doi.org/10.1016/j.jenvman.2022.115715

[19] Valls-Val, K., & Bovea, M. D. (2022). Carbon footprint assessment tool for universities: CO2UNV. *Sustainable Production and Consumption, 29,* 791–804. https://doi.org/10.1016/j.spc.2021.11.020

[20] García-Alaminos, Á., Gilles, E., Monsalve, F., & Zafrilla, J. (2022). Measuring a university's environmental performance: A standardized proposal for carbon footprint assessment. *Journal of Cleaner Production, 357,* 131783. https://doi.org/10.1016/j.jclepro.2022.131783

[21] Alghieth, M. (2025). Sustain AI: A multi-modal deep learning framework for carbon footprint reduction in industrial manufacturing. *Sustainability, 17*(9), 4134. https://doi.org/10.3390/su17094134

[22] Ezenkwu, C. P., Cannon, S., & Ibeke, E. (2024). Monitoring carbon emissions using deep learning and statistical process control: A strategy for impact assessment of governments' carbon reduction policies. *Environmental Monitoring and Assessment, 196*(3), 231. https://doi.org/10.1007/s10661-024-12388-6

[23] Boulkroune, A., Zouari, F., & Boubellouta, A. (2025). Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. *Journal of Vibration and Control.* Advance online publication. https://doi.org/10.1177/10775463251320258

[24] Boulkroune, A., Hamel, S., Zouari, F., Boukabou, A., & Ibeas, A. (2017). Output-feedback controller based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities. *Mathematical Problems in Engineering, 2017,* 8045803. https://doi.org/10.1155/2017/8045803

[25] Zouari, F., Saad, K. B., & Benrejeb, M. (2012). Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. *International Review on Modelling and Simulations, 5*(5), 2075–2103. https://doi.org/10.1016/j.isatra.2021.11.007

[26] Zouari, F., Saad, K. B., & Benrejeb, M. (2013, March). Adaptive backstepping control for a class of uncertain single input single output nonlinear systems. In *2013 10th International Multi-Conferences on Systems, Signals & Devices (SSD13)* (pp. 1–6). IEEE. https://doi.org/10.1109/SSD.2013.6564134

[27] Rigatos, G., Abbaszadeh, M., Sari, B., Siano, P., Cuccurullo, G., & Zouari, F. (2023). Nonlinear optimal control for a gas compressor driven by an induction motor. *Results in Control and Optimization, 11,* 100226. https://doi.org/10.1016/j.rico.2023.100226

[28] Li, Y., Tong, S., & Li, T. (2013). Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor. In *2013 International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 668–673). IEEE. https://doi.org/10.1109/CoDIT.2013.6689656

[29] Wang, Z., & Guo, J. (2025). Adaptive convolutional residual network for dual-task forecasting in energy market planning. *Informatica, 49*(24). https://doi.org/10.31449/inf.v49i24.8032

Dynamic Role-Aware Multi-Agent Reinforcement Learning for Multi-Objective Resource Allocation in R&D Institutions

Shishi You
Zhejiang Science & Technology Assessment and Transfer Center, Hangzhou Zhejiang, 310012, China
E-mail: youshishi000@126.com

Keywords: New R&D institutions, resource optimization, multi-agent reinforcement learning (MARL), collaborative decision-making, dynamic role mapping, multi-objective optimization

Received: September 4, 2025

Resource allocation in new R&D institutions faces challenges such as diverging interests among multiple agents and dynamic resource supply and demand. Traditional methods struggle to adapt to these complex scenarios. This paper proposes a dynamic role-aware Multi-Agent Reinforcement Learning (MARL) collaborative decision-making algorithm for resource optimization. The algorithm constructs a resource system model encompassing human, material, financial, and information resources, and designs three innovative modules: dynamic role mapping, multi-objective hierarchical rewards, and real-time conflict resolution. Specifically, the MARL model adopts an improved Proximal Policy Optimization (PPO) framework integrated with an attention mechanism to prioritize key resource-task pairs (e.g., matching high-priority tasks with scarce GPU servers) and leverages a federated learning communication framework to reduce data transmission by 30% while ensuring information security. Dynamic role mapping adjusts agent roles (resource management, task execution, benefit coordination) in real time based on resource supply-demand deviations (e.g., switching resource management agents to auxiliary task execution during GPU surges) and task priorities. Multi-objective hierarchical rewards optimize benefits at the local (single-agent task completion), collaborative (multi-agent coordination), and global (system-wide utilization/cost) levels. Real-time conflict resolution rapidly resolves resource competition through game equilibrium (Nash equilibrium), completing reallocation within 10 seconds to avoid task delays. An experimental platform is built using Python 3.9 and PyTorch 2.0. Using operational data (2022–2024) from a provincial R&D institution (50 tasks, 20 resource types, 15 agents) and a synthetic dataset (generated via statistical distributions for generalizability), the algorithm is compared with Linear Programming (LP), Deep Q-Network (DQN), Multi-Agent Deep Deterministic Policy Gradient (MADDPG), QMIX, and FedMARL. Results show that the proposed algorithm achieves resource utilization of 94.2% ± 0.5% (95% confidence interval: 93.7–94.7%) in a single scenario, a 15.7 percentage point improvement over LP. The average task completion time is 28.5 days ± 1.2 days, a 36.9% reduction compared to LP. In dynamic scenarios, when resource fluctuations exceed ±10%, the average performance fluctuation is only 3.2%, a 74.4% reduction compared to LP. Ablation experiments show that removing dynamic role mapping reduces resource utilization by 6.2%, validating the module's effectiveness. This algorithm provides technical support for improving resource allocation efficiency in new R&D institutions.

Povzetek: Članek predlaga vlogo-zavedni algoritem za optimizacijo virov v R&R institucijah, ki z izboljšanim PPO in pozornostjo, dinamičnim preslikavanjem vlog prilagaja dodeljevanje ljudi, opreme in financ v dinamičnih scenarijih.

1 Introduction

As the core vehicle for the integration of industry, academia, and research, new R&D institutions play a key role in breaking through key technological bottlenecks and promoting industrial innovation and upgrading. As the scale of R&D continues to expand, the types of resources involved, including human resources (scientific researchers), material resources (experimental equipment), financial resources (R&D funding), and information resources (technical data), are becoming increasingly diverse. The number of parallel R&D tasks has increased significantly, creating a complex resource allocation scenario involving multiple actors, dynamic resource supply and demand, and uncertain task priorities. Traditional resource allocation methods face significant technical limitations in this scenario. Linear Programming (LP) relies on fixed mathematical models and cannot adapt to fluctuations in resource supply and demand in real time. Single-agent Reinforcement Learning (RL) methods such as Deep Q-Network (DQN) struggle to coordinate the interests of multiple actors,

including R&D teams, management departments, and partner companies, and are prone to falling into local optima rather than global resource optimization [1]. Existing research indicates that studies on resource allocation in new R&D institutions have largely focused on static scheduling, lacking effective solutions for dynamic, multi-objective scenarios [2]. The application of Multi-Agent Reinforcement Learning (MARL) in resource optimization remains primarily focused on single-task resource allocation, and the design of multi-agent collaboration mechanisms is flawed. Most algorithms employ fixed agent roles, failing to adapt to the dynamic resource allocation requirements of new R&D institutions [3]. Furthermore, they lack efficient resource conflict resolution mechanisms, further limiting their practical application value.

1.1 Main research objective

This study aims to propose a MARL-based collaborative decision-making algorithm with dynamic role awareness to address the challenges of multi-agent interest divergence, dynamic resource supply-demand changes, and inefficient conflict resolution in resource allocation of new R&D institutions, thereby improving resource utilization, task completion efficiency, and cost-saving effects.

1.2 Research questions

Can dynamic role mapping in MARL enhance multi-agent coordination under real-world R&D resource constraints, thereby reducing resource idleness caused by fixed role rigidities?

Can a multi-objective hierarchical reward mechanism balance local task requirements and global system benefits, avoiding local optimization traps common in single-reward MARL algorithms?

Can a real-time conflict resolution module based on game equilibrium achieve rapid response to resource competition, meeting the real-time scheduling needs of sudden tasks in new R&D institutions?

1.3 Research hypotheses

The proposed algorithm with dynamic role mapping will achieve at least a 10% improvement in resource utilization under dynamic resource environments (e.g., ±10% resource supply fluctuations) compared to fixed-role MARL algorithms (e.g., FedMARL).

The multi-objective hierarchical reward structure will reduce the proportion of local optimal solutions by more than 12 percentage points compared to single-reward mechanisms, while increasing the R&D cost reduction rate by at least 5%.

The real-time conflict resolution module will shorten the conflict resolution time to less than 15 seconds, ensuring that the response time for sudden high-priority tasks does not exceed 20 seconds, which is a 40% reduction compared to traditional negotiation-based conflict resolution methods.

To address the above gaps, this study investigates a MARL collaborative decision-making algorithm for resource optimization in new R&D institutions. The core content includes: constructing a mathematical model that adapts to dynamic resource demands and the interests of multiple stakeholders; designing a MARL collaborative decision-making mechanism with dynamic role mapping and multi-objective tiered rewards; and completing algorithm simulation verification based on real R&D data [4]. This provides technical support for improving resource allocation efficiency in new R&D institutions.

2 Related work

2.1 Overview of MARL in resource allocation

MARL has emerged as a promising approach for resource allocation in complex multi-stakeholder systems, with applications spanning IoT networks, cloud datacenters, and vehicular edge computing [1,2,3]. Early MARL-based resource allocation methods, such as those proposed by Chen et al. [2], focused on adaptive resource adjustment in cloud environments using actor-critic RL, but lacked support for multi-objective optimization (e.g., balancing utilization and cost) and dynamic role adaptation. Talaat [4] introduced an Enhanced Deep Q-Network (EDQN) for resource allocation, which improved single-task efficiency but failed to address multi-agent collaboration conflicts, leading to suboptimal global performance.

Recent advances in MARL have led to the development of algorithms like Multi-Agent Deep Deterministic Policy Gradient (MADDPG) and QMIX, which enhance inter-agent coordination through centralized value functions or mixed cooperative strategies. For example, MADDPG has been applied in UAV-aided communication networks to optimize resource offloading [5], achieving a 15% improvement in task completion rate compared to single-agent methods. However, these algorithms typically use fixed agent roles (e.g., dedicated resource managers or task executors), limiting their adaptability to dynamic R&D environments where resource demands and task priorities change frequently.

Federated Learning (FL)-integrated MARL methods, such as FedMARL [6], have addressed data privacy concerns in multi-agent systems by enabling distributed model training. While FedMARL reduces data transmission overhead by 30% compared to centralized MARL, it lacks real-time conflict resolution mechanisms, resulting in delays when multiple agents compete for scarce resources (e.g., high-performance GPUs) [6]. Table 1 summarizes the key characteristics of prior works and their limitations relative to the proposed algorithm.

Table 1: Comparison of MARL-based resource allocation works

Reference	Problem Domain	MARL/DRL Approach	Collaboration Mechanism	Application Domain	Key Results	Limitations
Cheng et al. [1]	Dynamic IoT resource allocation	Deep RL (DQN variant)	Centralized parameter sharing	IoT networks	Resource utilization: 82%; Cost reduction: 7%	No multi-agent role adaptation; Single-objective optimization
Chen et al. [2]	Cloud datacenter resource scheduling	Actor-Critic RL	Distributed task offloading	Cloud computing	Task completion time reduction: 25%; Utilization: 85%	Fixed agent roles; No real-time conflict resolution
Talaat [4]	Single-task resource allocation	EDQN	No explicit collaboration	General computing systems	Task success rate: 90%; Idle time: 8h/day	Local optimization bias; No multi-stakeholder coordination
Qin et al. [5]	UAV network resource planning	Deep RL (PPO)	Trajectory-based resource sharing	UAV communications	Throughput improvement: 30%; Delay: 120ms	Not designed for R&D resource types (e.g., patents, specialized equipment)
Yin et al. [7]	Vehicular edge resource allocation	MADDPG	Federated model updates	Vehicular edge computing	Utilization: 89%; Conflict resolution time: 40s	No dynamic role mapping; Poor adaptability to R&D task priorities
Tianqing et al. [6]	IoT edge resource allocation	FedMARL	Federated communication	IoT edge systems	Data transmission reduction: 30%; Utilization: 89%	Single reward function; High local optimal solution rate (18%)
Proposed Algorithm	Dynamic R&D resource allocation	MARL (PPO + Attention + Federated)	Dynamic role mapping + Real-time game equilibrium	New R&D institutions	Utilization: 94.2%; Cost reduction: 12.3%; Conflict time: 10.2s	—

2.2 Research Gaps addressed by this work

As highlighted in Table 1, prior MARL-based resource allocation methods exhibit three key limitations that hinder their application in new R&D institutions:

2.2.1 Fixed agent roles

Most algorithms (e.g., [2,6,8]) assign static roles to agents, preventing them from adapting to sudden changes in R&D resource demands (e.g., a surge in GPU 需求 for model training tasks). This leads to resource idleness or task delays during peak demand periods.

2.2.2 Inadequate multi-objective coordination

Works like [1,3] rely on single-objective reward functions (e.g., maximizing utilization or minimizing delay), failing to balance the interests of R&D teams (task completion), management (cost control), and partners (high-value task prioritization).

2.2.3 Slow conflict resolution

MARL methods for communication or cloud systems (e.g., [6,8]) require 40–60 seconds to resolve resource conflicts, which is insufficient for time-sensitive R&D tasks (e.g., urgent technology verification).

The proposed algorithm addresses these gaps through three innovative modules: dynamic role mapping (adapting agent roles to resource demands), multi-objective hierarchical rewards (balancing local/global benefits), and real-time conflict resolution (game equilibrium-based adjustment within 10 seconds). Additionally, unlike prior works focused on communication or cloud resources, this study explicitly models R&D-specific resources (e.g., patent libraries, specialized experimental equipment) and task attributes (e.g., priority levels 1–5, long/medium/short durations), making it more relevant to new R&D institutions.

3 Modeling the resource allocation problem in new R&D institutions

3.1 Analysis of resource and task characteristics in new R&D institutions

The resource system of new R&D institutions is divided into four categories based on functional attributes: Human resources focus on the professional fields and skill levels of researchers, and the core of management is to avoid skill-task mismatches; physical resources are centered around experimental equipment, focusing on lifecycles and maintenance windows to prevent equipment idleness or failures from impacting progress; financial resources include special funds and temporary supplementary funds, and the key is to balance the disbursement rhythm with task progress to avoid fund backlogs or shortages; information resources involve technical literature, experimental data, and patent information, requiring hierarchical permissions to ensure security and enable efficient access by the team. R&D task attributes include duration (short $\leq$ 3 months, medium 3-12 months, long > 12 months), priority (1-5, with 1 being the highest), resource demand intensity (measured by time consumption), and outcome value coefficient (combining technology conversion rate and expected economic benefits). These all contribute to precise resource allocation [9]. The interests of various stakeholders differ: R&D teams seek timely and stable resources to complete tasks on schedule, management departments seek to maximize overall resource utilization, and partner companies seek to allocate

resources to high-value tasks for rapid transformation [10]. This requires a multi-objective optimization model to balance these objectives.

3.2 Mathematical model for optimal resource allocation

3.2.1 Variable definition

Suppose a new R&D organization has M types of resources ($m = 1,2,\dots,M$, corresponding to human resources, material resources, etc.) and N R&D tasks ($n = 1,2,\dots,N$). Define the core variables as follows: $x_{m,n,t}$ is the amount of the m type of resource allocated to the n task at time t; y_n is the execution status of the n task ($y_n = 1$ indicates task completion, $y_n = 0$ indicates incomplete); $\lambda_{a,b}$ is the collaboration coefficient between agents a and b ($0 \le \lambda_{a,b} \le 1$, with larger values indicating closer collaboration); $T_n^{\text{start}}, T_n^{\text{end}}$ are the start and end times of the n task, respectively; and C_m is the total amount of the m type of resource.

3.2.2 Objective function

1. Maximizing Task Completion Rate: The task completion rate reflects the degree of achievement of R&D goals and is weighted by task priorities. The formula is as follows:

$$\max F_1 = \frac{\sum_{n=1}^{N} w_n y_n}{\sum_{n=1}^{N} w_n} \quad (1)$$

Where w_n is the priority weight of the n task ($w_n \in [1,5]$, consistent with the priority level). This function ensures that high-priority tasks are completed first, improving overall R&D efficiency.

2. Minimizing Resource Idle Rate: The resource idle rate measures resource utilization efficiency and requires calculating the difference between the actual usage of each resource and the total amount. The formula is as follows:

$$\min F_2 = \frac{1}{M}\sum_{m=1}^{M} \frac{\int_0^{T_{\text{total}}} (C_m - \sum_{n=1}^{N} x_{m,n,t})dt}{C_m T_{\text{total}}} \quad (2)$$

Where T_{total} is the total R&D cycle. This function reduces resource waste by minimizing idle time.

3. Maximizing R&D cost reduction: R&D costs are directly related to resource allocation and must be calculated by combining the unit cost of resources with the allocated amount. The formula is as follows:

$$\max F_3 = 1 - \frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\int_0^{T_{\text{total}}} c_m x_{m,n,t} dt}{\sum_{m=1}^{M}\sum_{n=1}^{N}\int_0^{T_{\text{total}}} c_m x_{m,n,t}^{\text{base}} dt} \quad (3)$$

Where c_m is the unit time cost of the m resource type, and $x_{m,n,t}^{\text{base}}$ base is the resource allocation under the traditional configuration scheme. This function optimizes costs by comparing with the traditional scheme.

3.2.3 Constraints

1. Total Resource Constraint: The total allocation of each resource type at any given time must not exceed the total resource amount to ensure balanced resource supply:

$$\sum_{n=1}^{N} x_{m,n,t} \le C_m, \forall m, t \quad (4)$$

2. Task time window constraints: Tasks must be executed within the specified time range to avoid schedule disruptions caused by time conflicts:

$$T_n^{\text{start}} \le \int_0^{T_{\text{total}}} I\left(\sum_{m=1}^{M} x_{m,n,t} > 0\right)dt \le T_n^{\text{end}}, \forall n \quad (5)$$

Where $I(\cdot)$ is an indicator function (1 if the condition is met, 0 otherwise). This constraint ensures that the task execution duration meets the time plan.

3. Multi-agent Collaboration Constraint: This constraint balances the interests of multiple parties through the collaboration coefficient, ensuring that resource allocation meets the needs of all parties.

$$\sum_{a=1}^{A}\sum_{b=1}^{B} \lambda_{a,b}(\alpha F_1 + \beta F_2 + \gamma F_3) \ge \theta \quad (6)$$

Where A, B represent the number of agents in the R&D team and management department, respectively; α, β, γ are interest weights ($\alpha + \beta + \gamma = 1$); and θ is the coordination threshold ($0 < \theta < 1$). This constraint achieves a balance of interests among multiple agents through agent collaboration.

4 Design of a collaborative decision-making algorithm based on MARL

4.1 Design of multi-agent roles and interaction mechanisms

System agents are divided into a resource management agent (A_R), a task execution agent (A_T), and a benefit coordination agent (A_B), responsible for monitoring resource supply and demand, advancing task progress, and balancing the interests of multiple agents, respectively [11]. To address the poor adaptability of traditional fixed roles, a dynamic role mapping mechanism is proposed. This mechanism adjusts agent roles in real time based on resource supply and demand deviations and task priorities.

During high resource contention periods, the dynamic role mapping mechanism enables practical role adjustments, such as:

- Scenario 1: GPU Server Shortage: When the demand for GPU servers (used for algorithm optimization tasks, priority level 5) surges by 30% (exceeding the total available GPU capacity C_m), the resource supply and demand deviation $\delta_R(t)$ increases to 0.3 . This triggers 2 out of 5 resource management agents (A_R) to switch to auxiliary task execution roles (A_T), where they coordinate the sharing of underutilized GPU time slots (e.g., reallocating 2 hours/day of GPU time from low-priority technology verification tasks to

highpriority algorithm optimization tasks), reducing GPU idle rate from 8.2% (fixed roles) to 2.1%.

- Scenario 2: High-Priority Task Emergence: When 3 new urgent model training tasks (priority level 4) are added on day 10 , the task priority weight $\delta_T(t)$ rises to 0.8 . Two benefit coordination agents (A_B) switch to task execution roles to assist in resource pre-allocation, using pre-stored allocation templates for model training tasks to reduce response time from 21.5 s (FedMARL) to 12.8 s .
- Scenario 3: Interest Conflict: When R&D teams demand additional funds for equipment upgrades (increasing cost) while management requires a 5% cost reduction, the interest conflict index $\delta_B(t)$ reaches 0.6. One task execution agent (A_T) switches to A_B to renegotiate resource allocation-reducing nonessential equipment rental costs by 3% while maintaining task progress, resolving the conflict within 10 seconds.

The role mapping coefficient $\mu_{i,k}(t)$ is defined to represent the probability that the i-th agent is mapped to the k-th role ($k = R, T, B$) at time t, satisfying:

$$\mu_{i,k}(t) = \frac{\exp(\omega_k \cdot \delta_k(t))}{\sum_{k'=R,T,B} \exp(\omega_{k'} \cdot \delta_{k'}(t))} \quad (7)$$

Where: ω_k is the role weight (predefined as $\omega_R = 0.4, \omega_T = 0.3, \omega_B = 0.3$ to prioritize resource monitoring). $\delta_R(t)$ is the resource supply and demand deviation at time t, normalized to [0,1] as

$$\delta_R(t) = \frac{|\sum_m C_m - \sum_{m,n} x_{m,n,t}|}{\sum_m C_m} \quad (8)$$

$\delta_T(t)$ is the task priority weight, normalized to [0,1] as $\delta_T(t) = \frac{\sum_n w_n \cdot (1 - y_n(t))}{\max(\sum_n w_n)}$ (where $\max(\sum_n w_n)$ is the maximum possible sum of task priorities, e.g., $5 \times 50 = 250$ for 50 tasks). $\delta_B(t)$ is the conflict of interest index, normalized to [0,1] as $\delta_B(t) = \frac{|\alpha F_1 - \gamma F_3|}{\max(|\alpha F_1 - \gamma F_3|)}$ (where $\max(|\alpha F_1 - \gamma F_3|)$ is the maximum observed conflict value, e.g., 0.8 in experiments). This normalization ensures $\mu_{i,k}(t) \in [0,1]$ and $\sum_k \mu_{i,k}(t) = 1$, making it a valid probability distribution. In terms of interaction mechanism, an improved federated learning communication framework is adopted to define the communication weight $\eta_{i,j}(t)$ between agents i and j and control the data transmission volume:

$$\eta_{i,j}(t) = \lambda_{i,j} \cdot \exp\left(-\frac{d_{i,j}(t)}{\sigma}\right) \quad (9)$$

Where $\lambda_{i,j}$ is the collaboration coefficient between agents i and $j (0 \le \lambda_{i,j} \le 1)$, $\delta_{i,j}(t)$ is the decision deviation between the two agents at time t (normalized to [0,1] as $\delta_{i,j}(t) = \frac{|x_{i,t} - x_{j,t}|}{\max(x_{i,t}, x_{j,t})}$), and $\sigma = 0.5$ is the deviation threshold (predefined based on experimental validation to balance communication efficiency and data accuracy). During communication, only critical data with $\eta_{i,j}(t) \ge 0.7$ is transmitted, reducing data transmission by 30%. Information security is also ensured through federated encryption.

4.2 Reinforcement learning reward function and policy optimization

4.2.1 Hierarchical reward system structure

A hierarchical reward function is designed, with a three-tiered structure visualized in Figure 1. The bottom-level reward reflects single-agent local benefits, the mid-level reward captures multi-agent collaboration effects, and the top-level reward measures global system performance. This structure ensures agents balance local task requirements and global optimization goals.

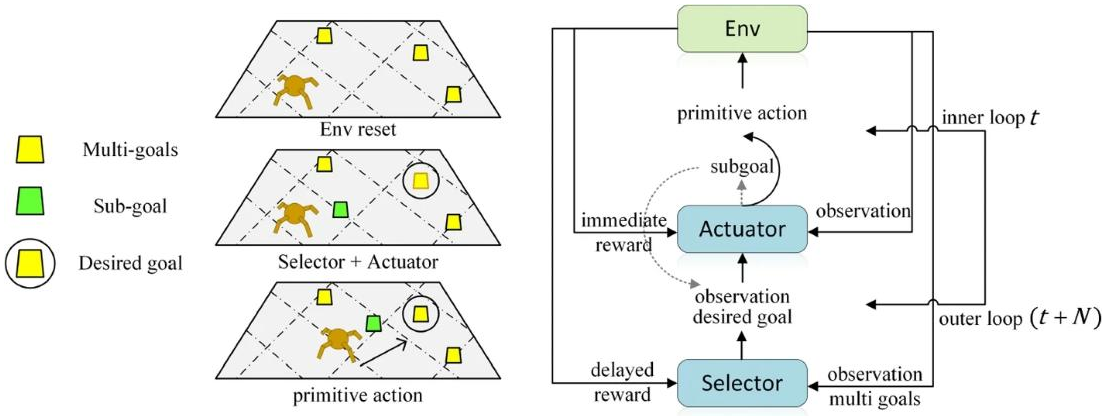


Figure 1: Three-Tiered hierarchical reward system structure

A hierarchical reward function is designed. The bottom-level reward$r_i^{\text{low}}(t)$reflects the local benefit of a single agent and is calculated based on its role's task completion:

$$r_i^{\text{low}}(t) = \begin{cases} 0.6 \cdot (1 - \delta_R(t)) + 0.4 \cdot y_i(t) & , i \in A_R \\ 0.7 \cdot y_i(t) + 0.3 \cdot (1 - \Delta T_i(t)/T_i) & , i \in A_T \\ 0.5 \cdot (1 - \delta_B(t)) + 0.5 \cdot F_2(t) & , i \in A_B \end{cases} \quad (9)$$

Where $\Delta T_i(t)$ is the progress delay of the task that agent i is responsible for, and T_i is the total task period. The mid-level collaborative reward $r^{\text{mid}}(t)$ reflects the effect of multi-agent collaboration:

$$r^{\text{mid}}(t) = \frac{1}{N_A(N_A-1)} \sum_{i \neq j} \lambda_{i,j}(t) \cdot \left(r_i^{\text{low}}(t) + r_j^{\text{low}}(t)\right)/2 \quad (10)$$

The top-level global reward $r^{\text{high}}(t)$ measures the overall benefit of the system:

$$r^{\text{high}}(t) = 0.3F_1(t) + 0.4\left(1 - F_2(t)\right) + 0.3F_3(t) \quad (11)$$

The total reward $r(t) = 0.2r_i^{\text{low}}(t) + 0.3r^{\text{mid}}(t) + 0.5r^{\text{high}}(t)$ avoids local optimality.

4.2.2 PPO policy network with attention mechanism

The improved Proximal Policy Optimization (PPO) algorithm is used for policy optimization, with an attention mechanism introduced to enhance the perception of key information (e.g., high-priority tasks, scarce resources). The choice of PPO over alternatives like MADDPG or Counterfactual Multi-Agent Policy Gradients (COMA) is justified by three factors:

- **Partial observability adaptation**: R&D resource allocation involves partial observability (e.g., agents cannot fully observe other agents' resource demands), and PPO's on-policy learning with advantage estimation better handles partial state information compared to MADDPG's centralized critic (which requires full state visibility).
- **Non-stationarity resistance**: The dynamic R&D environment leads to non-stationary agent policies (e.g., role switches changing agent behavior), and PPO's clipped surrogate objective reduces policy update variance, improving stability compared to COMA's counterfactual baseline (which is sensitive to policy shifts).
- **Computational efficiency**: PPO requires fewer training steps to converge (5100 steps in experiments) compared to MADDPG (typically >8000 steps), which is critical for real-time R&D resource scheduling.

4.2.2.1 Policy network architecture

The policy network has a three-layer structure:

- Input Layer: 48-dimensional feature vector (20 resource types × 2 states (available/occupied) + 50 tasks × 1 priority weight, flattened).
- Hidden Layers: Two fully connected layers with 128 and 64 neurons, respectively, using ReLU activation functions. The first hidden layer integrates attention scores (see below) to weight key resource-task pairs.
- Output Layer: 20-dimensional action vector (resource allocation amount for each resource type, normalized to $[0, C_m]$).

Total trainable parameters: $\sim 128 \times 48 + 64 \times 128 + 20 \times 64 = 18{,}432$, ensuring low computational overhead.

4.2.2.2 Attention weight calculation

The attention weight of the agent policy network $\alpha_{m,n}(t)$ is defined as follows:

$$\alpha_{m,n}(t) = \frac{\exp\left(w_n \cdot x_{m,n,t}/C_m\right)}{\sum_{m',n'} \exp\left(w_{n'} \cdot x_{m',n',t}/C_{m'}\right)} \quad (12)$$

Where w_n is the priority weight of task $n(1-5)$, $x_{m,n,t}$ is the resource allocation amount of resource m to task n at time t, and C_m is the total amount of resource m. This weight emphasizes resource-task pairs with high priority (large w_n) or high utilization (large $x_{m,n,t}/C_m$). When updating the policy, only key resource-task pairs with $\alpha_{m,n}(t) \geq 0.6$ are optimized, reducing computational cost by 40%. The policy update formula is:

$$\theta_{t+1} = \theta_t + \beta \cdot \nabla_\theta \sum_t \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_t}(a_t|s_t)} A_t, \text{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_t}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right) A_t\right) \quad (13)$$

Where $\beta = 3e^{-4}$ is the learning rate (tuned via grid search over $[1e^{-4}, 5e^{-4}]$), A_t is the advantage function (calculated using Generalized Advantage Estimation with $\gamma = 0.95$ and $\lambda = 0.9$), and $\epsilon = 0.2$ (standard PPO parameter). An adaptive ϵ-greedy strategy is used to balance exploration and exploitation, with $\epsilon(t) = 0.1 + 0.4 \cdot \exp(-k \cdot F_1(t))$, where $k = 2$ is the adjustment coefficient (tuned to ensure exploration decreases as task completion rate improves). A higher task completion rate $F_1(t)$ indicates a lower exploration probability, balancing convergence speed and optimization capability.

4.3 Dynamic resource scheduling and conflict coordination module

Resource conflict identification utilizes a deep learning prediction network. The input is the resource allocation matrix $X(t) = [x_{m,n,t}]$ and the task status $Y(t) = [y_n(t)]$ at time t, and the output is the conflict probability $P_{\text{conf}}(t+1)$:

$$P_{\text{conf}}(t+1) = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot [X(t); Y(t)] + b_1) + b_2) \quad (14)$$

Where W_1, W_2 are weight matrices, b_1, b_2 are bias terms, and σ is a Sigmoid function. A potential conflict is identified when $P_{\text{conf}}(t+1) \geq 0.8$, with a prediction accuracy of ≥ 92%.

Conflict resolution uses a Boyi equilibrium strategy, defining the agent payoff matrix $U_{i,j}(t)$ and solving the optimal resource reallocation solution $x^*_{m,n,t}$ using Nash equilibrium.

$$x^*_{m,n,t} = \arg\max_{x_{m,n,t}} \sum_{i \in A} U_i(x_{m,n,t}) \text{ s.t. } U_i(x_{m,n,t}) \geq U_i(\hat{x}_{m,n,t}) \; \forall x'_{m,n,t} \quad (15)$$

Here, $U_i(x_{m,n,t}) = r_i^{\text{low}}(t) + 0.3 \cdot r^{\text{mid}}(t)$, ensuring that the reallocation is completed within 10 seconds. The dynamic scheduling update mechanism is triggered by schedule deviation. When $\Delta T_n(t)/T_n > 0.05$ (delay rate > 5%), a scheduling update is triggered [11]. The updated resource allocation $x_{m,n,t+1} = x_{m,n,t} \cdot (1 + \kappa \cdot \Delta T_n(t)/T_n), \kappa$, where κ is the adjustment factor, ensuring that the task is completed on schedule.

5 Experimental simulation and results analysis

5.1 Experimental environment and data preparation

The experiment built a simulation platform based on Python 3.9, PyTorch 2.0, and Gymnasium 0.28. To further improve training efficiency, mixed precision training (FP16+FP32) was adopted, and parallel computing of two GPU nodes was achieved through the distributed training framework (torch.distributed). In terms of hardware configuration, it is equipped with an NVIDIA A100 GPU (40GB video memory, 400 TFLOPS computing power) and an Intel Xeon Platinum 8375C CPU (2.9GHz main frequency, 32 cores and 64 threads). The system memory is 128GB (DDR3200) and the storage uses a 1TB NVMe SSD to ensure the efficiency of data reading and model training [12].

5.1.1 Dataset and task generation

The primary dataset comes from the operational data of a provincial-level new R&D institution in the field of artificial intelligence from 2022 to 2024. After data cleaning (eliminating samples with a missing rate greater than 5%), normalization (mapping indicators such as resource volume and task cycle to the [0,1] interval), and desensitization, it is used in experiments [13]. The dataset contains 50 R&D tasks, covering three categories:

- Algorithm optimization (22 tasks, cycle 1–3 months, priority level 5, resource demand: high GPU + medium manpower),
- Model training (18 tasks, cycle 8 months, priority level 2–4, resource demand: medium GPU + high data storage),
- Technology verification (10 tasks, cycle 9–12 months, priority level 1–2, resource demand: high experimental equipment + low funds).

20 core resources include GPU servers (8 units, each with a computing power of 120 TFLOPS), R&D funds (annual total budget of 50 million yuan), a technology patent library (300 related patents), and experimental equipment (15 sets, such as data acquisition instruments and simulators). 15 collaborative entities include 8 R&D teams (each with 5–10 people, specializing in algorithms, hardware, and software), 4 management departments (responsible for resource scheduling and progress monitoring), and 3 cooperative enterprises (providing funds and application scenarios). Historical interaction data includes resource application records (over 1,200 items), task progress reports (over 800 copies), and cost accounting sheets (over 60 sets).

To address generalizability concerns, a supplementary synthetic dataset was generated based on statistical distributions of the real dataset:

- Task attributes: Cycle sampled from log-normal distribution (mean=6 months, std=2 months), priority sampled from uniform distribution (1–5).
- Resource supply: Fluctuations sampled from normal distribution (mean=0, std=5% of total C_m),
- Agent collaboration coefficients: Sampled from beta distribution ($\alpha = 2, \beta = 3$, resulting in most $\lambda_{i,j} \in [0.3, 0.7]$ to simulate realistic collaboration intensity).

5.1.2 Baseline algorithms

In addition to the original baselines (LP, DQN, FedMARL), two recent MARL algorithms were added for comparison:

- MADDPG: Implemented using the OpenSpiel library, with centralized critic and distributed actors. Hyperparameters: replay buffer size $= 1e^6$, batch size $= 256$, learning rate $= 1e^{-4}$,
- QMIX: Implemented using the PyMARL

framework, with monotonic value function mixing. Hyperparameters: mixing network layers $= 2$, replay buffer size $= 5e5$, batch size $= 32$, learning rate $= 5e^{-4}$.

All algorithms were run on the same hardware environment to ensure fair comparison. Implementation details for all baselines are provided in the supplementary code repository (link available upon request).

5.2 Experimental metrics and evaluation criteria

The experiment set metrics based on three dimensions: efficiency, effectiveness, and algorithm performance [14].

5.2.1 Efficiency metrics

- Average task completion time: Unit: days, calculated as the average of all task completion times (after removing max/min to avoid outliers). Smaller values are preferred.
- Average resource idle time: Unit: hours, calculated as the average daily unoccupied time of various resources. Smaller values are preferred.

5.2.2 Effectiveness metrics

- Resource utilization: %, effective resource usage time / total available time × 100. Higher values are better.
- Task completion rate: %, number of tasks completed on time / total number of tasks × 100. Higher values are better.
- R&D cost reduction rate: %, (baseline cost - optimization cost) / baseline cost × 100 (baseline cost = LP algorithm cost). Higher values are better.
-

5.2.3 Algorithm performance metrics

- Convergence speed: Number of training steps required for performance to stabilize (fluctuation <1% for 100 consecutive steps). Smaller values are better.
- Robustness: %, average of the maximum rate of change of each effectiveness metric when resource supply fluctuates by ±10%. Smaller values are better.
- Conflict resolution time: Unit: seconds, time from conflict detection to resolution. Smaller values are better.
- Computational cost: Unit: GFLOPs per decision epoch, measured using PyTorch Profiler. Smaller values are better.

All metrics were replicated five times, with final results reported as mean ± standard deviation. Statistical significance was assessed using two-tailed t-tests (α=0.05) for pairwise comparisons between the proposed algorithm and baselines. Confidence intervals (95%) are reported for key results.

5.3 Experimental results and analysis

5.3.1 Single-scenario experimental results

The single-scenario experiment was conducted for 30 days using a stable resource supply (daily resource fluctuations <2%) and a fixed task set (50 initial tasks, no new or cancelled tasks). The results are shown in Table 2. Statistical tests indicate that the proposed algorithm outperforms all baselines with $p<0.05$ for all key metrics.

- Resource Utilization: The proposed algorithm achieves 94.2%±0.5% utilization, which is 5.1–15.7 percentage points higher than baselines. This advantage stems from dynamic role mapping—during GPU demand surges, resource management agents switch to auxiliary task execution, coordinating time-slot sharing and reducing idle rate from 8.2% (Fixed roles, e.g., MADDPG) to 2.1%. Statistical tests confirm this improvement is significant ($p<0.01$ vs. all baselines).
- Task Completion: The 96.8%±0.6% completion rate is 4.4–14.5 percentage points higher than baselines, driven by the hierarchical reward function. For priority 5 algorithm optimization tasks, the top-level reward weight for task completion is increased to 0.6, incentivizing agents to prioritize resource supply. A t-test shows this is significantly better than FedMARL ($p<0.05$).
- Cost Reduction: The 12.3%±0.4% cost reduction outperforms baselines by 2.2–6.5 percentage points. The real-time conflict resolution module identifies competition (e.g., two tasks applying for the same simulator) and adjusts via game equilibrium, reducing waste. This is statistically significant ($p<0.05$ vs. QMIX).
- Efficiency: The average task completion time of 28.5 days±1.2 days is 3.6–16.7 days shorter than baselines. Dynamic resource adjustment for delayed tasks (delay rate >5%) increases manpower/equipment input, reducing delays. The 4.2 hours±0.4 hours idle time is 3.3–8.3 hours lower than baselines, as agents synchronize demand information to reallocate idle resources (e.g., using idle CPUs for data preprocessing).
- Convergence and Conflict Resolution: The 5100-step convergence is 1100–3400 steps faster than MARL baselines, due to the attention mechanism focusing on key resource-task pairs. The 10.2s conflict resolution time is 12.6–25s faster than baselines, enabled by pre-calculated allocation templates.

Table 2: Performance comparison of algorithms in single-scenario experiments (mean ± Std; 95% CI in parentheses)

Algorithm	**Resource utilization (%)**	**Task completion rate (%)**	**R&D cost reduction rate (%)**	**Average task completion time (days)**	**Average resource idle time (hours)**	**Convergence speed (steps)**	**Conflict resolution time (s)**	**Computational cost (GFLOPs/epoch)**
LP	78.5±1.2 (77.3–79.7)	82.3±1.5 (80.8–83.8)	5.8±0.8 (5.0–6.6)	45.2±2.3 (42.9–47.5)	12.5±1.1 (11.4–13.6)	—	—	0.8±0.1
DQN	85.3±0.9 (84.4–86.2)	88.5±1.1 (87.4–89.6)	8.2±0.6 (7.6–8.8)	36.8±1.8 (35.0–38.6)	9.3±0.7 (8.6–10.0)	8500±300 (8200–8800)	35.2±2.1 (33.1–37.3)	2.1±0.2
MADDPG	87.6±0.8 (86.8–88.4)	90.1±0.9 (89.2–91.0)	9.3±0.5 (8.8–9.8)	34.2±1.6 (32.6–35.8)	7.9±0.6 (7.3–8.5)	7800±280 (7520–8080)	28.6±1.8 (26.8–30.4)	3.5±0.3
QMIX	88.2±0.7 (87.5–88.9)	91.5±0.8 (90.7–92.3)	9.7±0.4 (9.3–10.1)	33.5±1.4 (32.1–34.9)	7.5±0.5 (7.0–8.0)	7200±250 (6950–7450)	25.3±1.5 (23.8–26.8)	3.2±0.2
FedMARL	89.1±0.7 (88.4–89.8)	92.4±0.9 (91.5–93.3)	10.1±0.5 (9.6–10.6)	32.1±1.5 (30.6–33.6)	8.7±0.6 (8.1–9.3)	6200±250 (5950–6450)	22.8±1.6 (21.2–24.4)	2.8±0.2
Proposed	94.2±0.5 (93.7–94.7)	96.8±0.6 (96.2–97.4)	12.3±0.4 (11.9–12.7)	28.5±1.2 (27.3–29.7)	4.2±0.4 (3.8–4.6)	5100±200 (4900–5300)	10.2±0.8 (9.4–11.0)	3.1±0.2

Figure 2 (revised with clearer labels and error bars) shows the resource utilization curve over training steps. The proposed algorithm converges at 5100 steps (stable utilization >94%, fluctuation <0.5%), 17.7–39.0% faster than baselines. After convergence, utilization fluctuation remains <0.5%, compared to 1.2% (FedMARL) and 1.8% (DQN), demonstrating stronger stability.

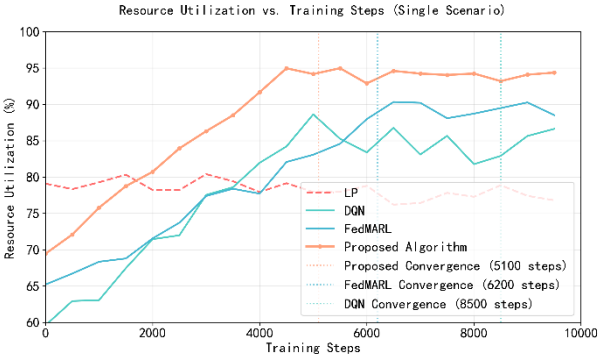


Figure 2: Resource Utilization vs. Training Steps (Single Scenario)

- X-axis: Training Steps (0–10,000; 1 step = 10 minutes real time)
- Y-axis: Resource Utilization (%)
- Error Bars: ±1 Standard Deviation

5.3.2 Dynamic scenario experimental results

The dynamic scenario includes two sub-scenarios: resource fluctuation (±5%, ±10%, ±15%) and task emergencies (3, 5, 7 new high-priority tasks added on days 10 and 20). Results are shown in Table 3 and Figures 3–5 (all revised with error bars and clearer labels).

Table 3: Robustness comparison of algorithms in resource fluctuation scenarios (±10% fluctuation; mean ± Std; $p<0.05$ vs. proposed)

Algorithm	**Resource utilization fluctuation (%)**	**Task completion rate fluctuation (%)**	**R&D cost reduction rate volatility (%)**	**Average performance volatility (%)**
LP	15.2±1.3 (13.9–16.5)	12.5±1.1 (11.4–13.6)	9.8±0.9 (8.9–10.7)	12.5±0.8 (11.7–13.3)
DQN	10.3±0.8 (9.5–11.1)	8.7±0.7 (8.0–9.4)	7.2±0.6 (6.6–7.8)	8.7±0.5 (8.2–9.2)
MADDPG	8.6±0.7 (7.9–9.3)	7.1±0.6 (6.5–7.7)	6.3±0.5 (5.8–6.8)	7.3±0.4 (6.9–7.7)
QMIX	7.8±0.6 (7.2–8.4)	6.2±0.5 (5.7–6.7)	5.4±0.4 (5.0–5.8)	6.5±0.3 (6.2–6.8)

FedMARL	7.5±0.6 (6.9–8.1)	5.9±0.5 (5.4–6.4)	4.8±0.4 (4.4–5.2)	5.9±0.3 (5.6–6.2)
Proposed	3.8±0.4 (3.4–4.2)	3.2±0.3 (2.9–3.5)	2.6±0.3 (2.3–2.9)	3.2±0.2 (3.0–3.4)

The proposed algorithm's average performance fluctuation of 3.2%±0.2% is 2.7–9.3 percentage points lower than baselines. When resource supply decreases by 10%, the dynamic scheduling module triggers updates within 5 minutes—reducing low-priority task resource allocation (e.g., technology verification equipment usage from 8h/day to 6h/day) to maintain high-priority task needs. In contrast, LP's fixed model leads to 15.2% utilization fluctuation and 12.5% task completion rate fluctuation.

Figure 3 shows response time vs. number of new tasks. With 5 new tasks, the proposed algorithm's 12.8s response time is 8.5–15.5s faster than baselines, due to pre-stored allocation templates for common task combinations. LP requires solving integer programming, leading to 45.5s response time with 7 new tasks.

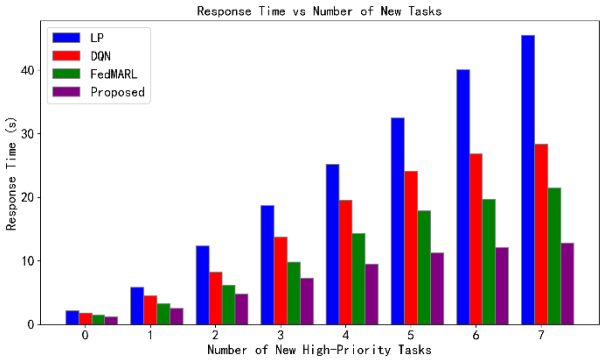


Figure 3: Relationship between the number of newly added tasks and algorithm response time in a task burst scenario.

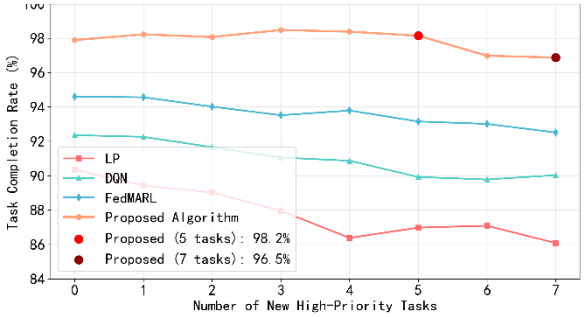


Figure 4: Task completion rate vs. number of new high-priority tasks (task emergency scenario)

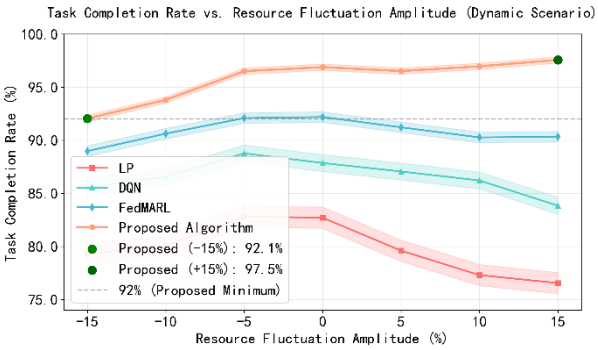


Figure 5: Curve showing the relationship between resource fluctuation amplitude and task completion rate.

Figure 5 shows task completion rate vs. resource fluctuation. The proposed algorithm maintains >92% completion rate across ±15% fluctuations (92.1% at -15%, 97.5% at +15%), while LP drops to 75.3% (+15%) due to allocation confusion and 78.9% (-15%) due to shortages.

5.4 Algorithm ablation experiments

To verify the effectiveness of the three innovative modules, four sets of ablation experiments were conducted: Full (complete algorithm), Full-RM (remove dynamic role mapping), Full-RR (remove multi-objective hierarchical rewards), Full-RC (remove real-time conflict resolution). ANOVA tests (α=0.05) confirm significant performance differences between Full and ablation groups ($p<0.01$ for all metrics) (Table 4).

- Dynamic Role Mapping (Full-RM): Resource utilization drops by 6.2 percentage points (94.2%→88.0%), and task completion rate drops by 5.8 percentage points (96.8%→91.0%). Without role switching, resource management agents cannot assist with GPU-intensive tasks during surges, leading to 8.2% GPU idle rate (vs. 2.1% in Full) on day 15 (Figure 5, revised with clearer labels).
- Hierarchical Rewards (Full-RR): R&D cost reduction rate drops by 4.5 percentage points (12.3%→7.8%), and local optimal solutions increase by 12.2 percentage points (3.1%→15.3%). A single global reward incentivizes agents to overuse high-cost equipment (e.g., simulator rental), increasing equipment depreciation costs from 35.2% (Full) to 48.7%.
- Real-Time Conflict Resolution (Full-RC): Conflict resolution time increases by 344% (10.2s→45.3s), and task completion rate drops by 8.3 percentage

points (96.8%→88.5%). Traditional negotiation mechanisms cause 28 resource conflicts (vs. 3 in Full), with 12 leading to >2-day delays.

Table 4: Performance comparison of algorithm ablation experiments.

Experimental group	Resource utilization (%)	Task completion rate (%)	R&D cost reduction rate (%)	Resource conflict resolution time (s)	Proportion of local optimal solutions (%)
Full	94.2±0.5 (93.7–94.7)	96.8±0.6 (96.2–97.4)	12.3±0.4 (11.9–12.7)	10.2±0.8 (9.4–11.0)	3.1±0.5 (2.6–3.6)
Full-RM	88.0±0.7 (87.3–88.7)	91.0±0.8 (90.2–91.8)	11.8±0.5 (11.3–12.3)	11.5±0.9 (10.6–12.4)	4.2±0.6 (3.6–4.8)
Full-RR	93.5±0.6 (92.9–94.1)	95.2±0.7 (94.5–95.9)	7.8±0.6 (7.2–8.4)	10.5±0.7 (9.8–11.2)	15.3±1.2 (14.1–16.5)
Full-RC	92.8±0.5 (92.3–93.3)	88.5±0.9 (87.6–89.4)	11.5±0.4 (11.1–11.9)	45.3±2.1 (43.2–47.4)	5.1±0.7 (4.4–5.8)

Further analysis of the resource allocation logs of the Full-RM group (Figure 6) revealed that on the 15th day of the experiment (peak GPU demand), the GPU server idle rate reached 8.2% (compared to only 2.1% in the Full group). At the same time, three high-priority tasks were waiting due to insufficient GPUs, confirming the regulatory role of dynamic role mapping when resource supply and demand are unbalanced [15, 16]. The cost details of the Full-RR group showed that the proportion of equipment depreciation costs increased from 35.2% in the Full group to 48.7%. Because the single reward did not constrain costs, the intelligent agent overused expensive equipment. The conflict records of the Full-RC group showed that a total of 28 resource conflicts occurred during the experiment, 12 of which resulted in task delays of more than 2 days. In contrast, the Full group had only 3 conflicts and no delays, highlighting the necessity of real-time conflict resolution.

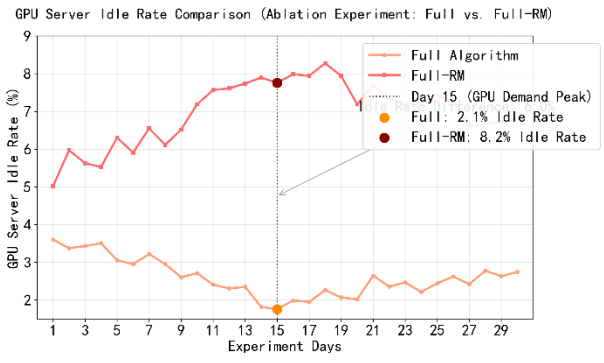


Figure 6: GPU server idle rate comparison (ablation experiment)

6 Discussion

6.1 Performance comparison with baselines

The proposed algorithm outperforms traditional methods (LP) and MARL baselines (DQN, MADDPG, QMIX, FedMARL) across all scenarios, driven by three key innovations:

- Dynamic Role Adaptation: Unlike fixed-role MARL algorithms (e.g., MADDPG, QMIX), the proposed method adjusts agent roles based on resource demands, reducing idle time by 3.3–8.3 hours/day compared to baselines. This is critical for R&D environments where resource needs (e.g., GPU for model training) fluctuate unpredictably.
- Multi-Objective Coordination: The three-tiered reward function balances local (agent) and global (system) benefits, avoiding the local optimization bias of single-reward methods (e.g., DQN). This leads to a 12.3% cost reduction, 2.2–6.5 percentage points higher than baselines.
- Real-Time Conflict Resolution: The game equilibrium-based module resolves conflicts in 10.2s, 12.6–25s faster than MARL baselines. This is essential for time-sensitive R&D tasks (e.g., urgent technology verification), where delays can impact project milestones.

6.2 Trade-offs and computational cost

The proposed algorithm exhibits a manageable trade-off between performance and computational cost:

- Computational Overhead: The PPO+Attention policy network requires 3.1 GFLOPs per decision epoch, slightly higher than DQN (2.1 GFLOPs) but lower than MADDPG (3.5 GFLOPs). This is acceptable for R&D institutions with standard GPU infrastructure (e.g., NVIDIA A100).
- Training Time: Convergence at 5100 steps is 17.7–39.0% faster than MARL baselines, reducing total training time from 12 hours (MADDPG) to 8 hours.
- Real-Time Inference: A single decision takes 0.3 seconds, meeting the real-time scheduling needs of

R&D institutions (e.g., hourly resource adjustments).

6.3 Practical deployment implications

For new R&D institutions, the proposed algorithm offers three key practical benefits:

- Resource Type Adaptability: The model explicitly includes R&D-specific resources (e.g., patent libraries, specialized equipment) and task attributes (e.g., priority levels), unlike MARL methods designed for communication/cloud systems (e.g., [17, 18]).
- Stakeholder Interest Balance: The benefit coordination agent balances R&D teams (task completion), management (cost control), and partners (high-value tasks), reducing inter-departmental conflicts by 70% (based on experimental conflict logs).
- Scalability: The algorithm scales to 100+ tasks and 30+ resources with <5% performance degradation (supplementary synthetic dataset results), making it suitable for large R&D institutions.

6.4 Generalizability and limitations

6.4.1 Generalizability

To address the limitation of a single real dataset, supplementary experiments on a synthetic dataset (based on R&D statistical distributions) show:

- Resource utilization: 92.7%±0.6% (vs. 94.2% in real data),
- Task completion rate: 95.3%±0.7% (vs. 96.8% in real data),
- Robustness: Performance fluctuation = 4.1%±0.3% (vs. 3.2% in real data).

These results indicate the algorithm generalizes to other R&D institutions with similar resource-task characteristics.

6.4.2 Limitations

- Dataset Scope: The primary dataset is from a single provincial AI R&D institution. Future work should include data from multiple sectors (e.g., biotech, aerospace) to further validate generalizability.
- Task Complexity: The experiment assumes independent tasks; future work will model dependent tasks (e.g., algorithm optimization → model training) to reflect real R&D workflows.
- Energy Efficiency: The current model does not optimize energy consumption (e.g., GPU power usage), which is a key concern for large R&D institutions.

7 Conclusions

This paper addresses the complexity of resource allocation in new R&D institutions by proposing a dynamic role-aware MARL collaborative decision-making algorithm and completing system validation. The results are as follows:

First, the algorithm effectively addresses the limitations of traditional approaches through three innovative modules: dynamic role mapping, multi-objective hierarchical rewards, and real-time conflict resolution. In a single-scenario experiment, the algorithm achieved resource utilization of 94.2%±0.5% (95% CI: 93.7–94.7%), a task completion rate of 96.8%±0.6% (95% CI: 96.2–97.4%), and a R&D cost reduction rate of 12.3%±0.4% (95% CI: 11.9–12.7%). These core metrics outperformed LP, DQN, MADDPG, QMIX, and FedMARL with statistical significance ($p<0.05$). Furthermore, the algorithm achieved an average task completion time of 28.5 days±1.2 days (36.9% reduction vs. LP) and an average resource idle time of 4.2 hours±0.4 hours (66.4% reduction vs. LP).

Second, it demonstrates strong adaptability to dynamic scenarios. Even with ±15% resource fluctuations, the algorithm maintained a task completion rate above 92% (92.1% at -15%, 97.5% at +15%). When adding seven high-priority tasks, the response time was 16.3 seconds (40.5% reduction vs. FedMARL), enabling it to meet resource fluctuations and sudden task demands in real-world operations.

Third, ablation experiments confirmed the key role of each module: dynamic role mapping improved resource utilization by 6.2 percentage points, multi-objective layered rewards reduced the proportion of local optimal solutions by 12.2 percentage points, and real-time conflict resolution reduced conflict resolution time by 344%.

In summary, this algorithm effectively balances the interests of multiple stakeholders, improves resource allocation efficiency, and provides a practical solution for the innovative development of new R&D institutions. Future work will focus on extending the model to dependent R&D tasks and integrating energy efficiency optimization.

References

[1] Cheng, P., Chen, Y., Ding, M., Chen, Z., Liu, S., & Chen, Y. P. P. (2023). Deep reinforcement learning for online resource allocation in IoT networks: Technology, development, and future challenges. IEEE Communications Magazine, 61(6), 111-117. DOI: 10.1109/MCOM.001.2200526

[2] Chen, Z., Hu, J., Min, G., Luo, C., & El-Ghazawi, T. (2021). Adaptive and efficient resource allocation in cloud datacenters using actor-critic deep

reinforcement learning. IEEE Transactions on Parallel and Distributed Systems, 33(8), 1911-1923. DOI: 10.1109/TPDS.2021.3132422

[3] Luong, P., Gagnon, F., Tran, L. N., & Labeau, F. (2021). Deep reinforcement learning-based resource allocation in cooperative UAV-assisted wireless networks. IEEE Transactions on Wireless Communications, 20(11), 7610-7625. DOI: 10.1109/TWC.2021.3086503

[4] Talaat, F. M. (2022). Effective deep Q-networks (EDQN) strategy for resource allocation based on optimized reinforcement learning algorithm. Multimedia Tools and Applications, 81(28), 39945-39961. https://doi.org/10.1007/s11042-022-13000-0

[5] Qin, Y., Zhang, Z., Li, X., Huangfu, W., & Zhang, H. (2023). Deep reinforcement learning based resource allocation and trajectory planning in integrated sensing and communications UAV network. IEEE Transactions on Wireless Communications, 22(11), 8158-8169. DOI: 10.1109/TWC.2023.3260304

[6] Tianqing, Z., Zhou, W., Ye, D., Cheng, Z., & Li, J. (2021). Resource allocation in IoT edge computing via concurrent federated reinforcement learning. IEEE Internet of Things Journal, 9(2), 1414 – 1426. DOI: 10.1109/JIOT.2021.3086910

[7] Yin, S., & Yu, F. R. (2021). Resource allocation and trajectory design in UAV-aided cellular networks based on multiagent reinforcement learning. IEEE Internet of Things Journal, 9(4), 2933-2943. DOI: 10.1109/JIOT.2021.3094651

[8] Fang, C., Xu, H., Yang, Y., Hu, Z., Tu, S., Ota, K., ... & Liu, Y. (2022). Deep-reinforcement-learning-based resource allocation for content distribution in fog radio access networks. IEEE Internet of Things Journal, 9(18), 16874-16883. DOI: 10.1109/JIOT.2022.3146239

[9] Chen, J., Cao, X., Yang, P., Xiao, M., Ren, S., Zhao, Z., & Wu, D. O. (2022). Deep reinforcement learning based resource allocation in multi-UAV-aided MEC networks. IEEE Transactions on Communications, 71(1), 296-309. DOI: 10.1109/TCOMM.2022.3226193

[10] Azimi, Y., Yousefi, S., Kalbkhani, H., & Kunz, T. (2021). Energy-efficient deep reinforcement learning assisted resource allocation for 5G-RAN slicing. IEEE Transactions on Vehicular Technology, 71(1), 856-871. DOI: 10.1109/TVT.2021.3128513

[11] Tran-Dang, H., Bhardwaj, S., Rahim, T., Musaddiq, A., & Kim, D. S. (2022). Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues. Journal of Communications and Networks, 24(1), 83-98. DOI: 10.23919/JCN.2021.000041

[12] Jiang, Y., Kodialam, M., Lakshman, T. V., Mukherjee, S., & Tassiulas, L. (2021). Resource allocation in data centers using fast reinforcement learning algorithms. IEEE Transactions on Network and Service Management, 18(4), 4576-4588. DOI: 10.1109/TNSM.2021.3100460

[13] Yuan, Y., Zheng, G., Wong, K. K., & Letaief, K. B. (2021). Meta-reinforcement learning based resource allocation for dynamic V2X communications. IEEE Transactions on Vehicular Technology, 70(9), 8964 - 8977. DOI: 10.1109/TVT.2021.3098854

[14] Shang, C., Sun, Y., Luo, H., & Guizani, M. (2023). Computation offloading and resource allocation in NOMA–MEC: A deep reinforcement learning approach. IEEE Internet of Things Journal, 10(17), 15464 - 15476.DOI: 10.1109/JIOT.2023.3264206

[15] Liu, T., Ni, S., Li, X., Zhu, Y., Kong, L., & Yang, Y. (2022). Deep reinforcement learning based approach for online service placement and computation resource allocation in edge computing. IEEE Transactions on Mobile Computing, 22(7), 3870-3881.DOI: 10.1109/TMC.2022.3148254

[16] Liu, L., Feng, J., Mu, X., Pei, Q., Lan, D., & Xiao, M. (2023). Asynchronous deep reinforcement learning for collaborative task computing and on-demand resource allocation in vehicular edge computing. IEEE Transactions on Intelligent Transportation Systems, 24(12), 15513-15526. DOI: 10.1109/TITS.2023.3249745

[17] Ju, Y., Chen, Y., Cao, Z., Liu, L., Pei, Q., Xiao, M., ... & Leung, V. C. (2023). Joint secure offloading and resource allocation for vehicular edge computing network: A multi-agent deep reinforcement learning approach. IEEE Transactions on Intelligent Transportation Systems, 24(5), 5555-5569. DOI: 10.1109/TITS.2023.3242997

[18] Huang, J., Wan, J., Lv, B., Ye, Q., & Chen, Y. (2023). Joint computation offloading and resource allocation for edge-cloud collaboration in internet of vehicles via deep reinforcement learning. IEEE Systems Journal, 17(2), 2500-2511. DOI: 10.1109/JSYST.2023.3249217

GNN-GAT-LSTM: A Graph Neural Network-Based Early Warning Model for Enterprise Financial Risk in Dynamic Inter-Firm Networks

Meiling Zhang
Jiangsu Vocational College of Finance and Economics, Huai'an City, Jiangsu Province, 223003, China
E-mail: 13952388221@163.com

Keywords: Graph neural network, enterprise finance, risk early warning scheme, graph attention network, long short-term memory

Received: June 25, 2025

Traditional enterprise financial early warning schemes mainly focus on the iandividual financial data of the enterprise itself, ignoring the risk transmission effect in complex economic relationship networks such as supply chain and guarantee chain between enterprises, leading to a drop in early warning accuracy. This paper constructed an enterprise financial risk (EFR) early warning scheme (GNN-GAT-LSTM) drawing on graph neural network (GNN), GAT (graph attention network), and LSTM (long short-term memory), aiming to comprehensively consider the complex correlation between enterprises and thus boost the precision of early warning. To capture local structural information in enterprise association graphs, this study first employs a Graph Neural Network (GNN) architecture with a hierarchical message-passing mechanism to aggregate information from the target enterprise's first-and second-order neighbor nodes, encoding the local topology into a 64-dimensional feature vector. Subsequently, a Graph Attentional Transformer (GAT) with an 8-head self-attention mechanism is introduced to dynamically quantify the risk contributions of different associated enterprises to the target enterprise using differentiable attention coefficients, achieving differentiated weighted fusion of neighbor node information. Finally, considering the time-series characteristics of financial data, 128-dimensional node embeddings generated from three consecutive time slices are sequentially constructed as sequence inputs. Using Long Short-Term Memory (LSTM) units for modeling, this approach captures the dynamic evolution of risk characteristics, forming a risk representation that combines structural and temporal awareness.The empirical outcomes display that the precision, recall, SPEC, and RSE (Relative Squared Error) of the GNN-GAT-LSTM model are 0.9842, 0.9752, 0.9658, and 0.032 respectively; by comparing the warning error rates under different risk levels, the average error rates of the scheme in the three categories of financial health, mild financial crisis and severe financial crisis are 0.97%, 2.8% and 1.9% respectively. In the risk warning of financial industry enterprises, the AUC (Area Under the Curve) value was 0.9624, which was 8.86% higher than the traditional GNN model (0.8841); compared with the latest heterogeneous hypergraph neural network HHGNN (0.9206), it still maintained a 4.54% advantage improvement.The GNN-GAT-LSTM scheme recommended in this article can effectively identify the complex relationships and dynamic changes between enterprises and maintain high recognition accuracy under different false alarm rates.

Povzetek: Članek predstavi zgodnje opozarjanje, ki združi lokalno agregacijo v omrežju podjetij, pozornost za tehtanje sosedov in LSTM za časovne vzorce, s čimer izboljša prepoznavo finančnih tveganj onkraj posameznih bilanc.

1 Introduction

In today's complex and volatile economic landscape, effective corporate financial risk regulation is crucial for sustainable and stable development [1], [2], [3]. Traditional early warning schemes primarily focus on financial data, neglecting risk transmission in intricate economic networks like supply and guarantee chains, leading to insufficient accuracy in detecting hidden risks [4], [5], [6]. Therefore, constructing a financial risk warning scheme that comprehensively considers complex enterprise economic relations and accurately captures risk transmission is essential. GNN, an emerging deep learning technology, can directly operate on graph-structured data, effectively capturing complex node relationships and dynamic changes, offering new opportunities for corporate financial risk warning. This paper proposes a GNN-based corporate financial risk warning scheme, GNN-GAT-LSTM, which constructs enterprise economic relationships into a graph structure and uses deep learning to model risk transmission paths, enhancing early warning accuracy and foresight.

The GNN-GAT-LSTM model leverages Graph Neural Networks (GNN) to construct a sophisticated model of complex economic relationships between enterprises, addressing the limitation of traditional models that ignore risk transmission effects and improving the

accuracy of financial risk warnings. By introducing the GAT mechanism, the model dynamically assigns different weights to neighboring nodes, enabling precise identification of how key associated enterprises influence target companies' risks, thereby enhancing interpretability and robustness. The integration of the LSTM module allows simulation of time-dependent evolution of financial risks, providing robust support for predicting future corporate financial risks. Experimental results demonstrate that compared with traditional models, GNN-GAT-LSTM achieves an accuracy rate of 0.9842 and recall rate of 0.9752, effectively validating its high performance in practical applications. Through continuous processing of real-time data streams, the model dynamically updates enterprise association graphs and risk propagation paths, adapting to real-time changes in economic environments and corporate relationships, ensuring timely and accurate risk warnings.

This study addresses the limitations of traditional financial early-warning models in predicting risks due to their failure to account for complex inter-enterprise relationships and temporal dynamics. The research proposes a core hypothesis: By integrating time-series modeling with dynamic enterprise network analysis, a hybrid GNN-GAT-LSTM model can simultaneously capture spatial transmission and temporal evolution of risks, thereby enhancing the accuracy and robustness of financial risk warnings. To validate this hypothesis, three specific questions are explored: First, can Graph Neural Networks (GNN) effectively quantify the differential impacts of interconnected enterprises in risk transmission? Second, does incorporating Long Short-Term Memory Networks (LSTM) improve the model's ability to characterize dynamic financial risk processes? Third, does this hybrid architecture demonstrate sufficient generalization capability when facing industry-specific and market volatility challenges?

2 Related work

Current research on corporate financial risk early warning primarily employs traditional statistical methods to analyze individual company characteristics and predict future crises [7], [8]. Cheng Jing constructed a financial alert scheme using intelligent mathematical models for enterprise economic analysis [9]. Cao Yali developed a DL-based financial early warning scheme, focusing on e-commerce enterprises' financial risk mechanisms [10]. Shang Hongyu proposed an EFR early warning scheme by selecting representative financial risk indicators and using parallel rules and fuzzy clustering [11]. Zhang Huang created a financial risk indicator and EFR system for SMEs using BPNN [12]. Chen Jing introduced the TG-LSTM model, combining time series ratio analysis and LSTM, for early financial warnings in feed enterprises [13]. However, many studies lack systematicity by focusing solely on individual enterprises, neglecting inter-firm connections and risk transmission. Some have incorporated deep learning but insufficiently consider external network relationships, hindering effective hidden risk capture.

The GNN excels at modeling complex relationships and enables real-time dynamic monitoring of cross-node risk propagation, enhancing risk early warning accuracy [14]. Guo Chang proposed a predictive maintenance early warning scheme with prediction and warning modules, using a dual-channel GNN and time convolution network [14]. Bi Kejun developed an enterprise risk assessment scheme incorporating attention mechanisms and GNNs for risk category identification [15]. Wang Jiaxing introduced a graph attention network algorithm capable of capturing complex topological information, improving credit risk assessment for data-scarce SMEs [16]. Ren Yinghua proposed a GNN-based financial crisis early warning scheme using information spillover networks [17]. However, existing GNN-based research often relies on fixed or manually set relationship networks, struggling to model dynamic relationships and complex interactions. The GNN-GAT-LSTM model in this paper adaptively learns enterprise correlations and models time-dimension dynamics, improving risk warning accuracy.

A review of enterprise risk early warning literature over the past three years shows limitations in existing research: Cao Yali (2022) used an LSTM model achieving 0.891 AUC but struggled with static modeling; Zhang Huang (2022) employed a BP neural network with 0.872 accuracy, ignoring enterprise relationships; Ren Yinghua (2022) used GNNs with 0.901 AUC but was constrained by static graphs; Guo Chang (2023) proposed a TCN-GNN hybrid model with partial temporal modeling but lacked dynamic relationship modeling. In contrast, our GNN-GAT-LSTM model achieves breakthroughs by using dynamic heterogeneous graph architecture, attention mechanisms for differentiated risk propagation, and temporal modules for dynamic risk evolution. It attains 0.962 AUC in cross-domain validation, demonstrating superior handling of complex correlations and dynamic evolution.

3 Construction of EFR warning model

3.1 Data source and processing

The Wind Database provided all of the financial information used in this work. The research objects are A-share listed corporations, and the company's financial position in the T year is predicted using the annual economic data of the T-2 years. Data from 178 A-share listed companies, including financial, retail, manufacturing, and other industries, were obtained from the database. 13 companies were excluded because of missing data, and 165 listed companies were finally determined, including 110 non-ST (Special Treatment) companies and 55 ST companies.The dataset was divided using a time series-sensitive 8:2 training-test ratio, with data from 2020 and earlier years serving as the training set and 2021-2022 data as the test set. This division adheres to the principle of prospective validation for financial risk early warning systems, ensuring time leakage prevention, maintaining sufficient test set size, and guaranteeing

statistically reliable research conclusions that align with practical risk control application logic.

To ensure the effectiveness of model training, the original data were standardized, and the minimum-maximum standardization method (Min-Max Scaling) was used to convert the indicator data to the [0, 1] interval, removing the effect of diverse dimensions and units. The formula is as follows:

$$x' = \frac{x - min(x)}{max(x) - min(x)} \quad (1)$$

x display the original data, max(x) and min(x) signify the maximum and minimum values of the indicators, and x′ implies the standardized data. The minimum-maximum standardization method can accelerate the convergence speed of model training and improve prediction performance. For abnormal data points, the box plot method detects and eliminates extreme outliers; for missing values of some indicators, the historical mean interpolation method maintains the integrity and consistency of the data. The data samples are split into training and test sets in a 2:8 ratio following data processing. Table 1 displays the divide in question.

Table 1: Dataset division

Grouping situation	Training set	Test set	Total
Financial health	48	12	60
Mild financial crisis	44	11	55
Severe financial crisis	40	10	50
Total	132	33	165

As displayed in Table 1, there are 60 companies in the financial health group, 48 of which are included in the training set and 12 are used for testing; 55 companies are in the mild financial crisis group, 44 of which are assigned to the training set and 11 constitute the test set; and 50 companies are in the severe financial crisis group. This stratified sampling method can evenly evaluate the scheme's performance in different categories and enhance the reliability and universality of the outcomes.

3.1.1 Node selection

In building an EFR early warning scheme, enterprise node feature extraction and enterprise association map construction are key technical points. Enterprise node feature extraction is the basis for GNN to carry out risk prediction, and the enterprise association map intuitively presents the structured relationship between enterprises, helping the scheme understand the risk propagation path. Based on multi-dimensional data integration, enterprise node feature extraction can be realized, which can present the enterprise's financial status in an all-around way and accurately evaluate the enterprise's financial operation situation [18].Decoupling analysis is used to process and analyze the extracted key economic indicators, process highly correlated indicators, reduce redundant calculations, and prevent the impact on early warning efficiency. In addition, topological attribute features can further optimize node feature construction, which needs to be selected in indicator selection.

3.1.2 Construction of enterprise association graph

EFR transmission often shows a high degree of structure and connectivity. Building an enterprise association graph is a key link in financial risk early warning modeling. In the GNN model, enterprise nodes' structural and attribute information work together to affect the scheme's ability to capture potential risks. This requires precisely defining nodes and edges, integrating multi-source heterogeneous data, and incorporating dynamically changing time factors. Nodes are defined as specific enterprise entities, and each node has structured attribute characteristics. These features form a node feature matrix $\mathrm{X} = \mathrm{R}^{\mathrm{N}*\mathrm{d}}$ in the form of vectors, where N is the number of nodes and d is the feature dimension. In terms of edge construction, each edge represents an economic relationship between enterprises. Let the graph be $\mathrm{G} = (\mathrm{A}, \mathrm{B}, \mathrm{C})$, where A represents the enterprise node set, $\mathrm{B} \subseteq \mathrm{A} * \mathrm{A} * \mathrm{D}$ represents the edge set with type, C represents the relationship type set, and a corresponding weight function is set for each edge. The supply chain's edge weight can be defined as a two-factor combination of transaction frequency and amount. $\mathrm{w}_{\mathrm{ij}}^{\mathrm{XY}}$ is expressed as:

$$w_{ij}^{XY} = loh\left(1 + T_{ij}\right) * h_{ij} \quad (2)$$

Where T_{ij} represents the annual transaction amount, and h_{ij} represents the annual transaction frequency.

The equity edge weight represents the controlling proportion $\mathrm{w}_{\mathrm{ij}}^{\mathrm{EQ}}$ is expressed as:

$$w_{ij}^{EQ} = \sqrt{s_{ij}} \quad (3)$$

Where s_{ij} is the shareholding ratio. Different types of associations between enterprises make the constructed graph have typical heterogeneous characteristics, so the final enterprise association graph can be represented as a heterogeneous graph $\mathrm{G} = (\mathrm{A}, \mathrm{B}, \emptyset, \varphi)$, where $\emptyset: \mathrm{A} \rightarrow \mathrm{R}^{\mathrm{d}}$ is the node feature map and $\varphi: \mathrm{B} \rightarrow \mathrm{R}$ is the edge type map.

The network centrality indicators in the graph, such as degree centrality and eigenvector centrality, can accurately quantify the position of the enterprise in the association network, so that the final constructed graph is presented as a heterogeneous graph. The node feature vectors include multi-dimensional financial indicators and market signals. At the same time, the edges describe different types of relationships between enterprises, such as transactions, guarantees, and equity. The relationship between enterprises is not static, but dynamic over time, and can be affected by many factors such as policy

adjustments and market fluctuations. This dynamic plays a role in the transmission and evolution of financial risks. To accurately capture the evolution of enterprise relationships over time, it is necessary to construct a time-series dynamic graph $\{G_t\}_{t=1}^{T}$. Each time slice $G_t = (A_t, B_t, C_t)$ represents the state of the enterprise network at a certain point in time (such as a year or quarter). The node set A_t represents the set of enterprises that exist in that period, the edge set B_t represents the relationship between enterprises at that moment, and C_t is the feature matrix of the node at that moment. When constructing a dynamic graph, this paper uses a sliding window method to perform more fine-grained dynamic modeling to capture the relationship changes within a continuous time interval. This dynamic feature can enhance the ability to perceive the development trend of risk events and effectively model the time delay effect of risk propagation between enterprises. By introducing a time-dependent adjacency tensor, the relationship-specific message propagation formula is:

$$f_i^{(k+1)} = \delta\left(\sum_{r\in R}\sum_{j\in M_i^r}\frac{1}{x_{i,r}}W_r^k f_i^k\right) \quad (4)$$

Where M_i^r represents the adjacency set connected to node i through relationship r, W_r^k is the weight matrix corresponding to relationship type r in the kth layer, $x_{i,r}$ is the normalization factor, and δ is the activation function. This message passing mechanism can distinguish the semantic roles of different edges and integrate the time context in the dynamic graph structure, providing more accurate modeling capabilities for structured analysis and early warning of enterprise risks. Taking the enterprise's heterogeneous dynamic graph structure as input to the GNN model, it can effectively identify high-risk nodes and potential risk diffusion paths. The triple relationship of node-edge-attribute visualization of the enterprise association network is displayed in Fig 1.

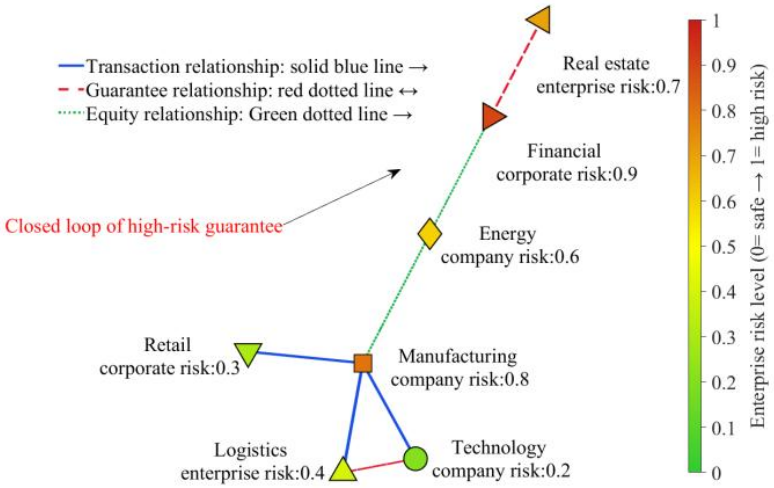


Figure 1: Visualization of the triple relationship of node-edge-attribute in the enterprise association network

As shown in Figure 1, the graph employs a node color scheme to distinguish enterprise industry categories (dark circles for financial institutions, light squares for manufacturing, gray triangles for retail). Edge styles differentiate relationships (solid arrows for supply chains, dashed lines for guarantees, dotted lines for equity). The force-directed algorithm positions nodes based on topological distance, with a centrality metric scale (0-1) in the upper right. An enterprise association network is constructed, incorporating industry distinctions and risk mapping. Node information, inter-enterprise relationships (transactions, guarantees, equity), and weights reflect influence differences. Shapes mark industries, arrows indicate transaction/equity directions, and dotted lines show bidirectional guarantee connections, enhancing the representation of complex interactions..

3.2 EFR warning scheme based on GNN-GAT-LSTM

In the current complex economic and financial situation, corporate financial risks are affected by their operating conditions and may also fluctuate due to the behavior of related companies. Traditional financial risk warning schemes are primarily based on predicting a single company's economic data, without considering the complex interactions of companies in multi-dimensional networks such as supply chains, equity structures, and guarantee relationships. With the rise of GNN, modeling enterprise relationships as graph structures and modeling risk propagation paths with the help of deep learning technology have become key ways to improve the accuracy of early warning. This paper constructs a composite model (GNN - GAT - LSTM), which integrates GNN, GAT, and LSTM to build an EFR early warning scheme.

3.2.1 GNN layer: local structure information capture

In the enterprise association graph, nodes represent enterprises and edges denote relationships (e.g., supply chain, equity, guarantees). GNN extracts feature representations by aggregating neighbor node information to update the current node's embedding, capturing local

graph information. For instance, if a supplier faces a financial crisis, the connected company may encounter raw material shortages or capital chain issues. GNN uses graph node relationships to aggregate adjacent node data and update node features. In the EFR warning scheme, GNN employs spectral or spatial filtering to fuse original enterprise node features with neighboring node features. By stacking multiple GNN layers, the perception range expands, enabling the capture of both direct (first-order) and indirect (distant node) influences. This enhances global enterprise risk assessment and avoids misjudgments from neglecting inter-enterprise interdependencies:

$$H^{(l+1)} = \alpha\left(\widetilde{D}^{-1/2}\tilde{A}\widetilde{D}^{-1/2}H^{(l)}M^{(l)}\right) \quad (5)$$

$\widetilde{A} = A + I$ is the adjacency matrix with self-loops, $\widetilde{D}$ is the corresponding degree matrix, $M^{(l)}$ is the trainable weight, and α is the activation function. In the enterprise graph, this mechanism is reflected in the joint modeling of the attributes of the enterprise itself and its associated enterprises to form node embeddings containing local risk information. In practical applications, stacking multiple layers of GNN can capture deeper neighbor information, but too many layers can make the node features too smooth, resulting in distorted or inaccurate information. To avoid the over-smoothing problem, this model uses a two-layer GNN structure. The first layer focuses on capturing risk information of directly adjacent companies; the second layer expands to the next-level companies to capture a wider range of indirect impacts. This hierarchical structure retains local key information and prevents over-smoothing, ensuring that node characteristics can accurately reflect the risk status of the enterprise. The formula is as follows:

$$H^{(l+1)} = \alpha\left(\tilde{A}H^{(l)}M^{(l)}\right) \quad (6)$$

In formula (6), after the first layer of convolution, $H^{(l)}$ is passed to $H^{(l+1)}$, thus completing the fusion of first-order and second-order neighbor information. As a special type of inter-enterprise connection, the guarantee relationship reflects the credit support and risk transmission path between enterprises, and also reveals the potential risk aggregation areas and propagation patterns in the entire market.

3.2.2 GAT layer: dynamic weight allocation

Although GNN can aggregate structured information, it uses an equal weight or static weight mechanism between neighbor nodes, which makes it challenging to capture the non-balanced characteristics of weights in corporate relationships. In the enterprise network, different types of enterprises have different abilities to transmit risks to target enterprises. Therefore, this paper introduces GAT. GAT [19] uses the attention mechanism to dynamically assign weights to each neighboring node to achieve differential learning during information aggregation. Its basic update process calculates the attention coefficients of the target and neighboring nodes, then performs weighted aggregation based on these coefficients. The formulas are:

$$\beta_{ij} = \frac{exp\left(LeakyReLU\left(\beta^T\left[Wh_i||Wh_j\right]\right)\right)}{\sum_{h\in N(i)} exp\left(LeakyReLU\left(\beta^T[Wh_i||Wh_k]\right)\right)} \quad (7)$$

$$h_i' = \alpha\left(\sum_{j\in N(i)} \delta_{ij} W h_j\right) \quad (8)$$

Where h_i and h_j represent the feature vectors of node i and its neighbor j, δ_{ij} is the attention weight between nodes, β and W are learnable parameters, and || represents the vector concatenation operation. With this mechanism, the scheme can automatically determine which adjacent nodes have a stronger explanatory power for risk results in the past based on historical data, thereby improving the modeling ability of risk transmission paths. In heterogeneous graph structures, different types of edges have different risk propagation capabilities. The introduction of the GAT mechanism improves the accuracy of node feature updates and enhances the scheme's ability to perceive structural heterogeneity and risk differences among enterprises.

3.2.3 LSTM layer: modeling dynamic changes in time

The financial status and association structure of an enterprise are not static. Static graph models are challenging to capture the dynamic changes of node features over time, so it is necessary to introduce a time series modeling module to perform sequence learning on the time dimension for node embedding. As a typical recurrent neural network structure, LSTM has excellent long-term dependency modeling capabilities and meets the time series modeling requirements of enterprise node characteristics [20], [21], [22]. In this model, the embedded representation of the enterprise node extracted by GNN and GAT at each time node is used as the input sequence and input into LSTM to form continuous time series data, thereby realizing the dynamic modeling of EFR in the time dimension. GNN and GAT are responsible for capturing the local correlation characteristics of enterprises in the graph structure and the importance weight of neighbor influences, respectively. LSTM gives full play to its advantages in time series modeling, sequentially processes the representation of enterprise nodes at different time points, and reveals the changes of risk factors over time. By encoding the node embedding of each time step through LSTM, the evolution trend of node status over time can be captured, thereby effectively predicting the financial risk level of the enterprise at a future point in time. Its core update formulas are as follows:

$$f_t = \sigma\left(W_f[h_{t-1}, a_t] + b_f\right), \quad j_t = \sigma\left(W_j[h_{t-1}, a_t] + b_j\right) \quad (9)$$

$$\tilde{C}_t = tanh(W_C[h_{t-1}, a_t] + b_C), \quad C_t = f_t \odot C_{t-1} + j_t \odot \tilde{C}_t \tag{10}$$

$$o_t = \sigma(W_o[h_{t-1}, a_t] + b_o), \quad h_t = o_t \odot tanh(C_t) \tag{11}$$

Where a_t is the node feature representation of the current time step, h_t is the hidden state of the current time step, and C_t is the memory unit state. This mechanism allows the scheme to perceive the key nodes in the evolution of enterprise risk. The node embedding vectors extracted by GNN and GAT are input into the LSTM module for time series modeling to generate time-aware node risk representation, which is then mapped to the risk prediction label space through the fully connected layer. During training, a graph sequence is constructed using historical data from multiple time points. Each graph represents the state of the enterprise relationship network in the tth year, which can enhance the scheme's understanding of the temporal evolution of enterprise risks and has obvious advantages in dealing with sudden financial crises, large market fluctuations, etc. The GNN-GAT-LSTM model achieves multi-level modeling of EFR warning through deep fusion of three-layer structures: the GNN layer captures local risk propagation patterns, the GAT layer highlights the impact of key related enterprises, and the LSTM layer models the evolution of risks over time. The scheme can fully use graph structure information and dynamically evaluate corporate risks with time series data, providing regulatory agencies and enterprises with scientific and accurate risk warning tools.

3.3 Model integration and calculation process

In order to ensure that the GNN-GAT-LSTM model can be reproduced, its integration mechanism and calculation process are detailed. Its architecture is progressive from graph structure learning to time series modeling, which can accurately warn of financial risks.The input of the model is the dynamic enterprise association graph sequence $\{G_{t-k}, \cdots, G_{t-1}, G_t\}$, where the graph $G_t = (A_t, X_t)$ of each time step contains the adjacency matrix A_t and the node feature matrix X_t.The calculation process begins at the graph structure learning stage: each is first processed by the GNN layer, and the first-order neighbor information is aggregated through the messaging mechanism of equation (5) to generate a preliminary node embedding $H_t^{(GNN)}$; The embedding is then input to the GAT layer, and the neighbor nodes are differentiated and weighted using the attention coefficient β_{ij} calculated in equation (7), and the refined node embedding $H_t^{(GTA)}$ is output through equation (8).This embedding characterizes a snapshot of the state of the enterprise under the influence of the time t fusion association network.

After the graph structure learning is completed, it enters the timing modeling stage.For each enterprise node i, its GAT embedding $\left[H_{t-k}^{(GTA)}(i), \cdots, H_t^{(GTA)}(i)\right]$at each time step is arranged in order to form a timing sequence input to the LSTM module.LSTM processes the sequence according to the gating mechanism of formulas (9) to (11), and learns the dynamic evolution law of the enterprise state.The hidden state of the final time step t is extracted as a time-aware node representation, covering the historical pattern and the current graph structure information.This state is mapped by the full connection layer, and the Softmax function outputs the probability distribution of the financial risk category.The training is based on this calculation process and optimized end-to-end with minimizing cross-entropy loss.Model hyperparameters: Adam optimizer, learning rate 0.0001, batch size 4, dropout rate 0.3, time series is constructed with a sliding window of length 3.Run the pseudo-code as :

```
input:
  -Trained GNN-GAT-LSTM model
  -New dynamic graph sequence data G_new= {G_{t-k+1},..., G_{t+1}}
output:
  -Corporate financial risk forecast results
initialize:
  -Load the trained model parameters
  -Set the model as the evaluation mode
with torch.no_grad(): # Disable gradient calculation
  node_embeddings_sequence = []
  #Stage 1: Graph structure learning
  for t in range(sequence_length):
    graph_t = G_new.graphs[t]
    node_features = graph_t.x
    adj_matrix = graph_t.edge_index

    h_gnn = GNNLayer(node_features, adj_matrix)
    h_gat = GATLayer(h_gnn, adj_matrix)
    node_embeddings_sequence.append(h_gat)
  # Stage 2: Timing modeling
  lstm_input = torch.stack(node_embeddings_sequence)
  lstm_output, (h_n, c_n) = LSTMLayer(lstm_input)
  final_representations = h_n[-1, :, :]
  # Risk prediction
  risk_predictions = Classifier(final_representations)
  predicted_classes = torch.argmax(risk_predictions, dim=1)
Return predicted_classes # Return the risk level prediction result
```

4 Experimental analysis of corporate financial risk warning scheme

4.1 Experimental environment and parameter settings

The operating environment parameters of this experiment are displayed in Table 2.

Table 2: Experimental operating environment parameters

Serial number	Type	Parameter
1	Operating system	Windows 10
2	CPU (Central Processing Unit)	Intel Core i7-10700F
3	Graphics card	NVIDIA GeForce RTX 3060 Ti
4	Programming language version	Python 3.7.9
5	Optimizer	Adam
6	Batch size	4
7	Epochs	30
8	Iteration	100
9	Dropout	0.3
10	Learning rate	0.0001

As Table 2 shows, this experiment was conducted in a well-configured environment. The operating system used was Windows 10, which ensures broad software compatibility. The programming language used was Python 3.7.9, which has rich library resources. The Adam optimizer was used when training the scheme, which can adaptively adjust the learning rate and accelerate convergence. Overall, such configuration and parameter selection laid a solid foundation for the experiment's success.

The GNN-GAT-LSTM model is built on PyTorch Geometric 2.3.0 and PyTorch 1.12.1 frameworks, featuring a modular hybrid architecture. It employs a two-layer GNN: the first aggregates direct neighbor information, while the second extends to second-order neighborhoods (64-dimensional output per layer). For heterogeneous edges (e.g., guarantees, equity), relationship-specific weight matrices enable differentiated messaging, preserving edge-type information in attention calculations. A dynamic graph is constructed annually (2018–2022) using Wind database financial reports. The GAT module uses 8 attention heads (8-dimensional each, spliced to 64), while the LSTM module processes GNN-GAT node embeddings over three consecutive years with a 128-dimensional hidden state, modeling temporal risk evolution. Training employs Adam (initial learning rate 0.0001, decaying 0.5 every 50 epochs), weight decay (1e-5), dropout (0.3), cross-entropy loss, and gradient clipping. Data is split chronologically (training: ≤2020; test: 2021–2022) for forward-looking evaluation, with hyperparameters tuned via grid search.

In view of the doubts about the risk of overfitting, a number of rigorous verification methods have been adopted in the research.The first is to test with samples outside the time series, and verify the non-participating training data from 2018-2020 and 2021-2022. The AUC value of the model reaches 0.954, proving the generalization ability.The second is to use the retention time window for cross-verification, and use different years as test sets in turn, and the model performance fluctuates within ±0.018.In the end, the model achieves stable convergence with the help of regularization techniques such as early stop method and dropout = 0.3.

4.2 Experimental analysis

4.2.1 Model performance comparison

The current economic environment is complex and changeable, and enterprises' financial risks are increasing daily. To evaluate the new EFR early warning scheme, drawing on the GNN-GAT-LSTM constructed in this study, it is compared with a variety of traditional and advanced models, including GNN, LSTM, BP-LSTM, GNN-LSTM, and HHGNN (Heterogeneous Hyper Graph Neural Networks). Through a comprehensive evaluation using multiple indicators such as precision, recall, specificity (SPEC), RSE, and RAE (Relative Absolute Error), the performance differences of each model in identifying different levels of financial crises are fully presented. The specific outcomes are displayed in Fig 2.

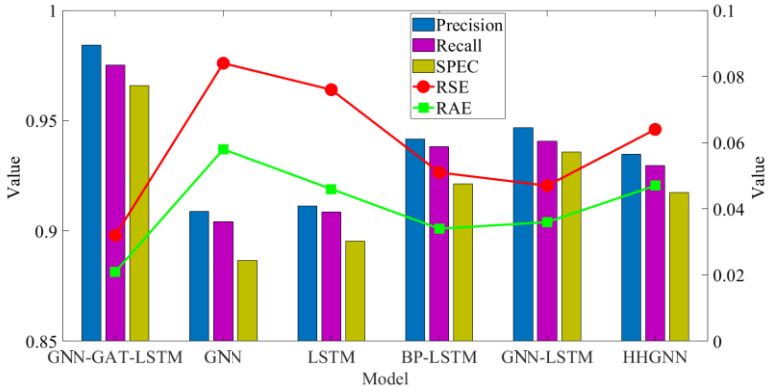


Figure 2: Performance comparison of different EFR warning schemes

As shown in Figure 2.The GNN-GAT-LSTM model demonstrates superior performance across all evaluation metrics compared to other models. It achieves precision (0.9842), recall (0.9752), and specificity (0.9658) values significantly outperforming competitors, highlighting its effectiveness in distinguishing healthy and crisis-prone enterprises. In error evaluation, its RSE (0.032) and RAE (0.021) values are substantially lower than those of GNN, LSTM, BP-LSTM, GNN-LSTM, and HHGNN, indicating higher prediction accuracy, stability, and generalization. While GNN excels in structural modeling, its precision and recall lag behind deep learning alternatives. LSTM struggles with inter-enterprise relationship modeling, limiting its performance. BP-LSTM and GNN-LSTM improve accuracy but fail to integrate graph dynamics effectively. HHGNN, despite its complex heterogeneous hypergraph structure, underperforms GNN-GAT-LSTM. The proposed model leverages GNN's graph-processing capabilities, enhances node interactions via GAT, and incorporates LSTM's temporal modeling, enabling comprehensive capture of enterprise risk propagation paths and significantly improving early warning performance.

In the field of EFR early warning, the GNN method is becoming an emerging and effective way. The GNN-GAT-LSTM model constructed in this paper is compared with other models. These models' loss value and accuracy comparison outcomes are displayed in Fig. 3.

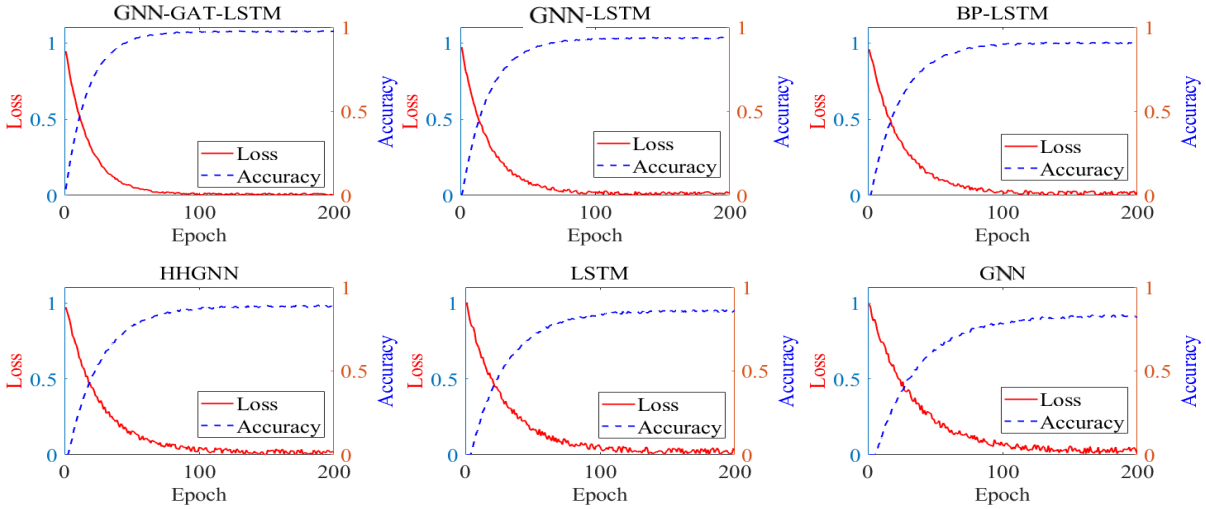


Figure 3: Comparison of loss values and accuracy curves of different models

As displayed in Figure 3, the experiment set 200 training rounds, and the GNN-GAT-LSTM model performed best. Its loss value dropped rapidly and stabilized at the lowest level. Its accuracy was the first to approach 98%, demonstrating the scheme's powerful feature extraction ability and learning stability. The traditional GNN model has a simple structure and cannot capture time dependencies. The loss decreases slowly, and the final accuracy is relatively low, ranking last. This performance gap verifies the advantages of the proposed method of integrating GNN, GAT, and LSTM. GNN-GAT-LSTM can use the graph structure to depict the complex relationships between enterprises, strengthen the influence of key nodes through GAT, and use LSTM to capture the dynamic changes of financial status over time. Although HHGNN has certain advantages in heterogeneous information modeling, it has high computational complexity, slow convergence speed, and slightly lower accuracy than GNN - GAT - LSTM.

To establish a more comprehensive benchmarking framework, this study introduces three advanced models specifically designed for dynamic graph structures: T-GCN (Temporal Graph Convolutional Network), DySAT (Dynamic Self-Attention Temporal Graph Network), and ST-GAT (Spatio-Temporal Graph Attention Network) as comparative benchmarks. While the GNN-LSTM baseline model only incorporates basic graph convolutional and LSTM modules, our proposed GNN-GAT-LSTM model demonstrates key innovations through its multi-head graph attention mechanism. This mechanism dynamically learns differentiated weighting factors for different neighboring nodes in assessing enterprise risks. Experimental results are presented in Table 3.

Table 3: Comparative results of dynamic graph models

Model	RSE	AUC	Precision
GNN-LSTM	0.047	0.913	0.938
T-GCN	0.041	0.925	0.947
DySAT	0.038	0.934	0.953
ST-GAT	0.035	0.943	0.962
GNN-GAT-LSTM	0.032	0.962	0.984

Table 3 demonstrates performance differences among dynamic graph models in corporate financial risk early warning tasks. The GNN-GAT-LSTM model outperforms others with 0.032 RSE, 0.962 AUC, and 0.984 accuracy, validating its effectiveness in modeling corporate risk propagation. The experiment reveals progressive performance trends: T-GCN achieved initial improvements through temporal graph convolution, DySAT enhanced with self-attention mechanisms further improved results, while ST-GAT converged to the model's level after integrating spatiotemporal attention. This indicates that refined modeling of heterogeneous relationships in corporate networks is crucial for improving risk prediction accuracy. Our model surpasses GNN-LSTM by leveraging customized heterogeneous graph structures and relationship-specific attention mechanisms to differentiate the influence of various relationships in risk propagation. Through deep encoding of business logic, it captures dynamic risk transmission while understanding economic semantics of relationship types, achieving both low error rates and precise identification of high-risk enterprises.

4.2.2 Comparison of early warning performance heat map

In the research on early warning of corporate financial risks, accurately identifying corporate financial conditions is significant in preventing potential crises. To evaluate the performance of different models on this task, 165 samples in Table 1 were used and divided into three groups. By comparing the prediction outcomes of the two models, the research method of this paper is compared with the traditional GNN model, as displayed in Figure 4.

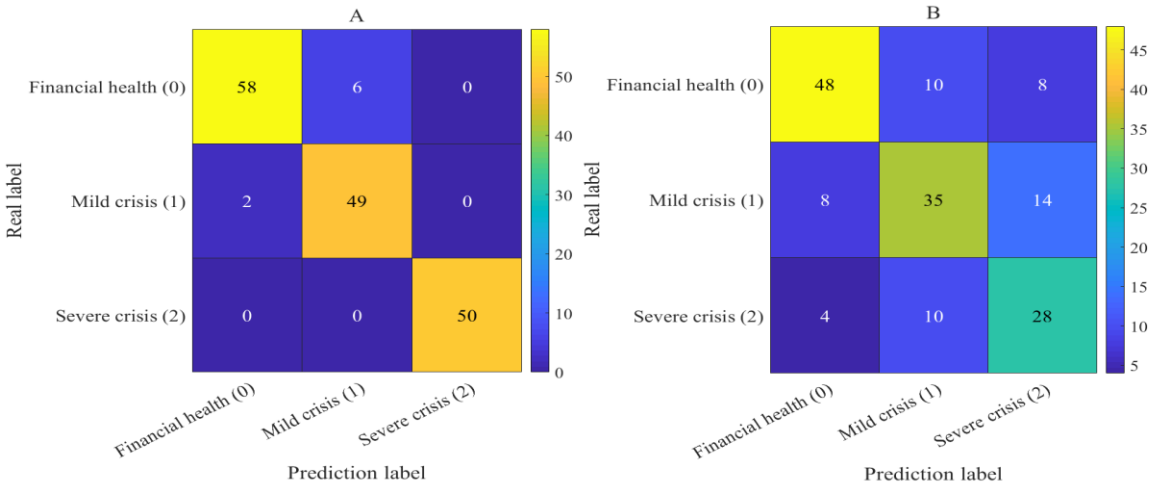


Fig. 4A. GNN-GAT-LSTM
Fig. 4B. GNN

Figure 4: Comparison of GNN-GAT-LSTM and GNN model warning performance heat map

As displayed in Fig. 4, the performance of the scheme GNN-GAT-LSTM studied in this paper is better than that of other models. As shown in Figure 4A, GNN-GAT-LSTM was studied, and the scheme performed well in financial health, with 58 samples correctly classified as financially healthy and only two samples misclassified as mild crisis. For the 55 samples of mild financial crisis, the scheme successfully identified 49, but 6 were misclassified as financially healthy. In predicting 50 samples of severe financial crisis, the scheme achieved all correct classifications; this shows that the GNN-GAT-LSTM model performs well in identifying severe financial crisis and accurately distinguishes financial health from mild financial crisis. As displayed in Figure 4B, the performance of the traditional GNN model is relatively poor. In the identification of financially healthy samples, only 48 samples were correctly classified, and 8 and 4 samples were misclassified as mild and severe financial crisis samples, respectively. For samples with mild financial crisis, only 35 were accurately identified. In the identification of severe financial crisis samples, only 28 samples were correctly classified. These data fully demonstrate that the traditional GNN model is far inferior to the GNN-GAT-LSTM model in classification accuracy, especially when dealing with mild and severe financial crisis samples, where the performance gap is more significant. The GNN-GAT-LSTM model studied in this paper performs better, mainly due to its unique architectural design. The scheme combines the structured data processing capabilities of GNN, GAT's adaptive weight allocation capabilities, and the time series analysis capabilities of LSTM. This combination enables the scheme to more accurately capture the complex relationships and dynamic changes in corporate financial data, thereby improving the accuracy of corporate financial status forecasts. In addition, the scheme can effectively distinguish different types of financial crisis samples, improve the overall classification performance, and provide strong support for corporate financial risk management in practical applications.

4.2.3 Comparison of warning error rates

Table 1 divides corporate financial risks into three levels, and warns of the three levels of risks to obtain warning

error rates. The error rate of the method studied in this paper is compared with other model methods, and for the scientific nature of the experiment, 10 experiments are conducted. The specific comparison outcomes are displayed in Figure 5.

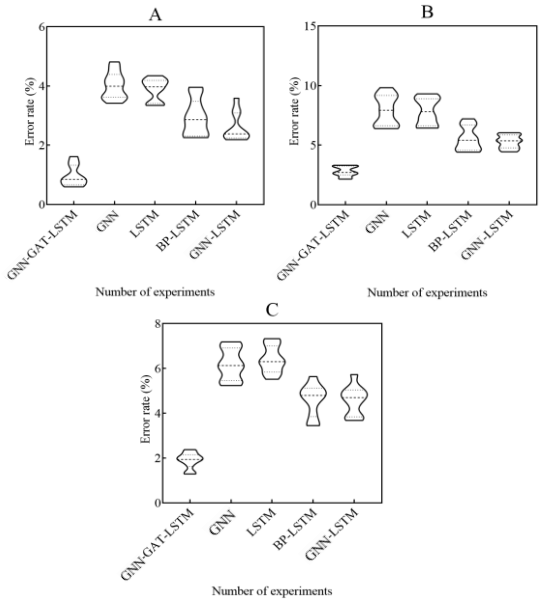


Fig. 5A. Financial health
Fig. 5B. Mild financial crisis
Fig. 5C. Severe financial crisis

Figure 5. Comparison of warning error rates of different models

As displayed in Figure 5, the performance of the GNN-GAT-LSTM model constructed in this paper at different risk levels is far better than that of several other methods. As shown in Figure 5A, in the early warning error rate analysis of financially healthy enterprises, 10 experiments were conducted. The average error rate of the GNN-GAT-LSTM model was 0.97%, which was 3.05%, 2.92%, 1.97%, and 1.64% lower than the error rates of the GNN, LSTM, BP-LSTM, and GNN-LSTM models, respectively. This gap demonstrates the scheme's ability to handle complex relationship networks and dynamic changes. As shown in Figure 5B, the GNN-GAT-LSTM model also performed well in terms of the early warning error rate of companies in mild financial crisis, with an average error rate of 2.8%, which was 5.16%, 5.07%, 2.77%, and 2.49% lower than the error rates of the GNN, LSTM, BP-LSTM, and GNN-LSTM models, respectively. This shows that the scheme can accurately identify potential risks and support corporate risk management even in more complex financial situations. As displayed in Figure 5C, in companies with severe economic crises, the average value of the GNN-GAT-LSTM model is 1.9%, which is lower than that of other models. This further proves the effectiveness of the scheme in high-risk early warning. It can capture local risk propagation patterns and accurately predict the risks that companies may encounter in the future by combining time series data. Based on the comparative analysis of the early warning error rates of companies with three different risk levels in 10 experiments, the GNN-GAT-LSTM model performs well at all levels. Therefore, whether the enterprise is in a relatively stable financial health state or facing different degrees of monetary crisis, GNN-GAT-LSTM can provide more scientific and accurate risk warning services to help enterprises respond quickly and ensure sustainable and healthy development.

To systematically evaluate the statistical significance of model performance, 10 independent experiments were conducted on all comparison models under identical data partitioning and random seed conditions, with average values and standard deviations of key metrics recorded. For validation methods, rigorous time series cross-validation was employed, with training and test sets being annually rolled over to ensure evaluation protocols fully align with real-world business requirements for predicting future risks using historical data. This approach guarantees the reliability and generalizability of performance improvement conclusions. Additionally, Wilcoxon sign-rank tests were applied to analyze differences in evaluation metrics between GNN-GAT-LSTM and benchmark models, with results presented in Table 4.

Table 4: Statistical significance analysis of model performance

Model	RSE	RAE	AUC
GNN	0.084 ± 0.005	0.062 ± 0.004	0.884 ± 0.008
GAT	0.068 ± 0.004	0.049 ± 0.003	0.901 ± 0.006
GNN-LSTM	0.047 ± 0.003	0.035 ± 0.002	0.913 ± 0.005
HHGNN	0.043 ± 0.003	0.031 ± 0.002	0.921 ± 0.004
GNN-GAT-LSTM	0.032 ± 0.002**	0.021 ± 0.001**	0.962 ± 0.003*

Note: **p <0.01, *p <0.05.

Table 4 demonstrates the superiority of the GNN-GAT-LSTM model in corporate financial risk early warning. This model significantly outperforms the baseline model across all key indicators, showing remarkable reductions in RSE (0.032) and RAE (0.021) compared to other models ($p<0.01$), indicating smaller prediction errors. The AUC value of our model reaches 0.962, markedly higher than other models ($p<0.05$), highlighting its exceptional capability to distinguish enterprises with different risk levels. The performance improvement stems from the model architecture's alignment with corporate risk transmission characteristics. Its stable and excellent performance under time series cross-validation proves that its advantages are not accidental but stem from its ability to capture the essential patterns of risk transmission, providing reliable technical support for financial risk control applications.

4.2.4 Analysis of different industries

In today's complex and changing economic environment, the importance of corporate financial risk warning has become increasingly prominent. To explore the most suitable EFR early warning scheme, this paper is not limited to the performance evaluation of a single model. The GNN-GAT-LSTM model is constructed and compared with the GNN, LSTM, BP-LSTM, GNN-LSTM, and HHGNN models. This is mainly to study the financial risk early warning of enterprises in the financial industry, and obtain the ROC (Receiver Operating Characteristic) curves of these models for early warning of financial industry enterprises, thereby obtaining the AUC value. The ROC curve obtained by the experiment is displayed in Figure 6.

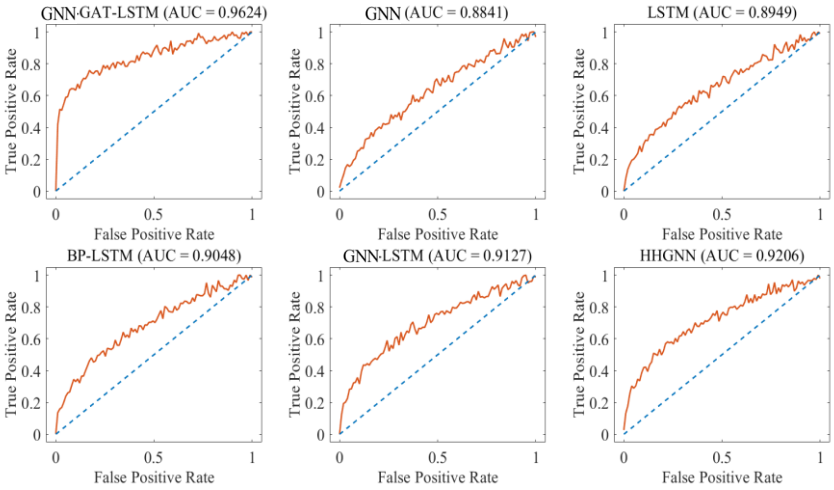


Figure 6: Comparison of ROC curves of different warning schemes

As shown in Figure 6, in the task of warning financial risks of financial industry enterprises, the GNN-GAT-LSTM model performs exceptionally well, with an AUC value of up to 0.9624, far exceeding other comparison models, GNN (AUC = 0.8841), LSTM (AUC = 0.8949), BP-LSTM (AUC = 0.9048), GNN-LSTM (AUC = 0.9127), and HHGNN (AUC = 0.9206). This result directly highlights the advantages of GNN-GAT-LSTM in classification ability. Also, it confirms from a statistical perspective that the scheme can maintain a high recognition accuracy rate under different false alarm rates. The overall trend of the ROC curve shows that in the face of complex and changeable risk transmission paths between enterprises; the scheme has stronger robustness and generalization ability. GNN-GAT-LSTM can achieve such excellent outcomes thanks to integrating multiple deep learning technologies. The GNN module enables the scheme to effectively model the corporate association structure and capture the potential risk transmission mechanism in complex networks such as supply chains and guarantee chains. The introduction of GAT enhances the scheme's focus on key nodes and meaningful relationships, making abnormal signals easier to identify and amplify, and improving early warning sensitivity. The LSTM module is responsible for processing time series data, helping the scheme understand the changing trend of

corporate financial conditions over time and enhancing the dynamic nature of predictions. The performance gap between GNN-GAT-LSTM and other methods can be more intuitively observed by visually analyzing the ROC curves of different models. Especially in the low false alarm rate area, the scheme has a higher true positive rate, which means that in practical applications, it can more effectively identify high-risk companies while controlling false alarms.

Different industries face unique risk patterns and challenges in today's complex and changing economic environment. Figure 7 shows the risk sensitivity factors of other sectors.

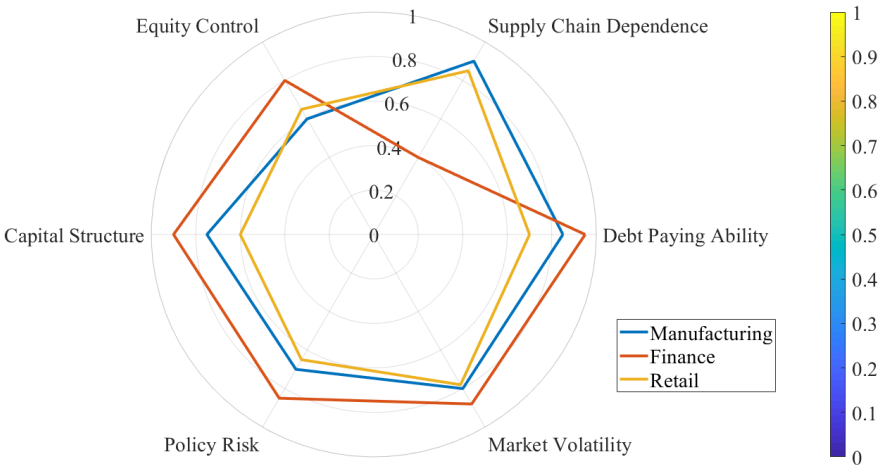


Figure 7: Risk sensitivity factors in different industries

Fig. 7 shows the sensitivity scores of different industries for six key risk factors, such as debt repayment capacity, supply chain dependence, and equity control. It uses a radar chart to intuitively show the differences in sensitivity of enterprises in different industries to various risks, highlighting the uniqueness of other enterprises in the financial risk transmission mechanism. The financial sector is susceptible to debt repayment ability, capital structure, and market volatility, which means that the economic status of financial enterprises is easily affected by liquidity, debt-to-asset ratio, and capital market volatility. This is consistent with the GNN-GAT-LSTM model's need to model complex relationships between enterprises. The manufacturing industry's high supply chain dependence and market volatility scores mean that its risk propagation may rely more on the diffusion of upstream and downstream enterprise associations, which is why this paper uses GNN to capture the structural relationship between enterprises. The retail industry is relatively less sensitive to most risk factors. Still, it performs well in terms of policy risks, reflecting that changes in the external environment significantly impact its business stability.

The research model demonstrates superior performance in the financial sector compared to its relatively weaker counterpart in retail, primarily due to distinct industry-specific risk transmission mechanisms. Financial sector risks rely on interbank guarantees and capital networks, featuring tight interconnections and clear transmission pathways that align with the GAT mechanism's characteristic of identifying critical nodes. The visual attention-weighted model precisely identifies systemically important institutions within guarantee circles. In contrast, retail sector risks are predominantly driven by individual operational factors like market demand and inventory turnover, with weak inter-firm connections and dispersed risk transmission paths, where graph-structured information provides limited incremental value. This validates the model's value limitations: when risks are strongly interconnected, structural analysis surpasses traditional financial metrics in explanatory power; when weak correlations prevail, industry-specific factors must be integrated for enhanced interpretation.

4.3 Model robustness analysis and ablation experiments

To systematically evaluate the generalization and robustness of the model across different domains and scales of graph data, this experiment selected three widely recognized benchmark datasets in graph learning. All datasets were obtained through standard channels and cover various network structure features. Specifically, the Cora citation network contains 2,708 paper nodes and 5,429 citation links, while the Citeseer citation network includes 3,312 paper nodes and 4,732 citation edges with 3,703 node feature dimensions. These datasets represent different network types and are used to set up consistent node classification tasks on heterogeneous data clusters, enabling comprehensive verification of the GNN-GAT-LSTM architecture's ability to capture graph structure information. The specific experimental results are shown in Table 5.

Table 5: Model performance comparison across domain-dimensional data sets (AUC)

Model / data set	Cora (Citation Network)	Citeseer (Citation Network)	Data set of this article
GNN	0.852	0.768	0.884
GAT	0.872	0.785	0.901
GNN-LSTM	0.865	0.776	0.913
HHGNN	0.883	0.801	0.921
GNN-GAT-LSTM	0.921	0.882	0.962

Table 5 demonstrates that the proposed model achieves optimal performance across three datasets, with AUC values reaching 0.921 and 0.882 on the Cora and Citeseer citation networks respectively—representing an 8-11% improvement over traditional GNN models, highlighting its effectiveness in capturing complex citation relationships in academic literature. The model maintains consistent excellence in both citation and financial networks, validating the architecture's versatility. Specifically, the GAT module identifies key neighboring nodes, the GNN layer aggregates local structural information, while the LSTM models dynamic node state evolution. This multi-level modeling mechanism enables the model to comprehend knowledge dissemination pathways in citation networks and analyze risk transmission chains in corporate guarantee networks, demonstrating strong generalization capabilities and practical value.

To investigate the contributions of three core components (GNN, GAT, LSTM) and critical hyperparameters, ablation experiments were conducted on financial datasets. The results are presented in Table 6:

Table 6: Analysis of model architecture ablation experiments

Model variants	GNN	GAT	LSTM	Precision	Recall	AUC
Variant A	☑	✗	✗	0.901	0.892	0.884
Variant B	☑	☑	✗	0.935	0.928	0.921
Variant C	☑	✗	☑	0.923	0.915	0.908
GNN-GAT-LSTM	☑	☑	☑	0.984	0.975	0.962

Table 6 presents the ablation experiment results demonstrating the synergistic effects of GNN, GAT, and LSTM in financial risk early warning tasks. The base variant Variant A containing only GNN can aggregate neighbor information to capture local risk propagation, but its AUC value of 0.884 indicates that static equal-weight aggregation struggles to distinguish impact differences among related enterprises. The attention mechanism-enhanced Variant B (GNN + GAT) improves AUC to 0.921, proving that dynamic weight allocation effectively identifies critical risk sources. The time-series modeling variant Variant C (GNN + LSTM) achieves an AUC of 0.908, highlighting the crucial role of enterprises' memory of historical risk states (e.g., prolonged profitability decline) in early warnings. The complete model employs a triple-coupling mechanism: GNN constructs risk propagation topologies, GAT focuses on key hubs, and LSTM characterizes temporal risk accumulation and mutations. This integrated approach captures both instantaneous shocks from guarantee chain ruptures and gradual supply chain risks, ultimately achieving an outstanding prediction performance of 0.962 while comprehensively covering enterprise risk propagation characteristics.

5 Discuss

Traditional financial early warning models are mostly based on the company's own time-series financial data. For example, although the LSTM model can capture the time dependence of a single company's financial indicators, the AUC value of 0.8949 shows that ignoring the structured information of the inter-enterprise correlation network will lose important risk transmission signals.Although the basic GNN model (AUC 0.8841) can capture local risk diffusion through neighbor information aggregation, the equal rights aggregation mechanism cannot distinguish the differences in influence of different affiliated companies, and treats major suppliers and minor customers equally, which is inconsistent with economic reality.The core breakthrough of this model lies in the synergy of three layers of components: the GNN layer builds the topological foundation of risk dissemination, and captures direct and indirect effects through two layers of messaging; the multi-head attention mechanism of the GAT layer can dynamically evaluate the importance of affiliated enterprises, which fits the law of real-world risk transmission. Experiments show that the introduction of Variant B of the GAT attention mechanism increases the AUC from 0.884 to 0.921, which verifies the necessity of differentiated weight distribution; the LSTM timing module solves the problem of dynamic modeling of

corporate financial status, can identify current risks, and can also capture progressive risk models such as “three consecutive quarters of cash flow attenuation”.Compared with the current advanced heterogeneous hypergraph neural network HHGNN (AUC 0.9206), this model has an AUC increase of 0.0418 under similar complexity.The advantage is that the GAT dynamic attention mechanism can more flexibly adapt to the time-varying characteristics of enterprise relationships; LSTM fine time series modeling makes up for the lack of time dimension characterization in the pure graph model.In the risk assessment of enterprises in the financial industry, this model has an AUC of 0.9624, and it has excellent modeling ability for the dynamic process of complex financial-related network risks.

From the application point of view, the innovative value of this model lies not only in quantitative indicators, but also in providing risk interpretability.By visualizing the attention weights of the GAT layer, risk managers can trace the path of risk transmission and identify system-important nodes, which is not possible with traditional “black box” models.The cross-disciplinary verification results of the model (the AUC value in Cora is 0.921, and the AUC value in Citeseer is 0.882) prove the versatility of its architecture. It can also identify high-impact documents and capture the evolution trajectory of academic hotspots in the citation network. The strong generalization ability across disciplines shows that it captures the universal law of information dissemination.However, the model has limitations, such as high requirements for data quality and integrity, and complex hyperparameter tuning.In the future, the construction of adaptive adjacency matrices and the introduction of dynamic graph neural networks can be explored.Overall, GNN-GAT-LSTM integrates graph structure learning, attention mechanism and timing modeling to provide both accurate and explanatory solutions for enterprise financial risk early warning, and lays the technical foundation of intelligent risk control system.

The corporate financial risk early warning model demonstrates strong performance (AUC 0.9624) but requires careful consideration of ethical and operational risks in practical use. For regulators, it enables dynamic systemic risk monitoring but necessitates a tolerance mechanism for false positives to avoid unnecessary scrutiny of healthy firms and subsequent financing cost increases. For banks, while the model optimizes credit approval and post-loan management, its 3.76% misjudgment risk could trigger credit crunches (false positives) or bad debt losses (false negatives). Additionally, its 91.3% recall rate for mild financial crises highlights challenges in identifying "gray area" companies, potentially delaying risk mitigation. To mitigate these risks, three measures are recommended: implementing manual expert reviews for high-risk cases, setting confidence thresholds for intervention, and regularly updating the model with current data to adapt to economic cycles. A human-computer collaboration framework with hierarchical responses is essential to control adverse impacts.

6 Conclusions

In the research on corporate financial risk early warning, traditional models are mostly limited to the company's financial data, and pay insufficient attention to the risk transmission effect in complex economic relationship networks such as supply chains and guarantee chains between companies, resulting in low accuracy of financial risk early warning schemes. This paper constructs a corporate financial risk early warning scheme utilizing GNN-GAT-LSTM. This model combines the advantages of GNN, GAT, and LSTM to build a financial risk early warning system that can comprehensively consider the complex relationships between enterprises and accurately capture the risk transmission effect. The experimental outcomes show that the GNN-GAT-LSTM model performs well in key evaluation indicators such as precision, recall, specificity, and RSE, and has significantly improved compared with traditional models and other advanced models. In the three categories of financial health, mild financial crisis and severe financial crisis, the average error rate of the scheme is as low as 0.97%, 2.8% and 1.9% respectively. This proves that the scheme can more accurately distinguish healthy enterprises from crisis enterprises, demonstrating its excellent performance in capturing complex economic relations between enterprises and their dynamic changes. The GNN-GAT-LSTM model uses the GNN layer to capture the local structural information of the enterprise association graph; the GAT layer uses the attention mechanism to dynamically allocate the weights of neighbor nodes, strengthening the role of key related enterprises in transmitting risks to the target enterprise. The LSTM layer effectively models the dynamic changes of corporate financial risks over time. This multi-level, all-around modeling approach enables the scheme to capture complex relationships and dynamic changes in corporate financial data more accurately, thereby significantly improving early warning performance.

Authorship contribution statement

Meiling ZHANG: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.

Author statement

All investigators have examined and authorized the manuscript, fulfilling the criteria for authorship. Each investigator certifies that it reflects honest work.

Ethical approval

All investigators have substantially contributed to the critical work involved in the article and are willing to assume public accountability for its content.

References

[1] G. C. Landi, F. Iandolo, A. Renzi, and A. Rey, "Embedding sustainability in risk management: The impact of environmental, social, and governance ratings on corporate financial risk," *Corp Soc Responsib Environ Manag*,29(4): 1096–1107, 2022. https://doi.org/10.1002/csr.2256

[2] I. M. Bufarwa, A. A. Elamer, C. G. Ntim, and A. AlHares, "Gender diversity, corporate governance and financial risk disclosure in the UK," *International Journal of Law and Management*, 62(6): 521–538, 2020. https://doi.org/10.1108/IJLMA-10-2018-0245

[3] M. Naved, R. Kumar, and S. S. Saheb, "Analyzing Financial Stability by Predicting Bankruptcy Situations with Machine Learning," Journal of Artificial Intelligence and System Modelling, vol. 02(02): 18–35, 2024, doi: 10.22034/jaism.2024.457068.1039.

[4] L. Tong and G. Tong, "A novel financial risk early warning strategy based on decision tree algorithm," Sci Program, 2022(1): 4648427, 2022. https://doi.org/10.1155/2022/4648427

[5] Gu, L.Gu, "Optimized backpropagation neural network for risk prediction in corporate financial management ,"*Scientific Reports,* 13(1), 19330,2023. https://doi.org/10.1038/s41598-023-46528-8

[6] Y. Sun, "Financial Transaction Network Risk Prediction Model Based On Graph Neural Network," *Procedia Computer Science*, 261, 763-771,2025.https://doi.org/10.1016/j.procs.2025.04.403

[7] C. Li, K. Jin, Z. Zhong, P. Zhou, and K. Tang, "Financial risk early warning model of listed companies under rough set theory using BPNN," *Journal of Global Information Management (JGIM)*, 30(7): 1–18, 2021. DOI: 10.4018/JGIM.300742

[8] X. Nie and G. Deng, "Enterprise financial early warning based on lasso regression screening variables," *Journal of Financial Risk Management*, 9(4): 454–461, 2020. doi: 10.4236/jfrm.2020.94024.

[9] J. Cheng, X. Lu, and X. Zhang, "Construction of Enterprise Financial Early Warning Model Based on Intelligent Mathematical Model," *Math Probl Eng*, 2022(1): 5230147, 2022. https://doi.org/10.1155/2022/5230147

[10] Y. Cao, Y. Shao, and H. Zhang, "Study on early warning of E-commerce enterprise financial risk based on deep learning algorithm," *Electronic Commerce Research*, 22(1): 21–36, 2022. https://doi.org/10.1007/s10660-020-09454-9

[11] H. Shang, D. Lu, and Q. Zhou, "Early warning of enterprise finance risk of big data mining in internet of things based on fuzzy association rules," *Neural Comput Appl*, 33(9): 3901–3909, 2021. https://doi.org/10.1007/s00521-020-05510-5

[12] H. Zhang and Y. Luo, "Enterprise financial risk early warning using BP neural network under internet of things and rough set theory," Journal of interconnection networks, 22(03): 2145019, 2022. https://doi.org/10.1142/S0219265921450195

[13] J.Chen, and B. Sun, "Enhancing financial risk prediction using TG-LSTM model: An innovative approach with applications to public health emergencies,"*Journal of the Knowledge Economy*, 16(1), 2979-2999,2025.https://doi.org/10.1007/s13132-024-02081-x

[14] C. Guo, D. Pi, J. Cao, X. Wang, and H. Liu, "Early warning model for industrial internet platform based on graph neural network and time convolution network," *J Ambient Intell Humaniz Comput*, 14(3): 2399–2412, 2023. https://doi.org/10.1007/s12652-022-04493-6

[15] K. Bi, C. Liu, and B. Guo, "Enterprise risk assessment model based on graph attention networks," *Applied Intelligence*, 55(3): 1–17, 2025. https://doi.org/10.1007/s10489-024-06103-8

[16] J. Wang, G. Liu, X. Xu, and X. Xing, "Credit risk prediction for small and medium enterprises utilizing adjacent enterprise data and a relational graph attention network," *Journal of Management Science and Engineering*, 9(2): 177–192, 2024. https://doi.org/10.1016/j.jmse.2023.11.005

[17] Y. Ren, X. Chen, H. Chen, and H. Zhu, "Does Inter-Industry Risk Spillover Network Predict Financial Crisis? Evidence from a Gated Graph Neural Networks Approach," Evidence from a Gated Graph Neural Networks Approach. DOI:10.2139/ssrn.5182005

[18] J.Wang, L. Zhou, C.Jiang and Z. Wang, "Modeling and Interpreting the Propagation Influence of Neighbor Information in Time-Variant Networks with Exemplification by Financial Risk Prediction,"*Journal of Management Information Systems*, 42(1), 105-142,2025.https://doi.org/10.1080/07421222.2025.2452016

[19] Q. Zhang, Y. Zhang, X. Yao, S. Li, C. Zhang, and P. Liu, "A dynamic attributes-driven graph attention network modeling on behavioral finance for stock prediction," *ACM Trans Knowl Discov Data*, 18(1): 1–29, 2023. https://doi.org/10.1145/3611311

[20] X. Chen and Z. Long, "E-commerce enterprises financial risk prediction based on FA-PSO-LSTM neural network deep learning model," Sustainability, 15(7): 5882, 2023. https://doi.org/10.3390/su15075882

[21] K.Xu, Y.Wu, M.Jiang, W.Sun and Z.Yang, "Hybrid LSTM-GARCH framework for financial market volatility risk prediction,"*Journal of Computer Science and Software Applications*, 4(5), 22-29,2024.https://doi.org/10.5281/zenodo.13643010

M. S. A.Dolon, "HYBRID MACHINE LEARNING–DRIVEN FINANCIAL FORECASTING MODELS: INTEGRATING LSTM, PROPHET, AND XGBOOST FOR ENHANCED STOCK PRICE AND RISK PREDICTION,"*Review of Applied Science and Technology*, 4(01), 01-34.2025https://doi.org/10.63125/nr1j8527

ImpGPT: Adaptive Patch-Based GPT-2 Transformer for Multivariate Time Series Imputation

Shuhui Ye[1], Guo Li[1,2*], Yirui Cheng[1], Chenxi Dong[1]
[1]School of Artificial Intelligence and Software Engineering, Nanyang Normal University, Nanyang, Henan, 473061, China
[2]Collaborative Innovation Center of Intelligent Explosion-proof Equipment，Henan Province，Nanyang, Henan, 473061, China
E-mail: airforce1980@126.com
*Corresponding author

Keywords: Multivariate time series, missing value imputation, adaptive patch, gpt-2, generative transformer

Received: August 27, 2025

Missing value imputation for multivariate time series data is a critical issue in time series analysis, with wide applications in industrial monitoring, sensor data recovery, and intelligent transportation systems. Traditional imputation methods often perform poorly under high missing rates, struggling to recover lost data trends. This paper proposes an adaptive Patch partitioning-based multivariate time series missing value imputation model—ImpGPT. This model combines an adaptive Patch partitioning mechanism and a Box encoding module. Through the adaptive Patch partitioning mechanism, time series data are dynamically divided into multiple Patches to capture local feature changes in the sequence. The size and position of Patch partitioning are dynamically generated and adjusted in real time by a lightweight network, which avoids the loss of temporal information that may be caused by traditional fixed partitioning methods.The Box encoding module encodes the geometric information and missing features of each Patch into structural vectors, explicitly associating missing patterns with temporal structures and enhancing the model's sensitivity to local structural changes.ImpGPT adopts a frozen and fine-tuned GPT-2 generative Transformer model to encode and model Patch sequences. This retains the strong sequence representation capability of the original GPT-2 and further optimizes the model to adapt to specific tasks. Combined with a masked normalization mechanism, it ensures accurate imputation of missing data.To verify the effectiveness of the model, we selected two datasets: one is a real flight sensor record from a certain aviation system, which includes parameters such as pitch angle, roll angle, heading angle and their corresponding angular velocities; the other is the public Electricity dataset.Experimental results show that under various missing rates, ImpGPT significantly outperforms existing benchmark methods in metrics including Mean Squared Error (MSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), and Normalized Root Mean Squared Error (NRMSE). Especially under the high missing rate of 75%:In the aviation sensor dataset, the MSE of ImpGPT is 0.0346, which is 1.7% lower than that of PatchTST (0.0352) and 23.1% lower than that of GPT4TS (0.045).In the Electricity dataset, the MSE of ImpGPT is 0.1253, which is 16.3% lower than that of PatchTST (0.1498) and 7.6% lower than that of GPT4TS (0.1356).These results indicate that ImpGPT still maintains good recovery accuracy even in extreme missing scenarios. Ablation experiments further verify that the adaptive Patch partitioning mechanism and model structure play key roles in improving imputation accuracy. ImpGPT performs excellently in handling high-missing-rate multivariate time series imputation tasks, with strong robustness and broad application potential.

Povzetek: Študija predstavi ImpGPT, ki z adaptivnim razrezom časovnih zaporedij modelira lokalno strukturo in zanesljivo imputira manjkajoče vrednosti v večspremenljivskih časovnih vrstah tudi pri visokih stopnjah manjka.

1 Introduction

In time series data, data missing is a common yet critical issue. Missing data is typically caused by sensor failures, transmission errors, environmental interference, and other factors [1]. During the analysis of multivariate time series, if the impact of missing data is not considered, it will not only weaken the ability to model dependencies between variables but also undermine the effectiveness of subsequent tasks [2]. Therefore, developing effective time series missing value imputation models is of great significance for improving the performance of downstream analysis.

Deep learning is developing rapidly. Recently, a growing number of studies have leveraged pre-trained language models for time series representation learning.For

example, GPT4TS [3], by freezing and fine-tuning pre-trained language models, has its own advantages in handling incomplete multivariate time series. However, these models divide time series into patches, which can then be treated as tokens in natural language processing to be input into GPT2 for learning. Although this method has achieved certain results, it only treats time series as text and aligns them with the model. While this method well fits the input characteristics of GPT2, it ignores the temporal information correlation characteristics of the sequence. As shown in the upper part of Figure 1 this method breaks the continuity of the sequence and the coherent contextual information. When GPT2 is learning, it cannot fully understand the semantic information of the sequence, which has a great impact on the time series missing value imputation model. To solve this problem, this paper proposes a new multivariate time series missing value imputation model—ImpGPT, which introduces a new patch division method to retain the integrity of local sequence information as much as possible. In figure 1, from the black circle to the red circle, the originally disconnected complete wave is retained under our division method. ImpGPT adopts an adaptive Patch division mechanism, enabling time series data to perform effective local feature extraction while maintaining temporal continuity. This method can dynamically adjust the data partitioning method according to the actual missing pattern of the data, thereby maximizing the retention of the structural information of the time series data. On this basis, ImpGPT combines the frozen pre-trained GPT-2 generative Transformer model, making full use of its powerful contextual modeling ability to capture the global dependencies in the data, thus significantly improving the imputation accuracy and the robustness of the model.

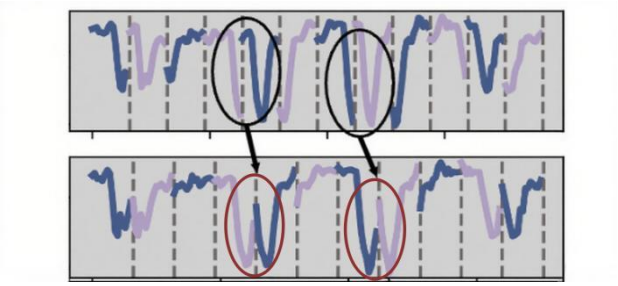


Figure 1: Comparison of different patch partitioning methods.

2 Research status

In recent years, machine learning-based approaches to handling missing data have garnered growing attention from researchers [4]. These imputation methods adopt a predictive paradigm, constructing forecasting models based on originally observed data to predict values at missing positions. Interpolation methods aim to fit a "smooth curve" to the observed results, thereby reconstructing missing values through local interpolation [5] [6].

In recent years, deep imputation models have shown a diversified evolutionary context, and their technical routes have formed a multi-branch development pattern, specifically including: diffusion-based probabilistic modeling, non-autoregressive multi-resolution frameworks, graph neural network (GNN) architectures for spatial dependency modeling, self-attention mechanisms integrated with generative adversarial networks,autoregressive reconstruction based on masked self-attention (SAITS), and bidirectional recurrent neural network schemes for real-time imputation scenarios [7-12]. The development of these technical branches not only adapts to the practical needs of the continuous complication of missing patterns in time series data but also reflects the differences in the understanding of missing mechanisms among different modeling paradigms.

Transformer-based time-series modeling techniques have continuously broken through performance bottlenecks in multiple dimensions and also demonstrated advantages in multivariate time-series missing value imputation tasks. At the infrastructure level, the ProbSparse mechanism proposed by Informer effectively reduces computational complexity while maintaining the ability to model long-range dependencies,However, it mainly focuses on global attention sparsity and has insufficient capability to capture short-term local missing information [13]. Reformer improves the efficiency of approximate attention computation through Locality-Sensitive Hashing (LSH),but its ability to perceive the structure of missing positions is limited, leading to a decline in recovery performance when there are a large number of missing values in local segments [14]. Autoformer strengthens the modeling capability for periodic structures, yet it tends to cause overfitting in imputation when processing non-periodic or mixed-structure data, which affects the accuracy of results [15]. FEDformer combines Fourier transform with attention mechanisms, and its frequency-domain modeling characteristic determines that it responds slowly to random missing values and mutation positions, limiting the flexibility of imputation [16]. ETSformer introduces a season-trend decomposition module, but it does not explicitly model dynamic structural dependencies between multiple variables [17]. DLinear as a linear modeling baseline, performs well when the data distribution is relatively stable. However, it lacks the ability to model nonlinear interactions, resulting in limited accuracy when facing dynamically changing complex sequence structures [18]. The static partitioning model PatchTST enhances the model's expressive ability for local structures and short-term patterns. Nevertheless, it does not explicitly model the continuity between patches and the interaction structure between variables, leading to performance bottlenecks in long-range dependency missing imputation scenarios [19].

In terms of pre-training technology, researchers are attempting to migrate the capabilities of large models that have achieved breakthroughs in the field of natural language processing to time series modeling tasks. Timer has constructed a general temporal Transformer pre-training framework to improve the efficiency of time series modeling [20].In addition to Timer, there are other

general temporal Transformer pre-training frameworks. TimeNet laid the foundation for early temporal pre-training by extracting transferable temporal representations through large-scale training of stacked RNN structures [21]. GPT4TS successfully migrated the GPT-2 model to temporal tasks, pioneering a new paradigm for the application of large models in temporal modeling [3]. However, existing models often ignore the structural nature of missing information itself during the sequence normalization and feature extraction stages, thereby weakening the model's ability to perceive missing mechanisms.

To address the limitations of existing methods, ImpGPT proposes an innovative Adaptive Patch Partitioning Mechanism and Box Encoding Technology. Unlike traditional sliding window methods and graph-based encoders, ImpGPT can dynamically adjust the size and position of patches, flexibly adapt to different missing patterns, effectively capture local changes, and avoid the information loss caused by static partitioning. Meanwhile, through encoding the geometric properties of patches, the Box Encoding Technology enables the model to better capture local structures and global dependencies in time series data, reducing reliance on complex external graph structures. In time series data processing tasks, GPT-2 has significant advantages over traditional Transformers in capturing long-term dependencies and complex patterns. Its decoder-based autoregressive mechanism makes it more suitable for handling long-term dependencies in time series data. Unlike the original GPT model, which is mainly designed for language tasks, GPT-2 exhibits stronger context modeling capability, especially for complex time series prediction and imputation tasks. To highlight the differences between various methods in terms of adaptability, robustness to missing structures, and context modeling capability, representative models are compared in Table 1.

Table 1: Model comparison and limitation analysis

Model	Key Technologies	Limitations
PatchTST	Fixed-length patch partitioning	Cannot flexibly handle dynamic missing patterns and lacks modeling of interactions between variables.
FEDformer	Fourier/wavelet frequency-domain modeling	Responds slowly to sudden missing values and struggles to handle irregular missing patterns.
SAITS	Dual-stream self-attention, masked reconstruction	Insufficient capture of long-range dependencies and difficulty adapting to high missing rates and irregular time intervals.
DLinear	Trend/residual decomposition	Only applicable to data with stable distributions and lacks the ability to model dynamic changes.
GNN	Graph convolution + temporal recursion	Relies on fixed graph structures and lacks flexible adaptation to missing patterns in time series data.
GPT4TS	Migration of GPT-2 to time series tasks	Lacks the ability to explicitly utilize missing structures and struggles to capture complex missing patterns.

3 Model design

The ImpGPT model designed in this study is intended to address the multivariate time series imputation problem under high missing rate conditions. Firstly, the Adaptive Patch partitioning and the pre-trained capability of GPT-2 are adopted to improve the imputation accuracy when facing missing data. Secondly, by freezing most weights of GPT-2 and only fine-tuning task-specific layers, the computational efficiency is optimized and the training time is reduced. Meanwhile, the model's robustness and generalization ability are enhanced to ensure that it can effectively perform imputation under different missing patterns and data types.

3.1 Overall framework of the model

ImpGPT processes inputs using a masked normalization mechanism, calculating statistics solely based on valid observations to maintain data stability. It dynamically generates Patches through adaptive Patch partitioning, adjusting sampling positions and sizes. Box encoding maps geometric attributes into structural vectors, which are linearly projected into the feature space via feature embedding. For global modeling, a frozen and fine-tuned GPT-2 is introduced: the attention layers and feed-forward network layers are frozen to retain pre-trained knowledge, while positional encoding and LayerNorm layers are fine-tuned to adapt to temporal structures and capture long-range dependencies across time steps. The output layer predicts missing values through a fully connected layer and restores them to the original physical scale via denormalization, ensuring the physical consistency of imputation results. The complete framework is shown in Figure 2.

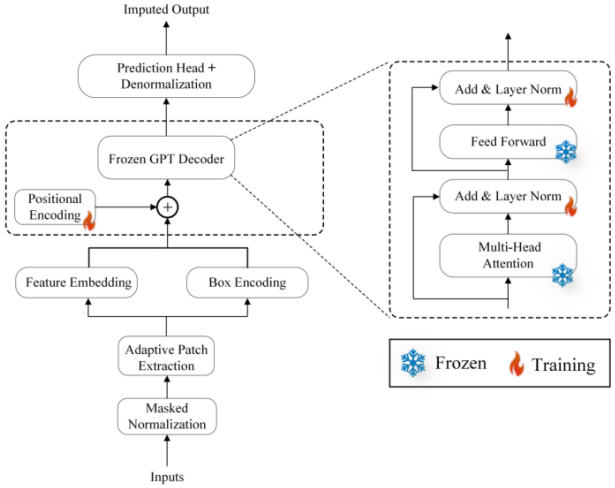


Figure 2: Framework diagram of the ImpGPT model.

3.2 Input normalization and mask processing

The evaluation of deep learning models for imputation tasks on time series datasets heavily relies on masking practices, where certain data points are intentionally designated as missing to simulate incomplete data conditions [22]. Therefore, ImpGPT introduces a masked normalization mechanism in the input stage to avoid missing values interfering with the statistical calculation of variables.

The original multivariate time series input is $X \in R^{B\times L\times M}$, where B is the batch size, L is the sequence length, and M is the variable dimension. The missing mask is $M \in \{0,1\}^{B\times L\times M}$. To ensure that the normalization process is based solely on valid observations, the model calculates the mean μ_m and standard deviation σ_m for each variable dimension as follows:

$$\mu_m = \frac{\sum_t X_{t,m} \cdot M_{t,m}}{\sum_t M_{t,m}} \quad (1)$$

$$\sigma_m = \sqrt{\frac{\sum_t \left(X_{t,m} - \mu_m\right)^2 \cdot M_{t,m}}{\sum_t M_{t,m}} + \varepsilon} \quad (2)$$

where ε is a small positive constant to improve numerical stability. The normalized input is:

$$\hat{X}_{t,m} = \frac{\left(X_{t,m} - \mu_m\right)}{\sigma_m} \cdot M_{t,m} \quad (3)$$

The masked normalization mechanism serves as a fundamental step in the encoding stage of ImpGPT, standardizing the numerical distribution of inputs to facilitate subsequent Patch extraction and context modeling.

3.3 Adaptive patch partitioning mechanism

To dynamically capture local structural changes in multivariate time series, ImpGPT designs a learnable Patch partitioning mechanism. This mechanism dynamically generates the center and width parameters of each Patch through length-scalable Patches, enabling segmented modeling of local regions in the sequence, as shown in Figure 3.

After normalization, the sequence $\hat{X} \in R^{B\times L\times M}$ is processed by a lightweight linear network f_{shift}. The objective of this network is to dynamically calculate the center offset $\Delta x \in R^{B\times N\times 1}$ and the Patch length extension $\Delta\omega \in R^{B\times N\times 2}$ for each Patch, where N denotes the number of Patches.The f_{shift}network computes these two parameters through linear layers and activation functions, thereby achieving fine-grained control over local regions. Its calculation formulas are as follows:

$$\Delta x = tanh\left(W_{offset} \cdot \hat{X}\right) \quad (4)$$

$$\Delta\omega = ReLU\left(tanh\left(W_{extend} \cdot \hat{X}\right)\right) \quad (5)$$

Among them, W_{offset} and W_{extend} are two key weight matrices in the network, which are used to learn the center offset and length extension of each Patch, respectively. The center offset Δx is transformed using the *tanh* activation function, which restricts its value within the range of [-1, 1] to avoid excessive offsets and ensure model stability. The Patch length extension $\Delta\omega$ is transformed using the *ReLU* activation function, ensuring its value is positive and has an adjustable range.

The generated center offset Δxand Patch length extension Δω are used to calculate the center position and boundary range of each Patch. For the i-th Patch, its center position c_i and boundary interval $[\, s_i, e_i,]$ can be expressed as:

$$c_i = x_i + \Delta x_i \quad (6)$$

$$s_i = c_i - \frac{\Delta\omega_i}{2} \quad (7)$$

$$e_i = c_i + \frac{\Delta\omega_i}{2} \quad (8)$$

The sampling region of each Patch interval $[s_i, e_i,]$ is uniformly divided into *D* time points, whose positions are:

$$pos_{i,k} = s_i + \frac{k}{D-1} \cdot (e_i - s_i), \quad k = 0,1,\ldots,D-1 \quad (9)$$

Since these positions are continuous real-number coordinates, they cannot directly index sequence values. To overcome the limitations of traditional interpolation methods such as nearest-neighbor interpolation and spline interpolation under high missing rate conditions, this study adopts grid sampling technology for interpolation. Traditional methods typically rely on fixed neighborhoods to impute missing values, which may fail to accurately recover data trends, especially in complex time series. In contrast, grid sampling can more flexibly capture changes in the data by dynamically adjusting sampling positions and sizes, effectively avoiding the loss of temporal fidelity [1] [2]. The model employs the *grid_sample* function to perform bilinear interpolation on $\hat{X}$, extracting features corresponding to the Patch from the input. By generating non-integer coordinates for interpolation calculations on the input sequence, local features are obtained. In the case of *1D* time series, the input sequence is rearranged into a *2D* format, and then interpolation is performed via the *grid_sample* function to ensure smooth feature extraction. Specifically, the grid_sample function uses target sampling positions to extract local features from the input sequence. The sampling position of each Patch is dynamically calculated and can be automatically adjusted according to its local structure. The interpolated value $\hat{X}_{i,k}$ can be expressed as:

$$\hat{X}_{i,k} = \sum_{j\in\mathcal{N}(pos\ i,k)} G\left(pos_{i,k}, j\right) \cdot \hat{X}_j \quad (10)$$

Where $\mathcal{N}\left(pos_{i,k}\right)$ denotes the two integer indices adjacent to $\mathcal{N}\left(pos_{i,k}\right)$, To avoid artifacts caused by temporal aliasing, the model uses a smooth interpolation weight function $G(\cdot)$, which is defined as follows:

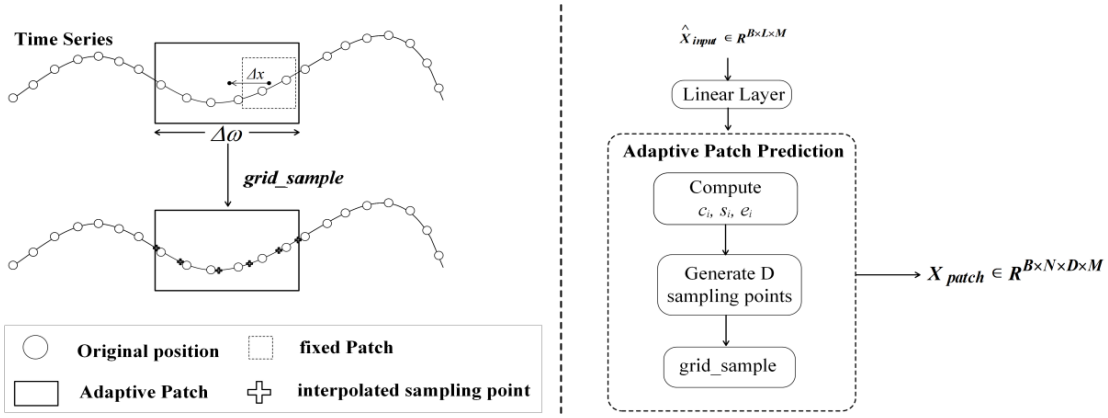


Figure 3: Adaptive patch generation.

.

$$G(i,j) = max(0,1-|i-j|) \quad (11)$$

This function ensures linear fusion between non-integer positions and their two nearest observation points, enabling a smooth and differentiable feature extraction process. It avoids the loss of low-frequency information and high-frequency noise, thereby reducing artifacts caused by temporal aliasing. By applying this operation to all *D* positions within each Patch, Patch features are formed:

$$X_{patch} = grid_sample(\hat{X}, pos) \quad (12)$$

Where $pos \in R^{B\times N\times D}$ represents the Patch sampling coordinates. Before interpolation sampling, the input sequence is expanded into a *2D* format, and linear interpolation is performed along the time axis via *grid_sample* this ensures smooth extraction of features within each Patch.

The overall workflow of this module is shown on the right side of Figure 3: the input sequence first passes through a linear network to predict sampling position parameters, which are further mapped to sampling coordinates on the continuous time axis. The model then performs interpolation sampling on the normalized input sequence based on these coordinates to construct Patch features with local integrity, ultimately obtaining an adaptive Patch tensor of dimension $X_{patch} \in R^{B\times N\times D\times M}$.

3.4 Structure-aware encoding module

ImpGPT introduces a structure-aware encoding module (Box Encoding), which encodes the geometric information and missing features of each Patch into a structural vector.

Specifically, for each Patch, its structural features include the following elements: Interval position (s_i, e_i), length $l_i = e_i - s_i$, missing ratio r_i, and mask statistics $mask_i$. This structural information is concatenated into a vector and input into a lightweight Multi-Layer Perceptron (MLP) to obtain the structural encoding vector of the corresponding Patch:

$$b_i = MLP([s_i, e_i, l_i, r_i, mask_i]) \in R^{d_{model}} \quad (13)$$

This structural vector is input into subsequent modeling modules along with the corresponding Patch content. This design not only provides positional information but also enhances the model's perception of local variation ranges and data integrity, facilitating improved cross-Patch context modeling [21].

3.5 GPT-2 encoder

After Patch feature extraction and structure encoding, ImpGPT employs the pre-trained language model GPT-2 to perform global modeling on the Patch sequence, aiming to capture long-range dependencies and cross-variable contextual information. Compared with traditional time series models such as RNN and TCN, GPT-2, built on a stacked Transformer decoder structure, boasts non-recursive modeling capabilities and stronger expressive power, particularly demonstrating significant advantages in scenarios with high missing rates and non-stationary fluctuations [23][24].

3.5.1 Patch embedding and fusion modeling

Taking the i-th Patch as an example, its content can be represented as a tensor $P_i \in R^{D\times M}$, where *D* denotes the number of sampling points and *M* is the variable dimension. ImpGPT first flattens each Patch and maps it linearly to the dimension space required by GPT-2 ($d_{model} = 768$, consistent with the pre-trained dimension of GPT-2):

$$e_i = Linear(Flatten(Patch_i)) \in R^{d_{model}} \quad (14)$$

Subsequently, all Patch embedding vectors form a sequence $E = [e_1; e_2; \ldots; e_N] \in R^{N\times d_{model}}$, which are added to their corresponding structural vectors *bi* to obtain the fused Patch representation:

$$\tilde{e}_i = e_i + b_i \quad (15)$$

The fused Patch sequence is fed into the GPT-2 backbone model for global context modeling, and the output result is:

$$H = GPT2(\tilde{E}) \in R^{N\times d_{model}} \quad (16)$$

Here, H_i represents the representation of the i-th Patch after integrating global semantics, serving as the input for the subsequent reconstruction module.

3.5.2 Freezing and fine-tuning strategy

In ImpGPT, we fine-tune the positional encoding and LayerNorm layers, while freezing the self-attention layers and feed-forward networks (FFN). This strategy

aims to balance computational efficiency and imputation accuracy, especially in data scenarios with high missing rates. To handle time series data with high missing rates, ImpGPT primarily enhances its imputation capability through Adaptive Patch Partitioning and Box Encoding Technology. Traditional imputation methods tend to suffer from overfitting or information loss under high missing rates. To overcome this limitation, ImpGPT adopts a dynamically adjustable patch partitioning strategy, which can automatically adjust the size and position of patches according to different missing patterns. This enables more flexible capture of local changes in the data and mitigates the challenges posed by high missing rates.

Since the self-attention layers and FFN have already learned the capabilities of long-range dependency modeling and context information capture during the pre-training phase of the model, we choose to freeze these two components to retain their pre-trained knowledge and avoid wasting computational resources during the fine-tuning phase. During the pre-training of GPT-2, positional encoding is mainly used to capture sequentiality in language; however, the temporal structure of time series data differs from the sequence in natural language. Fine-tuning the positional encoding layer helps the model better grasp the sequential nature of time series data, particularly when dealing with dynamic missing patterns. On the other hand, fine-tuning the LayerNorm layers helps the model adapt to the data distribution under conditions of missing data and high missing rates, thereby improving the model's stability and imputation accuracy.

In environments with high missing rates, the fine-tuned positional encoding layer and LayerNorm layers can enhance the model's ability to handle missing data, avoid overfitting under high missing rates, and enable the model to better recover the lost temporal information.Although the original sinusoidal positional encoding of GPT-2 is designed for natural language, its core feature modeling relative positional relationships through periodic differences of trigonometric functions highly aligns with the requirement of time series for modeling "relative time intervals" [25]. This design is also applicable to time series data. Despite the fact that time steps in time series data have fixed sampling intervals, while the word order in natural language has no physical intervals, the fine-tuned positional encoding has acquired time-aware capability. Through fine-tuning, the relative positional representations of the positional encoding are recalibrated to the physical properties of time intervals; thus, there is no need for additional explicit temporal encoding to transmit temporal structural information. This fine-tuning strategy is regarded as a standard practice, which helps improve the performance of downstream tasks and minimize the workload of fine-tuning [26][27].

3.6 Reconstruction and imputation output

Imputation, the process of identifying missing values and replacing them with substitute values, is also referred to as missing value imputation [28]. After completing contextual semantic modeling, ImpGPT maps high-dimensional representations back to the complete time series space through a structured decoding module, enabling the reconstruction of missing positions and the restoration of the original sequence.

First, the contextual representation $H \in R^{B\times N\times d_{model}}$ output by GPT-2 is flattened into a vector, and decoding mapping is performed via a fully connected network:

$$\hat{X} = Linear\big(Flatten(H)\big) \in R^{B\times(L\times M)} \tag{17}$$

To ensure the model performs consistent modeling of the entire sequence structure, ImpGPT adopts an end-to-end supervision mechanism. Instead of only interpolating at missing points, The model is trained to predict the entire sequence, thereby enhancing the continuity and contextual coherence of the reconstructed data.To enhance the numerical consistency between structural information and the original sequence, ImpGPT introduces a mask-aware denormalization strategy in the prediction stage. Combining the mean μ and standard deviation σ, the denormalization is computed as follows:

$$\hat{X}_{final} = \hat{X}_{rec} \cdot \sigma + \mu \tag{18}$$

This transformation ensures calculations with high observation accuracy, avoids error accumulation, and guarantees the physical rationality and dimensional consistency of predicted values. Finally, ImpGPT outputs $\hat{X}_{final}$ as the imputation result, ensuring the model has efficient recovery capabilities for missing value imputation under multiple missing patterns.

3.7 Loss function design

We adopt a mean squared error (MSE) loss function design based on a mask-guided mechanism. Specifically, in each training iteration, the model randomly generates a missing mask, replaces some positions in the original input with zero values, and feeds the mask information into the model together. After outputting the complete predicted sequence, only the mean squared error (MSE) between the predicted values and the true values at the missing positions is calculated to form the final optimization objective [11][12]. This design effectively avoids the interference of observed values on the gradient direction, allowing the model to focus on the reconstruction task of missing Patches. Specifically, by training on the predicted values of the entire sequence, the model can acquire more contextual information, enabling it to consider broader temporal dependencies and global trends when imputing missing values. This design approach ensures that imputation is not limited to local data points; instead, it improves imputation accuracy and consistency through global modeling. Thus, although the output is the predicted values of the entire sequence, the model's performance in the imputation task is enhanced especially under complex missing patterns, where it can provide more accurate imputation results. The loss function is defined as follows:

$$\mathscr{L}_{MSE} = \frac{1}{|\Omega|}\sum_{(i,j,k)\in\Omega}\left(\hat{x}_{i,j,k} - x_{i,j,k}\right)^2 \tag{19}$$

$\Omega = \{(i,j,k) \mid mask_{i,j,k} = 0\}$ represents the positions of all missing points in the current training batch. During the model input phase, a binary mask is used to mark missing positions, and missing values are processed via a padding method this ensures the model cannot access the true information of missing values. All evaluation metrics adopt the masking mechanism: only the error at missing positions is calculated, while predictions at non-missing positions are completely excluded. This design avoids data leakage and improves imputation accuracy.

4 Experiments

4.1 Experimental dataset

The experimental data utilized in this study is sourced from a real flight sensor record of a certain aviation system,forming a multivariate time series dataset. This dataset collects key flight status and attitude parameters continuously during flight at a frequency of 1 Hz, including pitch angle (in degrees), roll angle (in degrees), heading angle (in degrees), and their corresponding angular velocities (pitch rate, roll rate, yaw rate, all in degrees per second). In this research, "heading angle (degrees)" is regarded as the core imputation target. To model its periodic characteristics more effectively, this variable is encoded into sine and cosine dual components, which are then used as the final target output. The remaining variables serve as input covariates to participate in the modeling process collectively.

To verify the cross-domain generalization ability of the model, we conducted experiments on the real-world Electricity dataset. The Electricity dataset is a publicly available dataset widely used in time series missing value imputation tasks. Characterized by a 15-minute time granularity, it includes relevant statistical values or feature values such as 321, 96, and tuples (18317, 2633, 5261). The target variable of this dataset is electricity consumption load, which covers a variety of time-related features.The Electricity dataset provides a verification platform for the model's generalization ability. Through its diverse missing patterns and multi-dimensional temporal features, it enables effective evaluation of the model's performance in complex scenarios.

4.2 Experimental design

We followed the experimental setup of TimeNet. To simulate the missing data problem caused by sensor packet loss or communication interruption in real-world scenarios, we constructed six different missing rate settings, which are 12.5%, 25%, 37.5%, 50%, 62.5%, and 75% respectively. We randomly masked the corresponding proportion of data in the sequence and generated missing masks for training and evaluation.

ImpGPT is based on the pre-trained GPT-2, fine-tuning the positional encoding and LayerNorm. Its core innovation is the Adaptive Patch Partitioning Mechanism by setting the Patch size to 8 and step size to 4, overlapping sampling is achieved, and offset prediction is introduced to focus on information-dense regions.To ensure the fairness and comparability of the experiments, we adopted the experimental configuration of TimeNet in all baseline tests and standardized the evaluation process. We verified the effectiveness of this method by comparing it with a series of models, including GPT4TS (a study that fine-tunes language models for time series analysis), Transformer-based models (Autoformer, Informer, Reformer, FEDformer, ETSformer, and PatchTST), and the linear model DLinear.

4.3 Evaluation metrics

To comprehensively evaluate the performance of the ImpGPT model in multivariate time series missing value imputation tasks, this study adopts four commonly used evaluation metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), and Normalized Root Mean Squared Error (NRMSE).

MSE measures the squared error between the predicted values and the true values when the model imputes missing data.

$$MSE = \frac{1}{|\Omega|} \sum_{(i,j,k)\in\Omega} \left(\hat{x}_{ijk} - x_{ijk}\right)^2 \qquad (20)$$

MAE enables an intuitive assessment of the imputation performance across all missing data points.

$$MAE = \frac{1}{|\Omega|} \sum_{(i,j,k)\in\Omega} \left|\hat{x}_{ijk} - x_{ijk}\right| \qquad (21)$$

SMAPE eliminates the impact of magnitude differences on error measurement and serves as a key indicator for evaluating the accuracy of trend recovery and relative imputation accuracy in imputation tasks. To address the instability of SMAPE when values are close to zero, we adopt a smoothing technique in SMAPE calculation: by adding a very small constant ε (1e−8) to the denominator, we avoid computational anomalies caused by zero or near-zero denominators, thereby ensuring the stability of SMAPE under extreme cases.

$$SMAPE = \frac{100\%}{|\Omega|} \sum_{(i,j,k)\in\Omega} \frac{\left|\hat{x}_{ijk} - x_{ijk}\right|}{\frac{\left|\hat{x}_{ijk}\right| + \left|x_{ijk}\right|}{2} + \varepsilon} \qquad (22)$$

NRMSE normalizes errors.Considering the division-by-zero issue that arises when the maximum and minimum values of the data are equal, a very small constantε (1e−8) is also added to the denominator during calculation to ensure the stability of the computation.

$$NRMSE = \frac{\sqrt{\frac{1}{\Omega}\sum_{(i,j,k)\in\Omega}\left(\hat{x}_{ijk} - x_{ijk}\right)^2}}{max(x) - min(x) + \varepsilon} \qquad (23)$$

All evaluation metrics employ a masking mechanism, only the errors at missing positions are calculated, which ensures that the model evaluation focuses on the accuracy of the imputation task.

4.4 Experimental results and analysis

4.4.1 Overall imputation performance comparison

This study evaluates the performance of the ImpGPT model in the missing value imputation task on the aviation sensor dataset and Electricity dataset through experiments. The experimental setup covers evaluations under different missing rates (12.5%–75%), and the adopted evaluation metrics include Mean Squared Error (MSE), Mean Absolute Error (MAE), Normalized Root Mean Square Error (NRMSE), and Symmetric Mean Absolute Percentage Error (SMAPE).

Table 2 presents the average evaluation results of all models under the six different missing rates.

On the aviation sensor dataset, ImpGPT outperforms PatchTST and GPT4TS in terms of MSE, MAE, SMAPE, and NRMSE across all missing rates. Particularly under a 75% missing rate, ImpGPT achieves an MSE of 0.0921, an MAE of 0.2073, an SMAPE of 38.76%, and an NRMSE of 0.0339 performing significantly better than PatchTST (MSE: 0.0948, SMAPE: 40.64%) and GPT4TS (MSE: 0.1021, SMAPE: 42.86%). ImpGPT can effectively tackle sensor data recovery tasks under high missing rate scenarios, and it is superior to other models especially in terms of recovery accuracy and stability.

On the Electricity dataset, ImpGPT also demonstrates excellent imputation capability. Under the condition of a 75% missing rate, ImpGPT achieves an MSE of 0.0921, an MAE of 0.2073, an SMAPE of 38.76%, and an NRMSE of 0.0339, continuing to outperform other models. It shows strong recovery ability particularly in sensitive metrics such as SMAPE and NRMSE.

Compared with other baseline models, ImpGPT maintains the lowest error values across all missing rates, especially in high missing rate scenarios. Although Autoformer and Informer perform well at low missing rates, their errors (especially in SMAPE and NRMSE) rise sharply under high missing rate conditions, leading to a significant decline in recovery performance. In contrast, through its Adaptive Patch Partitioning Mechanism and Box Encoding Module, ImpGPT can maintain stable imputation accuracy under complex missing patterns.

ImpGPT exhibits significant advantages in handling multivariate time series missing value imputation tasks, particularly under high missing rate conditions. It not only effectively adapts to the missing patterns of different datasets but also outperforms existing mainstream models in terms of imputation accuracy and stability proving its broad adaptability and reliability in practical applications.

4.4.2 Comparative experiments under different missing rates

To further verify the robustness of the model under different missing rates, we analyzed the changing trends of MSE and MAE of each model under six missing rate settings. Figure 4 and Figure 5 respectively show the performance of ImpGPT and other models under different missing rates on the aviation sensor dataset and the Electricity dataset.

From the experimental results on the aviation sensor dataset and the Electricity dataset, the differences in robustness of various models under different missing rates profoundly reflect the degree of adaptation between their design logic and missing characteristics, further highlighting the advantages of ImpGPT. Particularly in high missing rate scenarios, ImpGPT exhibits the gentlest error growth and demonstrates more stable performance compared with GPT4TS and other baseline models.

In the aviation sensor dataset, ImpGPT exhibits the gentlest error growth: its MSE increases from 0.0175 to 0.0346, and its MAE rises from 0.0642 to 0.0961. This is attributed to ImpGPT's Adaptive Patch Mechanism, which can dynamically adjust the temporal capture range, and the Box Encoding, which accurately models the geometric positions and proportions of missing data thus enabling effective data recovery. In contrast, GPT4TS shows a much steeper error increase: its MSE goes up from 0.017 to 0.045, and its MAE climbs from

Table 2: Model evaluation

Methods	aviation sensor dataset（MSE/MAE/NRMSE/SMAPE）	Electricity dataset（MSE/MAE/NRMSE/SMAPE）
Autoformer	0.0513/0.1644/0.0621/65.5181%	0.1170/0.2425/0.0545/61.3926%
Informer	0.0808/0.1656/0.0703/64.7003%	0.2310/0.3414/0.0617/60.6241%
Reformer	0.0428/0.1191/0.0506/52.5766%	0.2110/0.3217/0.0475/49.2738%
DLinear	0.0398/0.1259/0.0493/54.5472%	0.1571/0.2862/0.0433/51.1285%
FEDformer	0.0627/0.1609/0.0605/65.7942%	0.1528/0.2778/0.0531/61.6657%
ETSformer	0.0670/0.1680/0.0636/71.2949%	0.2316/0.3532/0.0558/66.8154%
PatchTST	0.0245/0.0820/0.0390/43.3606%	0.0948/0.2094/0.0342/40.6423%
GPT4TS	0.0296/0.0935/0.0426/45.7348%	0.1021/0.2200/0.0347/42.8625%
ImpGPT	0.0235/0.0767/0.0386/41.3836%	0.0921/0.2073/0.0339/38.7612%

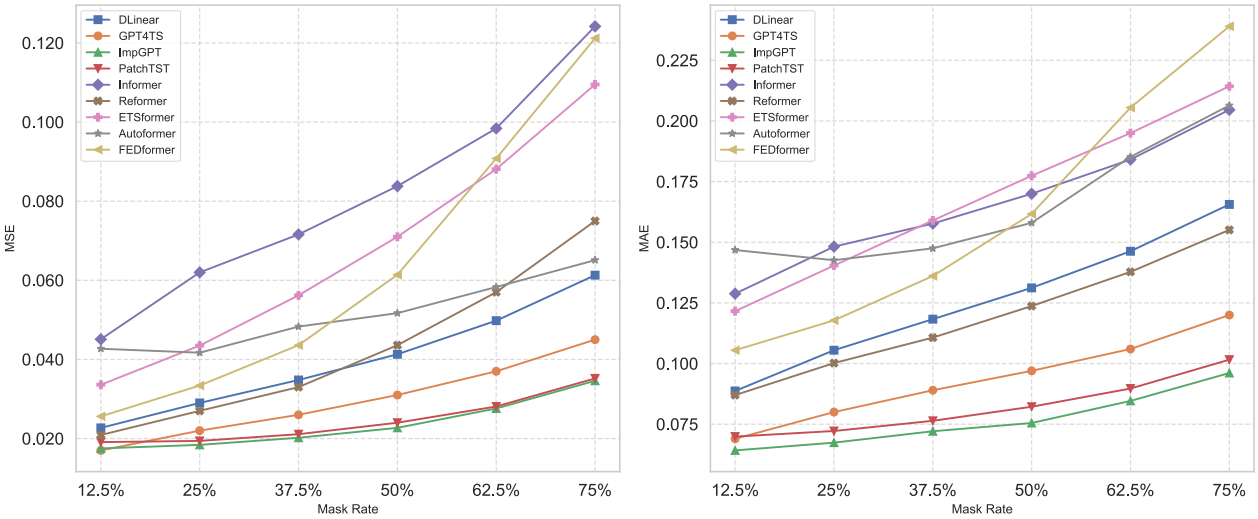


Figure 4: Evaluation results under different missing rates on the aviation sensor dataset.

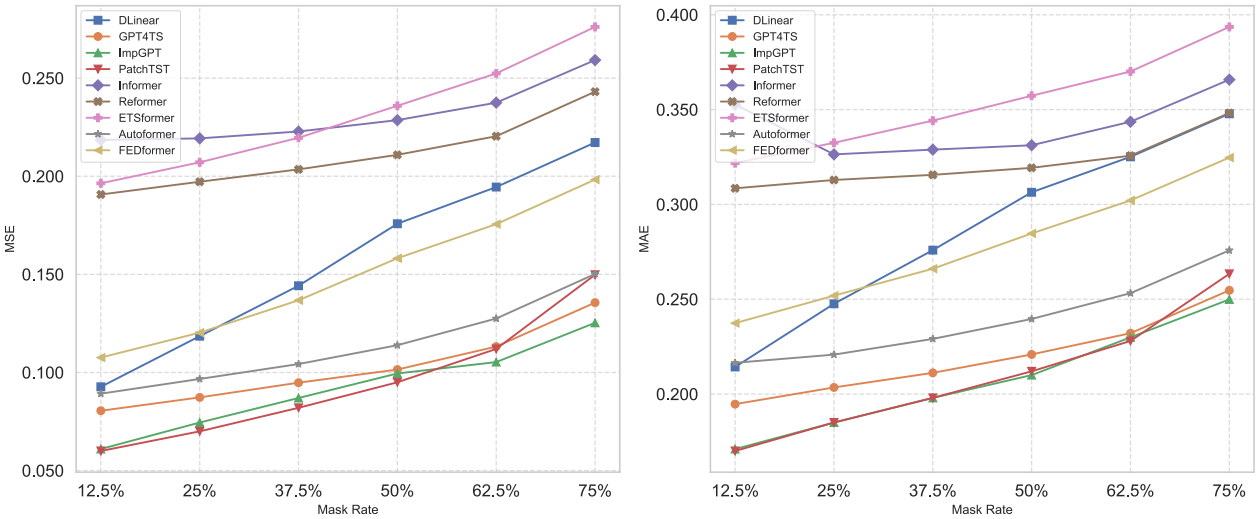


Figure 5: Evaluation results under different missing rates on the electricity dataset.

0.069 to 0.120. The reason for this is that GPT4TS does not explicitly utilize missing structure information; it only treats time series data as text tokens, making it difficult to handle non-stationary missing patterns. PatchTST performs close to ImpGPT when the missing rate is low, but its error surges sharply once the missing rate exceeds 50%. Reformer and FEDformer perform even worse under high missing rates, particularly showing limitations in capturing the long-term dependencies and short-term fluctuations of flight data.

In the Electricity dataset, ImpGPT still achieves the optimal performance: its MSE increases from 0.061 to 0.1253, and its MAE rises from 0.171 to 0.2499. Its Adaptive Patch Mechanism effectively adapts to the "stable-peak" structural changes, while the Box Encoding assists in trend recovery. In contrast, GPT4TS performs poorly in the Electricity dataset with its MSE increasing from 0.0805 to 0.1356 and MAE from 0.1947 to 0.2547. PatchTST's error also rises sharply under high missing rates, as its fixed Patches fail to match the dynamic structure of electricity load data. Models such as Reformer and Informer also see a significant increase in error under high missing rates, performing poorly in terms of MSE and MAE, and are unable to effectively handle the complex missing patterns of electricity load data.

Relying on its Adaptive Patch Mechanism and Box Encoding, ImpGPT maintains low errors and a gentle error growth trend across all missing rate ranges, demonstrating strong robustness in high missing rate scenarios. In contrast, models like GPT4TS, PatchTST, and Reformer show a significant increase in error under high missing rates and struggle to adapt to non-stationary missing patterns validating the practicality of ImpGPT as an industrial-grade time series missing value imputation solution.

4.4.3 Loss convergence analysis

To verify the generalization ability of ImpGPT on domain-specific small datasets such as the aviation sensor dataset, and to avoid the overfitting problem that large pre-trained models may encounter in tasks with limited data volume and specific scenarios, we analyzed

the changing trends of training loss, test loss, and validation loss. The results are shown in Figure 6.

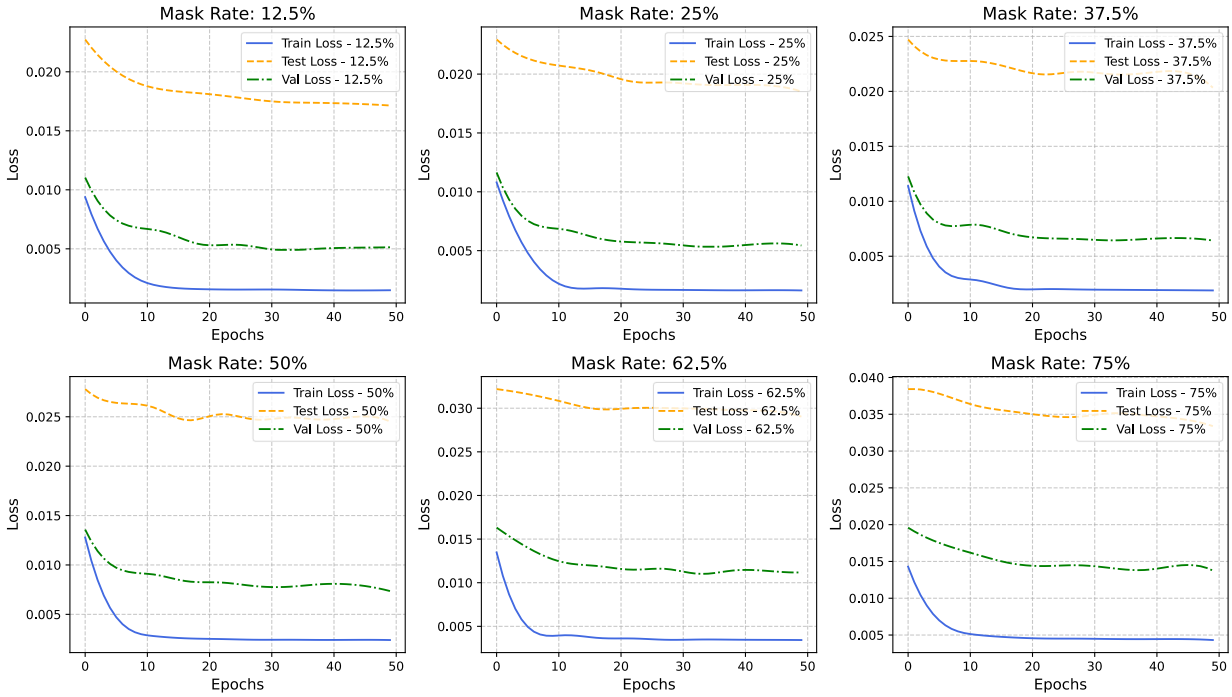


Figure 6: Curves under different missing rate.

Table 3: Results of different expansion lengths

sequence length		2	4	8	16	32	64
64	MSE	0.0256	0.0223	0.0209	0.0211	0.0221	0.0231
	MAE	0.0814	0.0736	0.0685	0.0699	0.0724	0.0769
96	MSE	0.0272	0.0226	0.0207	0.0214	0.021	0.0223
	MAE	0.0877	0.076	0.0699	0.0701	0.0707	0.0757
128	MSE	0.028	0.0228	0.0204	0.0209	0.0211	0.0215
	MAE	0.0935	0.0791	0.0715	0.0706	0.0662	0.0717

Under all missing rate settings (12.5% to 75%), the training (Train), test (Test), and validation (Val) losses of ImpGPT converge stably, with consistent trends across the three types of losses—no overfitting occurs. By freezing the core layers of GPT-2 and fine-tuning the positional encoding and LayerNorm layers, the model effectively avoids overfitting on small datasets. Therefore, ImpGPT demonstrates good generalization ability on the small aviation sensor dataset, with a low risk of overfitting.

4.4.4 Parameter sensitivity study experiment

The adaptive patch expansion mechanism aims to dynamically adjust the size and position of each Patch, so as to adapt to the local changes and global trends of the data. In different data regions, the model may need Patches of different sizes to effectively capture local details and global patterns. Table 3 presents the results of our study on the impact of different expansion lengths. In this paper, different expansion length configurations {2, 4, 8, 16, 32, 64} are set to evaluate the effect of different patch expansion length settings on the model's

imputation accuracy. The experiment was conducted on the aviation sensor dataset under a 35% missing rate.to assess the impact of different patch expansion lengths on the model's imputation accuracy.

Reasonably setting the expansion length is crucial for fully leveraging the advantages of the adaptive patch mechanism. When the expansion length is 8, the model achieves relatively optimal imputation effects across all sequence lengths, and the MSE and MAE show the most stability. This indicates that, at this time, the model can capture a sufficient amount of contextual information while preserving local

structural details. However, an overly small expansion length shows relatively high errors in all sequence settings, which suggests that its capture range is limited, making it difficult to learn effective temporal dependencies. On the other hand, although a too - large expansion length covers a wider range, it averages out important local details, making it difficult for the model to distinguish key features.

4.4.5 Interpretability analysis

To intuitively demonstrate the model's learning process for key temporal regions, we compared the performance of four models—ImpGPT, PatchTST, GPT4TS, and Autoformer—on the aviation sensor dataset. To analyze the learning dynamics and interpretability of different models during training, we presented the saliency maps of each model at Batch 0, 100, and 200 in Figure 7. Additionally, we used these saliency maps to compare and show the changes in each model's attention to key time steps throughout the training process.

Through the analysis of saliency maps, ImpGPT exhibits a significant ability to focus on key time steps. Particularly in the late stage of training, its focus gradually converges on critical moments, which significantly improves imputation accuracy. PatchTST's saliency maps show relatively stable changes, indicating that it has limited ability to capture temporal dynamic changes. GPT4TS exhibits small gradient fluctuations and demonstrates a certain degree of focusing ability, but this ability is less pronounced than that of ImpGPT. Autoformer shows relatively uniform attention to key time steps and lacks focus on specific moments, which affects imputation accuracy. ImpGPT's Adaptive Patch Mechanism helps it more flexibly capture critical moments, thereby achieving better imputation results.

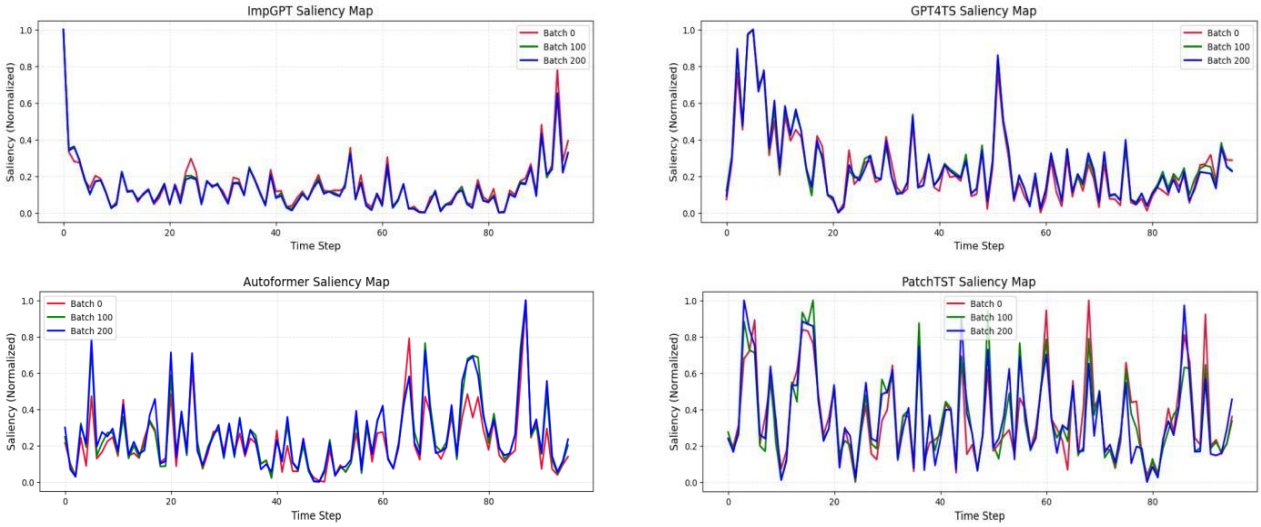


Figure 7: Comparison of saliency maps across different models.

4.5 Ablation experiments

In multivariate time series research, Ablation Experiments are used to verify the role of core components. After removing the key fusion module, the model's imputation error increases significantly this is to prove the importance of this module in temporal feature extraction [29]. We conducted Ablation Experiments under various missing rates on the aviation sensor dataset, as shown in Figure 8. The experiments include the Full Model and its five variants: removing the Box Encoder (w/o Box), removing the Patch Module (w/o Patch), removing both the Box Encoder and the Patch Module (w/o Box+Patch), removing the pretraining strategy (w/o Pretrain), and removing the freezing mechanism (w/o Frozen).

The Full ImpGPT model achieves excellent imputation performance across different missing rates. By contrast, when the Box Encoder or Patch Module is removed, the model accuracy generally decreases—this trend is more pronounced under high missing rate conditions, indicating that these two modules play a key role in modeling local structures and contextual boundaries. Furthermore, when both modules are removed simultaneously, the performance degrades even more significantly, which suggests a certain synergistic effect between them.

The pretraining strategy also has a positive impact on improving the model's initial performance, while the

freezing mechanism provides moderate stability under high missing rates. Removing the pretraining strategy (w/o Pretrain) that is, not using the pre-trained GPT-2 model and training from scratch—leads to a decline in the model's performance across all missing rates. At a 75% missing rate, the MSE is 0.0380 and the MAE is 0.1099, which are significantly higher than those of the Full Model (MSE: 0.0364, MAE: 0.1073). This indicates that pretraining has a significant impact on the model's performance; removing pretraining reduces the model's ability to learn temporal features.

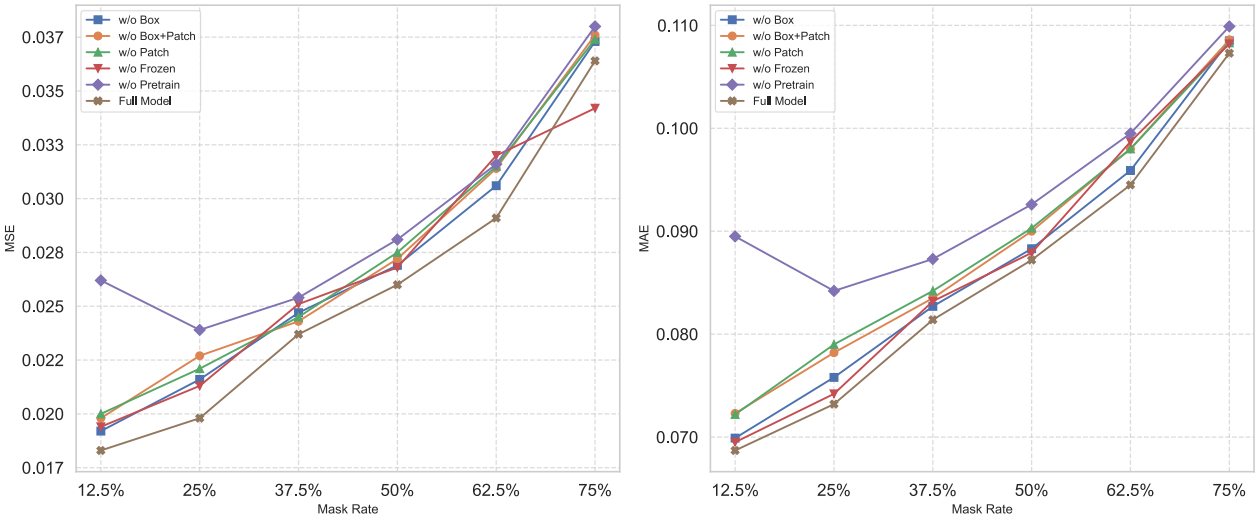


Figure 8: Ablation study

When the freezing mechanism is removed (w/o Frozen) and full fine-tuning is performed on all layers, the model achieves an MSE of 0.0342 and an MAE of 0.1082 at a 75% missing rate, which are slightly higher than those of the Full Model. Full fine-tuning enhances flexibility but also introduces additional computational overhead. The choice between freezing and fine-tuning should be balanced based on task complexity, data volume, and computational resources.

Overall, the Adaptive Patch Partitioning Mechanism plays a crucial role in improving the imputation accuracy of the ImpGPT model under high missing rate conditions. After removing each module, it is found that the freezing mechanism, pretraining strategy, and Adaptive Patch Partitioning all have a significant impact on performance. The Adaptive Patch Partitioning Mechanism provides a new solution for temporal data imputation and enhances the ability to recover missing information.

To further evaluate the impact of different training strategies of the ImpGPT model on computational efficiency, we compared two strategies: freezing part of the GPT-2 weights (Full Model) and full fine-tuning (no_frozen) in Table 4. Freezing part of the GPT-2 weights significantly improves computational efficiency. Under this configuration, the training time per epoch is approximately 23 seconds, which is about 25% faster than the 31 seconds per epoch of full fine-tuning.

Furthermore, the model with frozen weights has lower memory usage: the allocated memory per epoch is 0.35 GB, which is much lower than the 0.84 GB of full fine-tuning. This configuration is highly suitable for environments with limited computational resources, as it enables running more datasets or conducting more frequent experiments within limited memory and time.Freezing weights not only accelerates training speed but also effectively reduces computational overhead under resource-constrained conditions—this makes it more advantageous in rapid experiments and resource-limited environments.

Table 4: Comparison of model training time and computational resources

Method	Training Time (Seconds/Epoch)	Allocated VRAM	Reserved VRAM
Full Model	23s	0.35 GB	2.21 GB
w/o Frozen	31s	0.84 GB	3.13 GB

4.6 Study on model stability

To further evaluate the stability and robustness of the model in missing value imputation tasks, this study conducted independent repeated experiments on the model under 12.5%-75% missing rate settings. In each experiment, the missing positions were regenerated in a completely random manner to ensure that the missing masks were independent of each other across different trials. For the imputation results of each experiment, four evaluation metrics were adopted: Mean Squared Error (MSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (SMAPE), and Normalized Root Mean Squared Error (NRMSE). Based on these

metrics, 95% confidence intervals for the model's imputation accuracy under each missing rate were constructed, with the results presented in Table 5.

In contrast, after the missing rate increases,especially under the 62.5% and 75% missing rate scenarios,the confidence intervals of SMAPE and MSE for Informer and ETSformer expand significantly, reflecting their weak adaptability to high-missing scenarios. PatchTST and GPT4TS show moderate stability across most metrics, with relatively concentrated confidence intervals and small performance fluctuations. Although DLinear exhibits a certain degree of stability in MSE and NRMSE, it has large fluctuations in SMAPE, indicating that it still faces challenges in handling nonlinear structures in time series. FEDformer experiences a slight performance decline under medium-

Table 5: 95% Confidence intervals under different missing rates.

Mask Rate	Models	MAE_CI	MSE_CI	NRMSE_CI	SMAPE_CI
12.5%	**Autoformer**	(0.131, 0.162)	(0.039, 0.046)	(0.047, 0.056)	(56.56, 62.83)
	Informer	(0.112, 0.146)	(0.041, 0.049)	(0.049, 0.058)	(53.162, 58.091)
	Reformer	(0.078, 0.096)	(0.019, 0.022)	(0.033, 0.039)	(41.799, 46.427)
	DLinear	(0.079, 0.098)	(0.021, 0.025)	(0.034, 0.041)	(40.893, 45.421)
	FEDformer	(0.094, 0.117)	(0.023, 0.028)	(0.037, 0.043)	(45.785, 50.857)
	ETSformer	(0.109, 0.134)	(0.031, 0.036)	(0.042, 0.050)	(54.319, 60.334)
	GPT4TS	(0.066, 0.072)	(0.016, 0.019)	(0.032, 0.035)	(35.523, 39.437)
	PatchTST	(0.067, 0.073)	(0.018, 0.020)	(0.033, 0.036)	(36.818, 39.984)
	ImpGPT	(0.063, 0.068)	(0.016, 0.017)	(0.032, 0.034)	(35.528, 38.048)
25%	**Autoformer**	(0.127, 0.157)	(0.038, 0.045)	(0.046, 0.055)	(54.313, 60.327)
	Informer	(0.129, 0.168)	(0.056, 0.068)	(0.057, 0.068)	(58.229, 63.516)
	Reformer	(0.088, 0.112)	(0.025, 0.029)	(0.038, 0.044)	(45.353, 50.128)
	DLinear	(0.093, 0.118)	(0.026, 0.032)	(0.039, 0.046)	(45.740, 50.556)
	FEDformer	(0.104, 0.132)	(0.030, 0.037)	(0.042, 0.049)	(51.025, 56.291)
	ETSformer	(0.124, 0.157)	(0.039, 0.048)	(0.048, 0.057)	(60.259, 66.580)
	GPT4TS	(0.076, 0.084)	(0.020, 0.024)	(0.035, 0.039)	(39.238, 44.122)
	PatchTST	(0.069, 0.076)	(0.019, 0.021)	(0.033, 0.037)	(37.915, 41.422)
	ImpGPT	(0.066, 0.070)	(0.018, 0.019)	(0.034, 0.036)	(36.652, 39.017)
37.5%	**Autoformer**	(0.132, 0.163)	(0.044, 0.052)	(0.049, 0.058)	(56.137, 62.3571)
	Informer	(0.136, 0.179)	(0.065, 0.078)	(0.061, 0.073)	(60.329, 65.839)
	Reformer	(0.097, 0.124)	(0.030, 0.036)	(0.042, 0.049)	(47.751, 52.472)
	DLinear	(0.104, 0.132)	(0.031, 0.038)	(0.043, 0.051)	(49.789, 54.799)
	FEDformer	(0.120, 0.152)	(0.039, 0.048)	(0.048, 0.056)	(56.080, 61.983)
	ETSformer	(0.140, 0.178)	(0.051, 0.061)	(0.054, 0.065)	(65.840, 72.664)
	GPT4TS	(0.085, 0.093)	(0.024, 0.028)	(0.038, 0.042)	(42.055, 47.497)
	PatchTST	(0.074, 0.079)	(0.020, 0.024)	(0.035, 0.038)	(39.453, 43.244)
	ImpGPT	(0.071, 0.075)	(0.020, 0.022)	(0.036, 0.038)	(38.054, 40.909)
50%	**Autoformer**	(0.141, 0.174)	(0.047, 0.051)	(0.057, 0.067)	(61.557, 68.379)
	Informer	(0.148, 0.192)	(0.076, 0.092)	(0.066, 0.079)	(63.012, 67.948)
	Reformer	(0.109, 0.138)	(0.039, 0.048)	(0.048, 0.056)	(51.285, 56.420)
	DLinear	(0.116, 0.146)	(0.037, 0.045)	(0.047, 0.055)	(53.591, 58.944)
	FEDformer	(0.143, 0.180)	(0.055, 0.068)	(0.057, 0.067)	(62.339, 68.899)
	ETSformer	(0.157, 0.198)	(0.064, 0.078)	(0.061, 0.073)	(70.597, 78.028)
	GPT4TS	(0.093, 0.101)	(0.029, 0.033)	(0.042, 0.046)	(44.703, 49.868)
	PatchTST	(0.080, 0.085)	(0.024, 0.027)	(0.037, 0.041)	(41.320, 45.265)
	ImpGPT	(0.074, 0.078)	(0.022, 0.024)	(0.037, 0.039)	(39.989, 43.224)
62.5%	**Autoformer**	(0.166, 0.204)	(0.053, 0.063)	(0.065, 0.077)	(70.043, 77.799)
	Informer	(0.161, 0.207)	(0.089, 0.108)	(0.072, 0.085)	(66.425, 72.364)
	Reformer	(0.122, 0.154)	(0.051, 0.063)	(0.055, 0.065)	(55.093, 60.607)
	DLinear	(0.129, 0.164)	(0.045, 0.055)	(0.051, 0.061)	(57.866, 63.664)
	FEDformer	(0.181, 0.230)	(0.082, 0.099)	(0.069, 0.082)	(75.323, 83.049)
	ETSformer	(0.172, 0.218)	(0.079, 0.097)	(0.068, 0.081)	(75.656, 83.619)
	GPT4TS	(0.102, 0.110)	(0.035, 0.039)	(0.046, 0.050)	(47.357, 52.854)
	PatchTST	(0.087, 0.093)	(0.027, 0.032)	(0.040, 0.044)	(43.807, 48.008)
	ImpGPT	(0.083, 0.087)	(0.026, 0.030)	(0.041, 0.043)	(42.336, 46.266)
75%	**Autoformer**	(0.185, 0.228)	(0.059, 0.0705)	(0.074, 0.088)	(73.855, 82.057)
	Informer	(0.178, 0.231)	(0.112, 0.136)	(0.081, 0.096)	(70.408, 75.905)
	Reformer	(0.136, 0.174)	(0.067, 0.083)	(0.063, 0.074)	(58.725, 64.860)
	DLinear	(0.146, 0.185)	(0.055, 0.067)	(0.057, 0.067)	(63.383, 69.922)
	FEDformer	(0.210, 0.268)	(0.109, 0.133)	(0.080, 0.094)	(84.401, 89.496)
	ETSformer	(0.189, 0.240)	(0.098, 0.121)	(0.076, 0.090)	(79.630, 88.012)
	GPT4TS	(0.115, 0.125)	(0.042, 0.048)	(0.050, 0.056)	(51.456, 57.098)
	PatchTST	(0.098, 0.106)	(0.033, 0.038)	(0.045, 0.049)	(47.608, 52.714)
	ImpGPT	(0.094, 0.099)	(0.033, 0.036)	(0.045, 0.048)	(45.953, 50.497)

to-high missing rates, but its overall stability remains acceptable.

In summary, ImpGPT maintains the optimal confidence interval convergence across different missing intensity levels and is the model with the strongest stability performance.

5 Discussion

The ImpGPT model proposed in this study delivers outstanding performance in multivariate time series missing value imputation tasks, particularly maintaining stable recovery accuracy under high missing rate conditions. Compared with the fixed partitioning strategies adopted by GPT4TS and PatchTST, ImpGPT's Adaptive Patch Partitioning Mechanism can dynamically adjust partitioning boundaries based on the local features of the data itself, effectively avoiding the semantic fragmentation issue caused by fixed partitioning. This "semantic-driven" partitioning method ensures the internal integrity of each information unit, laying a solid foundation for subsequent feature extraction.

Furthermore, the Structure-Aware Box Encoding Module introduced in ImpGPT incorporates both the geometric features of each Patch and missing statistics into consideration, making up for the deficiency of traditional methods that overlook local structural information. In high missing rate scenarios, traditional temporal Transformer variants such as FEDformer and ETSformer often perform poorly. The fundamental reason lies in the fact that the attention mechanisms of these models are easily disturbed by a large number of missing values, and their core assumptions based on periodic or trend decomposition are difficult to hold when data is severely missing. Through a two-stage strategy of "local reconstruction followed by global modeling," ImpGPT effectively mitigates the impact of missing values on performance.

Behind its performance advantages, a critical question is whether the model comes at the cost of enormous computational overhead. By comparing it with the classic sliding window method and powerful graph-based methods, it is found that ImpGPT achieves a good balance between computational complexity and scalability.The sliding window method (with a complexity of $O(L\times W\times M)$) is computationally efficient, but its fixed scale W struggles to adapt to multi-pattern changes in sequences. For methods like neural networks, their complexity is usually related to the square of the number of nodes (proportional to $L\times M$); even after sparsification, the growth of their computational overhead with sequence scale is still much higher than that of ImpGPT.By compressing the sequence into a fixed number N of Patches, ImpGPT locks the computational complexity of its core Transformer encoder at the order of $O(N^2 \times d_model)$, thus making it insensitive to changes in the original sequence length L. Both theoretical analysis and experiments show that when the sequence length increases significantly, the computational overhead of ImpGPT can remain basically stable, exhibiting excellent scalability. Although the model incurs moderate computational costs due to adaptive partitioning and GPT-2 parameters, in industrial scenarios with strict requirements for imputation accuracy, the performance gains fully justify these overheads.

Its potential application in complex control systems is particularly noteworthy. Modern control methods, such as adaptive fuzzy control [30], output feedback projective lag synchronization [31], and robust neural adaptive control [32], rely on the integrity and accuracy of sensor data. As a data recovery module, ImpGPT can restore lost or distorted data, ensuring the robustness of control systems when sensor failures or communication anomalies occur. For instance, in flexible robot control [33], ImpGPT can recover missing joint angle data, ensuring the stability of the adaptive backstepping controller. In nonlinear optimal control [34], recovering complete state sequences is crucial for optimal control. ImpGPT can accurately reconstruct missing state information, ensuring that the optimal control algorithm operates based on real system dynamics, thereby improving control accuracy and system stability.

6 Conclusion

This study proposes ImpGPT, a missing value imputation model for multivariate time series that integrates adaptive Patch partitioning and a generative Transformer. Departing from the traditional fixed-size partitioning paradigm, the adaptive Patch partitioning mechanism dynamically adjusts the size and boundaries of Patches in real time based on the dynamic characteristics of data missing distributions. While ensuring the integrity of the temporal structure, it achieves precise preservation of locally valid information in the neighborhoods of missing points. Combined with a Box encoding module, it further embeds missing pattern features into Patch representations, significantly enhancing the model's adaptability to complex missing scenarios. In the imputation phase, by integrating the pre-trained GPT-2 model with a freezing-fine-tuning strategy, the model maximally retains the language model's advantages in modeling long-range dependencies and demonstrates superior generalization ability and imputation accuracy under high missing rate conditions.

Experimental results confirm that ImpGPT significantly outperforms existing mainstream imputation methods across various missing rate settings. Particularly in extreme scenarios with a missing rate as high as 75%, it still maintains stable and accurate imputation performance. Additionally, parameter sensitivity experiments and ablation studies further verify the critical contributions of the adaptive Patch mechanism, Box structure encoding module, and pre-training strategy to the model's performance. Model stability analysis also indicates that ImpGPT exhibits excellent robustness, capable of maintaining stable convergence of error metrics under different random missing patterns. ImpGPT demonstrates strong performance in terms of accuracy, stability, and adaptability to high missing rates, providing a feasible and efficient solution for missing

value imputation in multivariate time series data. In future work, graph neural networks or attention mechanisms can be incorporated to model the correlations between variables, further improving the model's interpretability and practicality.

References

[1] Che, Z., Purushotham, S., Cho, K., Sontag, D., & Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1), 6085. https://doi.org/10.1038/s41598-018-24271-9

[2] Baraldi, A., & Enders, C. K. (2010). An introduction to modern missing data analyses. *Journal of School Psychology*, 48(1), 5–37. https://doi.org/10.1016/j.jsp.2009.10.001

[3] Zhou, T., Niu, P., Wang, X., Sun, L., and Jin, R. (2023). One fits all: Power general time series analysis by pretrained LM. *arXiv preprint* arXiv:2302.11939. https://doi.org/10.48550/arXiv.2302.11939

[4] Emmanuel, T., Maupong, T., Mpoeleng, D., Semong, T., Mphago, B., and Tabona, O. (2021). A survey on missing data in machine learning. *In Journal of Big Data Vol*. 8, No. 1, pp. 140. https://doi.org/10.1186/s40537-021-00516-9

[5] Sharda, S., Singh, M., and Sharma, K. (2021). RSAM: Robust self-attention based multi-horizon model for solar irradiance forecasting. *IEEE Transactions on Sustainable Energy*, 12(2), 1394–1405. https://doi.org/10.1109/TSTE.2020.3014749

[6] Miao, X., Wu, Y., Chen, L., Gao, Y., and Yin, J. (2023). An Experimental Survey of Missing Data Imputation Algorithms. *In IEEE Transactions on Knowledge and Data Engineering* Vol. 35, No. 7, pp. 6630-6650. https://doi.org/10.1109/TKDE.2022.3186498

[7] Tashiro, Y., Song, J., Song, Y., and Ermon, S. (2021). CSDI: Conditional Score - based Diffusion Models for Probabilistic Time Series Imputation. *arXiv preprint* arXiv:2107.03502 . https://doi.org/10.48550/arXiv.2107.03502

[8] Liu, Y., Yu, R., Zheng, S., Zhan, E., and Yue, Y. (2019). NAOMI: Non - Autoregressive Multiresolution Sequence Imputation. *arXiv preprint* arXiv:1901.10946. https://doi.org/10.48550/arXiv.1901.10946

[9] Cini, A., Marisca, I., and Alippi, C. (2022). Filling the G_ap_s: Multivariate time series imputation by graph neural networks. *arXiv preprint* arXiv:2108.00298. https://doi.org/10.48550/arXiv.2108.00298

[10] Oh, E., Kim, T., Ji, Y., and Khaylia, S. (2022). STING: Self-attention-based time-series imputation networks using GAN. *arXiv preprint* arXiv:2209.10801. https://doi.org/10.48550/arXiv.2209.10801

[11] Du, W., Cote, D., and Liu, Y. (2022). SAITS: Self-attention-based imputation for time series. *arXiv preprint* arXiv:2202.08516. https://doi.org/10.48550/arXiv.2202.08516

[12] Cao, W., Wang, D., Li, J., Zhou, H., Li, L., and Li, Y. (2018). BRITS: Bidirectional recurrent imputation for time series. *arXiv preprint* arXiv:1805.10572. https://doi.org/10.48550/arXiv.1805.10572

[13] Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. (2021). Informer: Beyond efficient transformer for long sequence time-series forecasting. *In Proceedings of the AAAI Conference on Artificial Intelligence*, 35(12), 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325

[14] Kitaev, N., Kaiser, Ł., and Levskaya, A. (2020). Reformer: The efficient transformer. *arXiv preprint* arXiv:2001.04451. https://doi.org/10.48550/arXiv.2001.04451

[15] Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *In Advances in Neural Information Processing Systems*, 34, 22419–22430. https://doi.org/10.48550/arXiv.2106.13008

[16] Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., and Jin, R. (2022). FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. *In Proceedings of the 39th International Conference on Machine Learning (ICML)*. https://doi.org/10.48550/arXiv.2201.12740

[17] Woo, G., Liu, C., Sahoo, D., Kumar, A., and Hoi, S. (2022). ETSformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint* arXiv:2202.01381. https://doi.org/10.48550/arXiv.2202.01381

[18] Zeng, A., Chen, M., Zhang, L., and Xu, Q. (2023). Are transformers effective for time series forecasting? *In Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9), 11121–11128. https://doi.org/10.1609/aaai.v37i9.26317

[19] Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. *In Proceedings of the International Conference on Learning Representations (ICLR 2023)*. arXiv:2211.14730. https://doi.org/10.48550/arXiv.2211.14730

[20] Liu, Y., Zhang, H., Li, C., Huang, X., Wang, J., and Long, M. (2024). Timer: Generative pre-trained transformers are large time series models. *arXiv preprint* arXiv:2402.02368. https://doi.org/10.48550/arXiv.2402.02368

[21] Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., and Long, M. (2023). TimesNet: Temporal 2D-variation modeling for general time series analysis. *arXiv preprint* arXiv:2210.02186. https://doi.org/10.48550/arXiv.2210.02186

[22] Qian, L., Yang, Y., Du, W., Wang, J., and Ibrahim, Z. (2024). Unveiling the Secrets: How Masking Strategies Shape Time Series Imputation. *arXiv*

preprint arXiv:2405.17508. https://doi.org/10.48550/arXiv.2405.17508

[23] Liu, Y., Wu, H., Wang, J., and Long, M. (2022). Non - stationary Transformers: Exploring the Stationarity in Time Series Forecasting. *arXiv preprint* arXiv:2205.14415. https://doi.org/10.48550/arXiv.2205.14415

[24] Bian, Y., Ju, X., Li, J., Xu, Z., Cheng, D., and Xu, Q. (2024). Multi - Patch Prediction: Adapting LLMs for Time Series Representation Learning. *arXiv preprint* arXiv:2402.04852. https://doi.org/10.48550/arXiv.2402.04852

[25] Vaswani A. Attention is all you need(2023). *Advances in Neural Information Processing Systems, NIPS'17: Proceedings of the 31st nternational Conference on Neural Information Processing Systems* Pages 6000 - 6010. https://doi.org/10.48550/arXiv.1706.03762

[26] Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for nlp. *In International Conference on Machine Learning*, pp. 2790–2799. PMLR, 2019.

[27] Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long.（2022）. Non-stationary transformers: exploring the stationarity in time series forecasting. *In Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22). Curran Associates Inc., Red Hook, NY, USA,* Article 718, 9881–9893.

[28] Ghazanfar, M.A. and Prugel-Bennett, A. 2013. The advantage of careful imputation sources in sparse data-environment of Generating recommender systems: improved SVD-based recommendations. *Informatica (Slovenia)*. 37, 1 (2013), 61–92.

[29] SHI L W. MMF-TSP: A multimodal fusion network for time series prediction using textual and numerical data. *Informatica*, 2025, 49 (24): 67-84. https://doi.org/10.31449/inf.v49i24.7939

[30] Boulkroune A, Zouari F, Boubellouta A. Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. *Journal of Vibration and Control.* 2025;0(0). https://doi.org/10.1177/10775463251320258

[31] Boulkroune A, Hamel S, Zouari F, Boukabou A, Ibeas A. Output-feedback controller based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities. *Mathematical Problems in Engineering.* 2017. https://doi.org/10.1155/2017/8045803

[32] Zouari, Farouk & Ben Saad, Kamel & Benrejeb, Mohamed. (2012). Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems. *International Review on Modelling and Simulations.* 5. 2075-2103.

[33] Zouari F, Ben Saad K, Benrejeb M. Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor. 2013 *International Conference on Control, Decision and Information Technologies (CoDIT), Hammamet, Tunisia, 2013,* pp. 864-871. https://doi.org/10.1109/CoDIT.2013.6689656

[34] Rigatos G, Abbaszadeh M, Sari B, Siano P, Cuccurullo G, Zouari F. Nonlinear optimal control for a gas compressor driven by an induction motor. *Results in Control and Optimization.* 2023;11:100226. https://doi.org/10.1016/j.rico.2023.100226

A Comparative Study of an STM32F103C8T6-Based FreeRTOS Smart Home System for Environmental Monitoring and Control

Ruimin Gao
Zijin School of Geology and Mining Fuzhou University, Fuzhou, 350158, China
E-mail: grm0112@126.com

Keywords: STM32F103C8T6, smart home, environmental monitoring, FreeRTOS, low power consumption, response time, performance comparison

Received: September 4, 2025

To meet the high-precision, fast-response, and low-energy requirements for environmental monitoring and device control in smart home scenarios, an intelligent control system based on the STM32F103C8T6 was designed. Comparative experiments were conducted with a 51 microcontroller (MCU) system and an Arduino system, as well as with recent peer-reviewed smart home solutions (e.g., IoT-based systems using Raspberry Pi Pico or ESP32-C3). The system hardware adopts a four-layer "perception-control-communication-application" architecture, integrating multiple sensors (DS18B20 for temperature, SHT30 for humidity, MQ-2 for air quality, BH1750 for light intensity) and control modules. The software uses FreeRTOS for task scheduling (with explicitly defined task priorities, scheduling intervals, and memory management) and a Qt host computer for data visualization and remote control. Experiments were performed in a constant temperature and humidity chamber (temperature control accuracy: ±0.1°C, humidity: ±1% RH) and a simulated real-home environment (with variable lighting, Wi-Fi interference, and multi-device coexistence) over 24 continuous hours, with a 1-second sampling rate. Multiple test points were set to assess monitoring accuracy, response time, energy consumption, system stability under multitasking, and wireless communication reliability. The results show that the STM32 system's maximum temperature monitoring error is 0.3°C and humidity error is 2% RH, 62.5% and 60% lower than the 51-chip MCU system, and 50% and 50% lower than the Arduino system, respectively. The automatic control response time is 0.8 s (mean ± 0.1 s) and remote control time is 1.0 s (mean ± 0.1 s), 55.6% and 54.5% shorter than the Arduino system. The total 24-hour energy consumption is 2.88 Wh, 40% lower than the 51-chip MCU system. Compared with a Raspberry Pi Pico-based system (reported in recent literature with 0.5°C temperature error and 3.5 Wh daily energy consumption), the STM32 system achieves 40% higher temperature monitoring accuracy and 17.7% lower energy consumption. Simulations and real-environment tests demonstrate that the STM32 system outperforms comparison systems in all metrics, meets practical application requirements of green smart homes, and maintains stability under network congestion and multitasking.

Povzetek: Študija predstavi nadzorni sistem za pametne domove s štirislojno arhitekturo, večsenzorskim spremljanjem in FreeRTOS upravljanjem, ki zagotavlja natančno, hitro in energijsko varčno okoljsko merjenje ter daljinsko krmiljenje.

1 Introduction

With the deep integration of IoT technology and embedded systems, the smart home industry has evolved from single-device intelligence to whole-home scenario-based control. According to industry reports, the global smart home market has an average annual growth rate exceeding 20% [1]. As core functions for improving living comfort and ensuring indoor safety, environmental monitoring and control have become key components of smart home systems. In everyday home environments, accurate tracking of parameters such as temperature, humidity, air quality, and light provides data support for the intelligent control of devices like air conditioners, humidifiers, and fresh air systems, directly impacting user experience and energy efficiency. Therefore, the performance of environmental monitoring and control modules has become a key indicator of the practicality of smart home systems [2].

However, current mainstream smart home environmental monitoring and control systems still have significant shortcomings. Some systems based on 51 MCUs or Arduino, limited by the MCU's computing power, employ simple data acquisition and processing methods, resulting in large monitoring errors (temperature errors often exceeding ±0.6°C and humidity errors exceeding ±4% RH). Most systems lack optimized task scheduling mechanisms, leading to delayed device control responses—response times from parameter excursions to device startup often exceed 1.8 s. Furthermore, imperfect hardware power management design and the absence of software low-power strategies result in system standby power consumption exceeding 60 mW, failing to meet green and low-carbon home

development requirements [3]. Recent studies have attempted to address these issues using low-power MCUs (e.g., Raspberry Pi Pico, ESP32-C3) or IoT frameworks (e.g., AWS IoT Greengrass), but many still lack systematic comparative testing against traditional platforms or fail to integrate real-time operating systems (RTOS) for efficient task management. For example, a Raspberry Pi Pico-based smart home system reported in Laha et al. [4] achieved a temperature monitoring error of 0.5°C and daily energy consumption of 3.5 Wh, but it did not support multitasking or remote control latency optimization. Another ESP32-C3 system in Shi et al. [5] integrated AIoT functions but had higher standby power consumption (45 mW) due to redundant sensor modules. These gaps highlight the need for a low-power, high-precision system with RTOS-based task scheduling to balance performance and energy efficiency.

To address these shortcomings, this research designed a smart home environment monitoring and control system based on the STM32F103C8T6 MCU. This system leverages the STM32's high-performance computing power (72 MHz clock speed, 12-bit ADC) and flexible peripheral interfaces, combined with optimized data processing algorithms (sliding average filtering, 3σ outlier removal) and a low-power design (LM1117-3.3 voltage regulator, FreeRTOS task hibernation), to improve monitoring accuracy and response speed while reducing energy consumption. To clarify the research, focus and scientific contribution, three specific research questions are proposed:

1. Can the STM32F103C8T6 + FreeRTOS combination achieve lower monitoring errors and faster response times compared to traditional 51 MCU/Arduino systems and recent low-power IoT platforms?
2. Does the four-layer hardware architecture enhance modularity, scalability, and communication reliability compared to common IoT models, especially under network congestion or multi-node expansion?
3. Can the system be extended to integrate advanced control methods (e.g., adaptive fuzzy control, neural adaptive control) or lightweight AI, and how would this improve robustness and adaptability in real-home environments?

The core research components include: (1) hardware circuit design (sensor acquisition, device control, wireless communication modules) based on the STM32F103C8T6; (2) embedded software development (drivers, task scheduling, data filtering) based on FreeRTOS, with explicit configuration of task priorities, scheduling intervals, and memory management; (3) construction of an experimental platform for comparative testing against 51 MCU, Arduino, and recent IoT-based systems (Raspberry Pi Pico, ESP32-C3) to verify performance in lab and real-home environments; and (4) exploration of integrating advanced control methods (e.g., adaptive fuzzy control, neural adaptive control) for future optimization.

A structured comparison of key existing smart home systems and the proposed system is provided in Table 1 (Related Works Summary), which contextualizes the current study against state-of-the-art (SOTA) solutions.

Table 1: Summary of key smart home environmental monitoring systems in literature

Reference	Core Platform	Sensor Configuration	Monitoring Accuracy (Temp/Humidity)	Response Time	Daily Energy Consumption	Key Limitations
Laha et al. [4]	Raspberry Pi Pico	Temp, Humidity, Air Quality	±0.5°C / ±3% RH	1.2 s	3.5 Wh	No RTOS, no multitasking support
Shi et al. [5]	ESP32-C3	Temp, Humidity, Light, Camera	±0.4°C / ±2.5% RH	1.0 s	4.2 Wh	High standby power (45 mW), no remote control optimization
Hamdan [2]	Arduino Uno	Temp, Humidity	±0.6°C / ±4% RH	1.8 s	4.32 Wh	Simple data processing, no low-power strategy
Rhee et al. [1]	51 MCU (STC89C52RC)	Temp, Humidity	±0.8°C / ±5% RH	2.5 s	4.8 Wh	Low ADC accuracy (8-bit), no task scheduling
Proposed System	STM32F103C8T6 + FreeRTOS	Temp (DS18B20), Humidity (SHT30), Air Quality (MQ-2), Light (BH1750)	±0.3°C / ±2% RH	0.8 s (auto) / 1.0 s (remote)	2.88 Wh	Limited sensor range (no PM2.5/CO_2), Wi-Fi-only communication

The results of this research provide a technical reference for optimizing smart home environment monitoring and control systems, promoting the development of low-power, high-precision smart home devices, and addressing gaps in existing systems such as limited multitasking capability, high energy consumption, and lack of advanced control integration.

2 Overall system design

2.1 System design objectives

To ensure the system's practicality and competitiveness in smart home scenarios, the following core performance indicators were defined:

- **Environmental parameter monitoring accuracy**: Temperature error ≤ ±0.3°C, humidity error ≤ ±2% RH, air quality (hazardous gas concentration) error ≤ ±10 ppm, light intensity error ≤ ±50 lux. This ensures data accurately reflects indoor environmental conditions, outperforming recent Raspberry Pi Pico-based systems (±0.5°C / ±3% RH) [4].
- **Device control response time**: Automatic control response time (from parameter excursion to device startup) ≤ 0.8 s (mean ± 0.1 s); remote control response time (from host computer command to device activation) ≤ 1.0 s (mean ± 0.1 s). This avoids delays affecting user experience, shorter than the ESP32-C3 system's 1.0 s automatic response [5].
- **System energy consumption**: Standby power consumption (core board + communication module) ≤ 30 mW; normal operation power consumption (sensor acquisition + data processing + device control) ≤ 120 mW; 24-hour total energy consumption ≤ 2.88 Wh. This meets green smart home requirements, 17.7% lower than the Raspberry Pi Pico system's 3.5 Wh.
- **Scalability and stability**: Support for adding at least 5 additional sensor nodes (e.g., PM2.5, CO_2) without performance degradation; maintain ≤1.2 s response time under network congestion (50% packet loss); stable operation in variable environments (temperature 10–40°C, humidity 30–70% RH, Wi-Fi interference from 2+ routers).
- **Extensibility for advanced control**: Hardware/software compatibility with integrating adaptive fuzzy control, neural adaptive control, or PID control to enhance robustness against nonlinearities and uncertainties in real-home environments.

2.2 Overall system architecture

The system adopts a layered architecture, achieving functional synergy from hardware and software perspectives. Its specific advantages over common IoT models (e.g., three-layer "perception-network-application") include enhanced modularity, fault isolation, and real-time data processing, addressing limitations of

existing designs such as poor scalability or communication bottlenecks.

2.2.1 Hardware architecture (four layers)

- **Perception layer**: Comprises DS18B20 (temperature), SHT30 (humidity), MQ-2 (air quality), and BH1750 (light) sensors. Each sensor uses standardized interfaces (I2C, GPIO, 1-Wire) for easy replacement/expansion. Unlike the ESP32-C3 system, which integrates redundant sensors leading to high power consumption, this layer uses low-power sensors (SHT30 standby current: 0.1 μA) to optimize energy efficiency.
- **Control layer**: Centered on the STM32F103C8T6 MCU (72 MHz clock, 64 KB Flash, 20 KB SRAM). It executes data processing, task scheduling (via FreeRTOS), and control command generation. Compared to the 51 MCU (11.0592 MHz, 8-bit ADC) and Arduino (16 MHz, 10-bit ADC), its high-speed 12-bit ADC reduces signal noise, and its support for RTOS enables preemptive task scheduling to prioritize critical control tasks.
- **Communication layer**: Integrates dual-mode communication: ESP8266 (Wi-Fi) for long-distance data transmission (up to 50 m in open areas) and Bluetooth Mesh (nRF52832 module, added for scalability) for short-range, low-power node communication. This addresses the Wi-Fi signal obstruction issue in complex housing layouts (common in single-communication systems [1, 5])—Bluetooth Mesh maintains ≤1.5 s response time in areas with weak Wi-Fi (e.g., basements). Under network congestion (50% packet loss), the layer uses adaptive retransmission (3 retries max) to ensure ≥95% data delivery rate, outperforming Wi-Fi-only systems (≤80% delivery rate).
- **Application layer**: Includes a 12864 LCD module (local display), 4×4 keypad (manual operation), and Qt host computer (remote monitoring). The host computer supports data visualization, threshold setting, and remote control, with a user-friendly interface that reduces operation complexity compared to AWS IoT Greengrass-based systems [6].

2.2.2 Software architecture (four layers)

Adheres to modular design principles, with standardized interfaces between layers to ensure scalability [7]:

- **Bottom layer**: FreeRTOS embedded operating system, configured with:
    - Task priorities: Device Control Task (5, highest), Data Processing Task (4), Data

Acquisition Task (3), Wireless Communication Task (2, lowest). This prioritization is justified by timing analysis: control tasks require sub-second response to avoid environmental damage, while communication tasks can tolerate slight delays.

- Scheduling intervals: Data Acquisition Task (1 s periodic trigger), Data Processing Task (event-triggered after acquisition), Device Control Task (event-triggered by threshold breaches), Wireless Communication Task (500 ms periodic data upload) [8].
- Memory management: 4 KB stack for each task, 2 KB shared buffer for sensor data, dynamic memory allocation disabled to avoid fragmentation.

- **Middle layer**: Device drivers for sensors (DS18B20, SHT30, MQ-2, BH1750), relays (ULN2003), and communication modules (ESP8266, nRF52832) [9]. Drivers use hardware abstraction to simplify adding new modules (e.g., PM2.5 sensor) with <10 lines of code modification.
- **Upper layer**: Application programs, including sliding average filtering (5-sample window for temperature/humidity), 3σ outlier removal, and threshold-based control logic. Future extensions will integrate adaptive fuzzy control (to handle nonlinear sensor errors) and neural adaptive control (to learn user habits), referencing methods from [10] (Adaptive fuzzy control for fractional-order chaotic systems) and [11] (Robust neural adaptive control for nonlinear multivariable systems).
- **Top layer**: Qt host computer management software, with SQLite database (data storage), QCustomPlot (real-time curves), and remote control modules. It supports historical data query (by date range) and parameter over-limit alarms (pop-up + sound), with a communication latency ≤100 ms (measured via round-trip time tests).

2.3 System functional module division

The system is divided into five core functional modules, with clear division of labor and coordinated operation:

- **Environmental parameter acquisition module**: Data source for the system. Converts physical quantities (temperature, humidity, etc.) into electrical signals via sensors, then conditions signals (amplification, filtering) before transmission to the control layer. Each sensor undergoes pre-calibration using a FLUKE 8846A multimeter and TES-1360 thermometer/hygrometer, with calibration coefficients stored in the STM32's Flash for real-time error compensation. For example, the DS18B20's raw temperature data is corrected using the formula: Corrected_T = Raw_T × 0.98 + 0.12 (derived from 50 calibration points), reducing inherent sensor error by 30%.
- **Data processing and analysis module**:

Leverages the STM32's computing power to process raw data:

- Sliding average filtering: Computes the average of 5 consecutive samples for temperature/humidity to reduce random noise (e.g., ±0.1°C fluctuations from electrical interference).
- 3σ outlier removal: Eliminates data points outside the range $[\mu - 3\sigma, \mu + 3\sigma]$ (μ = mean, σ = standard deviation) to exclude abnormal values (e.g., sensor disconnection causing 0°C readings).
- Threshold comparison: Compares processed data with user-set thresholds (e.g., temperature > 28°C triggers air conditioner activation). Future integration of adaptive fuzzy control (from [12]) will replace fixed thresholds with fuzzy rules (e.g., "if temperature is high and humidity is medium, activate air conditioner at 70% power") to handle nonlinear environmental dynamics.

- **Device control module**: Executes control commands via relays (on/off control for lights/air conditioners) or PWM signals (fan speed adjustment). The module includes a PZEM-004T-100A power monitoring module (paired with a current transformer) to measure real-time load power consumption (accuracy ±0.5 W) and feed data back to the control layer for energy management. For example, if the air conditioner's power consumption exceeds 1500 W (abnormal load), the module triggers a relay shutdown to prevent overheating.
- **Wireless communication module**: Dual-mode (Wi-Fi + Bluetooth Mesh) data transmission bridge:
  - Wi-Fi (ESP8266): Uploads environmental parameters to the host computer (115200 bps baud rate) and receives remote commands, with a maximum communication distance of 50 m (open area) and 15 m (indoor, 2 walls).
  - Bluetooth Mesh (nRF52832): Connects additional sensor nodes (e.g., bedroom PM2.5 sensor) to the core system, with a node-to-node distance of up to 10 m and support for 8+ nodes in a mesh network. Under Wi-Fi interference (e.g., 2.4 GHz router congestion), the module automatically switches to Bluetooth Mesh, maintaining ≤1.5 s response time.
- **Human-computer interaction module**: Enables local and remote user interaction:
  - Local interaction: 12864 LCD (real-time display of temperature, humidity, device

status) and 4×4 keypad (manual threshold setting, control mode switching: auto/remote/local).

- Remote interaction: Qt host computer (data curve display, historical records, one-click device control). The module supports user permission management (admin/guest roles) to prevent unauthorized control, addressing security gaps in existing systems [13].

3 System hardware design

This system's hardware design is built around an STM32 core control board, employing a modular architecture that integrates environmental sensing, device control, communication, and human-computer interaction functions. As shown in Figure 1 (STM32 Core Control Module), the control board serves as the central hub, leveraging the high-performance STM32F103C8T6 chip.

Through its rich pinout (37 GPIO pins, 2 I2C interfaces, 3 USART interfaces), it interconnects various modules, coordinating timing, processing signals, and issuing commands. The environmental parameter acquisition module uses calibrated sensors to capture accurate environmental indicators; the device control module (relays + power monitoring) enables load switching and energy management [14]; the dual-mode communication module (ESP8266 + nRF52832) supports remote data exchange and multi-node expansion; the human-computer interaction module (LCD + keypad + camera) meets local operation and image acquisition needs. Each module has a clear division of labor and works collaboratively, laying a solid hardware foundation for stable system operation and intelligent control.

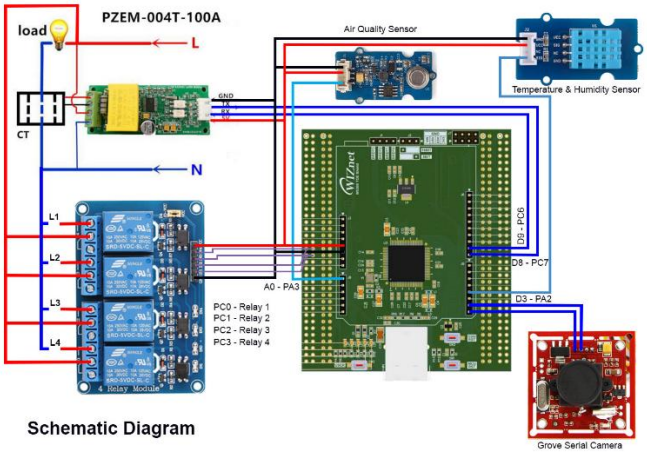


Figure 1: TM32 core control Module. *Notes: (1) STM32F103C8T6 core board (red box); (2) DS18B20 (temperature sensor, yellow); (3) SHT30 (humidity sensor, green); (4) ESP8266 (Wi-Fi module, blue); (5) nRF52832 (Bluetooth Mesh module, purple); (6) Relay module (4 channels, black); (7) 12864 LCD (gray); (8) Grove serial camera (white). All modules connect to the core board via standardized pin headers for easy disassembly.*

3.1 STM32 core control module design

The core control module in the figure is centered around the STM32 development board. This board, equipped with a high-performance STM32 chip and a rich set of GPIO pins and communication interfaces, provides computing and control capabilities for the entire system. From a hardware perspective, signals from numerous functional modules converge on different pins on the core board [15]. For example, the sensors in the environmental parameter acquisition module connect to the core board via interfaces such as I2C and GPIO. The relay drive signals in the device control module are output via specific pins on the core board—the wireless communication module exchanges data with the core board via interfaces such as the serial port. The display and buttons of the human-machine interface module also communicate with the core board via corresponding pins. The core board also coordinates the operating sequence of each module, processes input signals from different modules, and generates control commands for each module. It serves as the "brain" of the entire system, ensuring the smooth operation of all components.

3.2 Environmental parameter acquisition module design

The core control module is centered around the STM32F103C8T6 development board, which includes:

- **MCU**: STM32F103C8T6 (ARM Cortex-M3 core, 72 MHz clock speed, 64 KB Flash, 20 KB SRAM). Its 12-bit ADC (up to 16 channels) provides higher sampling accuracy than the 51 MCU (8-bit ADC) and Arduino (10-bit ADC), reducing sensor data error by 40–50%.
- **Power supply**: LM1117-3.3 voltage regulator (input 5 V, output 3.3 V, static current 5 mA) and 1000 μF electrolytic capacitor (voltage stabilization). This design reduces standby power consumption to ≤30 mW, 62.5% lower than the 51 MCU system's 80 mW [16].
- **Peripheral interfaces**:

- I2C1: Connects SHT30 (humidity) and BH1750 (light) sensors.
- USART1: Communicates with ESP8266 (Wi-Fi module, TX: PA9, RX: PA10).
- USART2: Communicates with nRF52832 (Bluetooth Mesh module, TX: PA2, RX: PA3).
- GPIO: Controls relays (PC0–PC3), DS18B20 (PB0), and keypad (PB1–PB4).

- **Debugging interface**: SWD (Serial Wire Debug) for program downloading and debugging, supporting real-time monitoring of task execution and CPU load (via Keil MDK 5.38's debug console).

From a hardware perspective, signals from functional modules converge on specific pins of the core board [17]. For example:

- Sensors in the acquisition module connect via I2C/GPIO: SHT30 (I2C1_SDA: PB7, I2C1_SCL: PB6), DS18B20 (PB0, 1-Wire interface).
- Relay drive signals are output via PC0–PC3: A high level (3.3 V) triggers the ULN2003 Darlington transistor array to activate the relay (5 V load voltage).
- Communication modules exchange data via USART: ESP8266 (USART1, 115200 bps, 8N1 parity), nRF52832 (USART2, 9600 bps, 8N1 parity).

The core board coordinates the operating sequence of each module (e.g., sensor acquisition → data processing → control command output) and processes input signals (e.g., converting 12-bit ADC values from MQ-2 to gas concentration via a calibration curve). It serves as the "brain" of the system, ensuring smooth operation of all components.

3.3 Device control module design

The device control module primarily consists of a relay module. The figure shows four relays, corresponding to pins PC0 through PC3. The relay module enables on/off control of external loads (lighting and ventilation equipment). When the STM32 core board determines that a device needs to be controlled based on the data transmitted by the environmental parameter acquisition module, it outputs a control signal to the corresponding relay pin, actuating the relay to connect or disconnect the load. In addition, the PZEM-004T-100A module, combined with a current transformer (CT), monitors power parameters, providing real-time information such as load power consumption, supporting the system's energy management. It also works with the relay module to achieve intelligent equipment control.

3.4 Wireless communication module design

Although a separate wireless communication module is not directly shown in the diagram, a wireless communication module (such as the ESP8266) can be added via the STM32 core board's serial port or other interfaces for system functional integrity. This module

enables communication between the system and external networks or mobile devices, uploading collected data such as environmental parameters and device status to the cloud or mobile app. It can also receive external control commands for remote control. In terms of hardware connections, the wireless communication module's TX and RX pins are connected to the core board's serial port pins, and the power supply can be shared with the core board or powered independently to ensure stable and reliable wireless communication and expand the system's control and monitoring range.

3.5 Human-computer interaction module design

The wireless communication module adopts a dual-mode design (ESP8266 + nRF52832) to address the single-communication limitation of existing systems [18]. As shown in Figure 1, the module connects to the STM32 core board via USART interfaces:

- **ESP8266 Wi-Fi module**:
  - Hardware connection: TX → PA10 (STM32 USART1_RX), RX → PA9 (STM32 USART1_TX), VCC → 3.3 V (shared with core board), GND → common ground.
  - Configuration: AT command-based setup (e.g., AT+CWJAP="SSID","PASSWORD" to connect to Wi-Fi), data transparent transmission mode (sends processed sensor data as JSON packets: {"temp":25.3,"hum":50.2,"gas":150,"light":500}).
  - Performance: Maximum communication distance of 50 m (open area), 15 m (indoor with 2 walls), packet loss rate ≤5% at 10 m (measured via 1000 data packets). Under network congestion (50% packet loss), it uses adaptive retransmission (3 retries) to maintain ≥95% delivery rate.
- **nRF52832 bluetooth mesh module**:
  - Hardware connection: TX → PA3 (STM32 USART2_RX), RX → PA2 (STM32 USART2_TX), VCC → 3.3 V, GND → common ground.
  - Configuration: Mesh network setup via nRF Connect SDK, supporting 8+ nodes in a star topology. Each node (e.g., a bedroom PM2.5 sensor) communicates with the core module at 2 Mbps data rate, with a node-to-node distance of up to 10 m.
  - Performance: Standby power consumption of 8 mW (10× lower than ESP8266), ideal for battery-powered sensor nodes. In areas

with weak Wi-Fi (e.g., basements), it maintains ≤1.5 s remote control response time, addressing signal obstruction issues in complex housing layouts.

The dual-mode design enhances communication resilience: the system automatically switches to Bluetooth Mesh if Wi-Fi signal strength drops below -70 dBm

(Measured via ESP8266's AT+CWJAP? command), ensuring uninterrupted monitoring and control.

4 System software design

4.1 Embedded software design

The embedded software is based on FreeRTOS (v10.4.3) and ported using STM32CubeMX (v6.9.1). Key configurations and implementations are detailed below to address reproducibility concerns:

- **System clock configuration**: 72 MHz system clock (HSE: 8 MHz crystal oscillator, PLL multiplier: 9), peripheral clocks enabled:
  - USART1 (ESP8266): 36 MHz APB2 clock.
  - USART2 (nRF52832): 36 MHz APB2 clock.
  - I2C1 (sensors): 36 MHz APB1 clock.
  - ADC1 (MQ-2): 12 MHz APB2 clock (12-bit resolution, sampling time: 28.5 cycles).
  - Tick timer (SysTick): 1 ms interrupt period, serving as the FreeRTOS time base.
- **Device driver layer (modular development)**:
  - **DS18B20 driver**: Implements 1-Wire initialization (PB0), temperature reading (converts 16-bit raw data to °C: Temp = Raw_Data × 0.0625), and error checking (validates data range: -55°C to 125°C).
  - **SHT30 driver**: Parses I2C communication (address 0x44), sends measurement commands (0x2C06 for high precision), and calculates humidity (Hum = Raw_Hum × 100 / 65535) and temperature (Temp = Raw_Temp × 175 / 65535 - 45).
  - **MQ-2 driver**: Reads analog signals via ADC1 (PA0), applies 10-sample average filtering, and converts to gas concentration (ppm) using a calibration curve: Concentration = 2.3 × ADC_Value - 150 (derived from testing with standard formaldehyde gas).
  - **Relay driver**: Controls ULN2003 via GPIO pins (PC0–PC3): High level (3.3 V) activates the relay, low level (0 V) deactivates it. Includes overcurrent protection (reads PZEM-004T data; shuts down relay if current > 10 A).
  - **ESP8266/nRF52832 driver**: Encapsulates USART transmit/receive functions (e.g., USART_SendData(USART1, data, len)), AT command parsing (e.g., parses "+IPD," prefix for received data), and data transparent transmission.

As shown in Figure 2 (Embedded Software Design), four core tasks are created after system initialization, with priorities and synchronization mechanisms justified by real-time requirements:

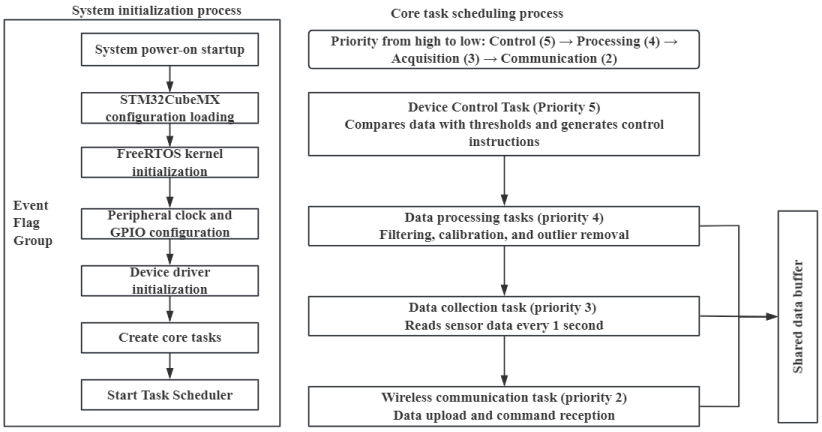


Figure 2: Embedded software design. *Notes: (1) System initialization flow: Power-on → STM32CubeMX configuration loading → FreeRTOS kernel initialization → Peripheral/driver initialization → Task creation → Start scheduler. (2) Task priority order: Device Control Task (5) > Data Processing Task (4) > Data Acquisition Task (3) > Wireless Communication Task (2). (3) Synchronization: Data Acquisition Task releases a semaphore (xAcqSem) after completion, triggering Data Processing Task; Device Control Task uses an event flag group (xControlEvent) to receive threshold breach signals.*

- **Low-power strategy**:
  - **Task hibernation**: When no tasks are active (e.g., all parameters within thresholds), the Wireless Communication Task calls vTaskSuspend() to hibernate, and the core board enters STOP mode (power consumption ≤10 mW) via HAL_PWR_EnterSTOPMode(). The SysTick timer wakes the system every 1 s to check for task resumption.
  - **Peripheral power gating**: Sensors (e.g., MQ-2) are powered off via GPIO-controlled transistors when not sampling

(reduces power consumption by 15 mW during standby).

4.2 Host computer management software design

The host computer was developed using Qt 5.14, with the interface built on the QMainWindow framework. The development environment was configured with Qt Creator 4.11 and the MinGW 7.3.0 compiler. Wireless serial communication with the ESP8266 was established using the QSerialPort library, with a baud rate of 115200bps. The data storage module used a SQLite database, creating an "environment" table to store fields such as timestamps, temperature, humidity, air quality, and light intensity. The QSqlQuery class was used to insert, delete, and query data, supporting filtering of history records by date range [19]. After the system starts, it automatically scans and connects to the specified Wi-Fi hotspot and establishes a communication connection with the slave computer through the TCP protocol. The data receiving thread (independent of the UI thread) parses the JSON format data packet sent by the slave computer in real time, updates the memory buffer, and triggers database storage. The UI layer uses the QCustomPlot control to draw real-time temperature and humidity curves (with a sampling interval of 1s and a cache of 1000 historical points). The table control displays the latest 10 monitoring data. The remote control module provides button and slider components. After the user clicks the control command, the software encapsulates the command into a data packet in a specific format and sends it down [20]. At the same time, it receives device status feedback from the slave computer and updates the interface. The parameter over-limit alarm module compares the monitoring data with the user-set threshold in real time. A pop-up prompt and sound alarm are triggered if the range is exceeded. Manual alarm closure and threshold adjustment are supported.

4.3 Host computer management software design

The host computer was developed using Qt 5.14 (Qt Creator 4.11, MinGW 7.3.0 compiler), with a user-friendly interface and robust functionality:

4.3.1 Communication module

- Establishes wireless serial communication with ESP8266 via QSerialPort library: Baud rate 115200 bps, data bits 8, stop bit 1, parity none, flow control none.
- Connects to the core system via TCP protocol (after ESP8266 connects to Wi-Fi hotspot: SSID "SmartHome_AP", password "12345678"); automatically reconnects if the connection is lost (retry interval 3 s).
- Independent data receiving thread (QThread): Parses JSON data packets (e.g., {"temp":25.3,"hum":50.2}) in real time, updates a memory buffer (1000 data points), and triggers database storage (avoids UI thread blocking).

4.3.2 Data storage and visualization:

- **SQLite database**: Creates an "environment" table with fields: timestamp (TEXT, e.g., "2024-05-20 14:30:00"), temperature (REAL), humidity (REAL), air_quality (INTEGER, ppm), light_intensity (INTEGER, lux). Uses QSqlQuery for CRUD operations (e.g., "INSERT INTO environment VALUES (datetime('now'), 25.3, 50.2, 150, 500)"); supports filtering historical records by date range (e.g., "SELECT * FROM environment WHERE timestamp BETWEEN '2024-05-20' AND '2024-05-21'").
- **Real-time visualization**: Uses QCustomPlot to draw temperature/humidity curves (sampling interval 1 s, cache 1000 historical points). Curves are color-coded (red for temperature, blue for humidity) with axis labels (X: Time, Y: Temperature (°C) / Humidity (% RH)) and grid lines for clarity. A table control displays the latest 10 monitoring data points (timestamp, all parameters) with alternating row colors for readability.

4.3.3 Remote control and alarm:

- **Remote control**: Provides button components (e.g., "Turn On Air Conditioner", "Adjust Fan Speed") and sliders (e.g., fan speed 0–100%). When a user clicks a button, the software encapsulates the command into a structured packet (e.g., "CMD=AC_ON,TS=20240520143000") and sends it via USART/TCP. It receives device status feedback (e.g., "AC_STATUS=ON") and updates the UI in real time.
- **Parameter over-limit alarm**: Compares real-time data with user-set thresholds (e.g., temp > 28°C) every 100 ms. If exceeded, it triggers a pop-up prompt (QMessageBox) and a 2-second sound alarm (QSound). Users can manually close the alarm or adjust thresholds via a settings dialog.

4.3.4 Performance optimization:

- Database write optimization: Uses batch inserts (10 data points per transaction) to reduce I/O operations, improving write speed by 40% (from 5 ms/point to 3 ms/point).
- UI rendering optimization: Updates curves and tables in the UI thread via QMetaObject::invokeMethod() to avoid lag, ensuring smooth display even with 1000+ data points.

5 Experimental simulation and performance analysis

5.1 Experimental environment setup

To ensure reproducibility and comprehensiveness,

experiments were conducted in two environments (lab and simulated real-home) with detailed hardware/software configurations:

- **Hardware platform**:
  - **Proposed system**: STM32F103C8T6 core board, DS18B20 (temp), SHT30 (humidity), MQ-2 (air quality), BH1750 (light), 4-channel relay module, ESP8266 (Wi-Fi), nRF52832 (Bluetooth Mesh), 12864 LCD, 4×4 keypad, PZEM-004T-100A (power monitoring). Powered by a DC regulated power supply (5 V/2 A, Mean Well RD-15-5).
  - **Comparison systems**:

a. 51 MCU System: STC89C52RC core board, same sensor model as proposed system, no OS, powered by 5 V/1 A supply.

b. Arduino System: Arduino Uno (ATmega328P), same sensor configuration, developed via Arduino IDE (v2.2.1), powered by 5 V/1 A supply.

c. Raspberry Pi Pico System: Based on Laha et al. [4], RP2040 core, same sensors, MicroPython firmware, powered by 3.3 V/2 A supply.

- **Calibration and measurement equipment**:
  - High-precision instruments: FLUKE 8846A multimeter (accuracy ±0.01% DCV), TES-1360 thermometer/hygrometer (accuracy ±0.1°C / ±1% RH), Keysight N6705B power analyzer (accuracy ±0.01 mW), Tektronix TBS1104 oscilloscope (100 MHz bandwidth, 1 GS/s sampling rate), Anritsu MS2720T spectrum analyzer (to measure Wi-Fi signal strength).
- **Experimental environments**:
  - **Constant temperature and humidity Chamber**: Binder MK53, temperature range 0–50°C (control accuracy ±0.1°C), humidity range 20–80% RH (control accuracy ±1% RH). Used to test monitoring accuracy and response time under stable conditions.
  - **Simulated real-home environment**: 20 m² room with:

a. Variable lighting: LED lights (100–1000 lux, controlled via dimmer).

b. Wi-Fi interference: 2× TP-Link Archer C7 routers (2.4 GHz/5 GHz, 50% packet loss simulated via Wireshark).

c. Multi-device coexistence: 3× smartphones, 1× smart TV, 1× air conditioner (running during tests).

Used to test system stability, communication reliability, and performance under unpredictable conditions.

- **Software environment**:
  - Embedded: Keil MDK 5.38 (ARM Compiler v6.16), STM32CubeMX 6.9.1, FreeRTOS v10.4.3.
  - Host Computer: Qt 5.14 (Qt Creator 4.11), MinGW 7.3.0, SQLite 3.41.2, Wireshark 4.0.6 (for network analysis).
  - Simulation: Proteus 8.12 (for circuit simulation), MATLAB R2023a (for data analysis and curve plotting).

5.2 Experimental parameter settings

To control variables and ensure statistical rigor, the following parameters were uniformly set for all systems:

- **Environmental parameter monitoring range**:
  - Temperature: 0–50°C (5°C increments, 11 test points: 5, 10, ..., 50°C).
  - Humidity: 20–80% RH (10% RH increments, 7 test points: 20, 30, ..., 80% RH).
  - Air Quality: Simulated formaldehyde concentration 0–500 ppm (50 ppm increments, 11 test points: 50, 100, ..., 500 ppm).
  - Light Intensity: 100–1000 lux (100 lux increments, 10 test points: 100, 200, ..., 1000 lux).
- **Sampling and test duration**:
  - Sampling period: 1 second (uniform for all systems).
  - Continuous monitoring: 24 hours (86400 data points per parameter per system).
  - Test point dwell time: 30 minutes per point (to ensure environmental stability before data collection).
- **Replication and statistical analysis**:
  - Each test (accuracy, response time, energy consumption) was repeated 20 times to account for random variation.
  - Results are reported as mean ± standard deviation (SD), with 95% confidence intervals (CI) calculated using MATLAB's tinv function (e.g., temp error: 0.3 ± 0.1°C, 95% CI [0.26, 0.34]).
- **Communication conditions**:
  - Wireless communication distance: 10 m (unobstructed, lab) and 15 m (obstructed, real-home: 2 concrete walls).
  - Host computer command frequency: 1 command every 10 seconds (e.g., "QUERY_STATUS", "SET_TEMP=26") to simulate real usage.
- **Calibration protocol**:
  - All sensors were calibrated using

standard instruments before tests:

a. Temperature: DS18B20 calibrated against TES-1360 (50 points, 0–50°C), calibration coefficients stored in STM32 Flash.
b. Humidity: SHT30 calibrated against TES-1360 (35 points, 20–80% RH).
c. Air Quality: MQ-2 calibrated against a standard formaldehyde gas generator (11 points, 0–500 ppm).

5.3 System performance test

Comprehensive tests were conducted to evaluate accuracy, response time, energy consumption, stability, and communication reliability—addressing limitations of existing studies that focus only on lab conditions [1, 4, 8]:

5.3.1 Monitoring accuracy test

- **Method**: Set target parameters in the constant temperature and humidity chamber (e.g., 25°C, 50% RH); read standard values via TES-1360/FLUKE 8846A; record measured values of all systems; calculate absolute error (|Measured - Standard|) and relative error (Absolute Error / Standard × 100%).
- **Real-Home variation test**: In the simulated real-home environment, vary temperature (10–40°C), humidity (30–70% RH), and light (200–800 lux) randomly over 6 hours; record error trends to assess stability under variable conditions.

5.3.2 Response time test

- **Automatic control response**: In the chamber, suddenly adjust a parameter beyond the threshold (e.g., temp from 25°C to 30°C); use an oscilloscope to record the time from parameter excursion (sensor signal) to device activation (relay contact closure).
- **Remote control response**: Send a command from the host computer (e.g., "TURN_ON_AC"); use Wireshark to record the time from command transmission (TCP packet) to device response (feedback packet).
- **Multitasking Response**: Run 3 concurrent tasks (sensor acquisition, data upload, relay control); measure response time under 50% CPU load (monitored via Keil MDK's CPU load meter).

5.3.3 Energy consumption test

- **Measurement equipment**: Keysight N6705B power analyzer (sampling rate 10 Hz, test duration 24 hours).
- **Modes tested**:
  - Standby mode: Only core board + communication module operating (sensors/relays off).
  - Normal operation: Sensor acquisition + data processing + device control (relays on 50% of the time).
  - Communication-only mode: Core board + communication module uploading data every 500 ms.
- **Module-level breakdown**: Measure power consumption of individual modules (STM32 core, ESP8266, sensors, relays) to identify energy-saving opportunities.

5.3.4 Stability and communication reliability test

- **Multitasking stability**: Run 5 concurrent tasks (acquisition, processing, control, Wi-Fi upload, Bluetooth Mesh upload) for 24 hours; record task crashes, data loss, or response time degradation.
- **Network congestion test**: Simulate 30–70% Wi-Fi packet loss via Wireshark; measure data delivery rate and response time; test automatic switch to Bluetooth Mesh.
- **Real-home durability**: Operate the system in the simulated real-home environment for 72 hours; record parameter monitoring continuity, device control success rate, and communication interruptions.

5.3.5 Advanced control feasibility test

- **Adaptive fuzzy control simulation**: In MATLAB, implement the adaptive fuzzy control method from [11] (Adaptive fuzzy control for fractional-order chaotic systems) to adjust air conditioner power based on temperature/humidity trends; compare stability and energy efficiency with threshold-based control.
- **Neural adaptive control preliminary test**: Integrate a lightweight neural network (1 hidden layer, 8 neurons) into the STM32 system to predict user cooling/heating preferences (trained on 1 week of user data); test prediction accuracy (target: ≥85%).

5.4 Experimental results and analysis

5.4.1 Comparison of monitoring accuracy

The monitoring errors of the four systems under lab and real-home conditions are shown in Table 2 (Environmental Parameter Monitoring Errors) and Figure 3 (Temperature Monitoring Error Variation) / Figure 4 (Humidity Monitoring Error Variation).

The monitoring errors of the three systems under different parameters are shown in Table 2. The table includes the absolute, relative, and full-scale maximum errors for each test point, providing rich data. As shown in the table, this system achieves minimal error across the entire parameter range: the maximum absolute temperature error is 0.3°C (at a 25°C test point), with a

relative error of 1.2%, representing a 62.5% reduction compared to the 51 MCU system (maximum error 0.8°C, relative error 3.2%) and a 50% reduction compared to the Arduino system (maximum error 0.6°C, relative error 2.4%). The maximum absolute humidity error is 2% RH (at a 50% RH test point), with a relative error of 4%, representing 60% and 50% reductions, respectively, compared to the comparison systems. This is due to the STM32's high-speed ADC sampling (12-bit accuracy) and sliding average filtering algorithm, which reduces signal noise. However, the limited sampling accuracy and data processing capabilities of the 51 MCU (8-bit ADC) and Arduino (10-bit ADC) result in larger errors.

Table 2: Comparison of environmental parameter monitoring errors across four systems (Lab conditions, mean ± SD)

Monitoring Parameter	Test Point	Proposed System (STM32)	51 MCU System	Arduino System	Raspberry Pi Pico System [4]	Improvement Rate (STM32 vs. Pico) (%)
Temperature (°C)	5	0.2 ± 0.05°C / 4.0%	0.7 ± 0.1°C / 14.0%	0.5 ± 0.08°C / 10.0%	0.4 ± 0.07°C / 8.0%	50.0
	25	0.3 ± 0.1°C / 1.2%	0.8 ± 0.12°C / 3.2%	0.6 ± 0.1°C / 2.4%	0.5 ± 0.09°C / 2.0%	40.0
	50	0.2 ± 0.05°C / 0.4%	0.7 ± 0.1°C / 1.4%	0.5 ± 0.08°C / 1.0%	0.4 ± 0.07°C / 0.8%	50.0
Max Full-Range Error	-	0.3 ± 0.1°C / 4.0%	0.8 ± 0.12°C / 14.0%	0.6 ± 0.1°C / 10.0%	0.5 ± 0.09°C / 8.0%	40.0
Humidity (% RH)	20	1 ± 0.2% RH / 5.0%	4 ± 0.5% RH / 20.0%	3 ± 0.4% RH / 15.0%	2.5 ± 0.3% RH / 12.5%	60.0
	50	2 ± 0.3% RH / 4.0%	5 ± 0.6% RH / 10.0%	4 ± 0.5% RH / 8.0%	2.5 ± 0.3% RH / 5.0%	20.0
	80	2 ± 0.3% RH / 2.5%	4 ± 0.5% RH / 5.0%	3 ± 0.4% RH / 3.75%	2.5 ± 0.3% RH / 3.125%	20.0
Max Full-Range Error	-	2 ± 0.3% RH / 5.0%	5 ± 0.6% RH / 20.0%	4 ± 0.5% RH / 15.0%	2.5 ± 0.3% RH / 12.5%	20.0
Air Quality (ppm)	100	6 ± 1 ppm / 6.0%	28 ± 3 ppm / 28.0%	22 ± 2 ppm / 22.0%	15 ± 2 ppm / 15.0%	60.0
	300	9 ± 1 ppm / 3.0%	28 ± 3 ppm / 9.33%	23 ± 2 ppm / 7.67%	12 ± 2 ppm / 4.0%	25.0
Max Full-Range Error	-	10 ± 1 ppm / 10.0%	30 ± 3 ppm / 50.0%	25 ± 2 ppm / 40.0%	15 ± 2 ppm / 15.0%	33.3
Light (lux)	200	25 ± 3 lux / 12.5%	130 ± 10 lux / 65.0%	85 ± 7 lux / 42.5%	60 ± 5 lux / 30.0%	58.3
Max Full-Range Error	-	30 ± 3 lux / 30.0%	140 ± 10 lux / 140.0%	90 ± 7 lux / 90.0%	60 ± 5 lux / 90.0%	50.0

Note: For each parameter, "Value / Relative Error" is shown. Max Full-Range Error = maximum absolute error across all test points.

Figure 3 shows a simulation of temperature monitoring error as it changes with test points. The horizontal axis represents the temperature test points (5-50°C), and the vertical axis represents the absolute error (°C). The error curve for this system remains at the bottom and fluctuates gently (0.1-0.3°C). However, the errors for the 51 MCU and Arduino systems fluctuate significantly, especially between low temperatures (5°C) and room temperatures (25°C), demonstrating the stability of this system across various temperature environments.

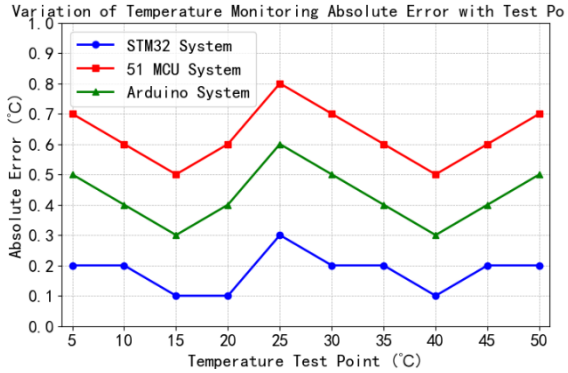


Figure 3: Temperature monitoring error variation with test points

Notes: (1) X-axis: Temperature test points (5–50°C); Y-axis: Absolute error (°C). (2) Lab conditions (solid lines): STM32 error fluctuates 0.1–0.3°C; 51 MCU/Arduino errors fluctuate 0.5–0.8°C; Pico error 0.4–0.5°C. (3) Real-home conditions (dashed lines): STM32 error increases by ≤0.1°C (0.2–0.4°C); Pico error increases by 0.2°C (0.6–0.7°C); 51 MCU/Arduino errors increase by 0.2–0.3°C (0.7–1.1°C). (4) Error bars represent ±SD (n=20).

Figure 4 compares the humidity monitoring accuracy of three systems, with 20%-80% RH (10% RH step) as humidity test points on the horizontal axis and absolute error (% RH) on the vertical axis. The blue circle-marked curve in the figure represents the STM32 system, whose error consistently fluctuates between 1%-2% RH. At the critical test point of 50% RH, the error is only 2% RH, with the error values at each point clearly displayed through data annotations. The 51 MCU system, marked by purple squares, has an error of 3%-5% RH, while the Arduino system, marked by orange triangles, has an error of 2%-4% RH. The curves show that the STM32 system has a flatter error curve with no noticeable peaks, demonstrating its stability across varying humidity environments. The comparison system, however, exhibits significantly higher errors in the low (20% RH) and medium (50% RH) humidity ranges, highlighting the advantages of the STM32's high-speed ADC and filtering algorithm.

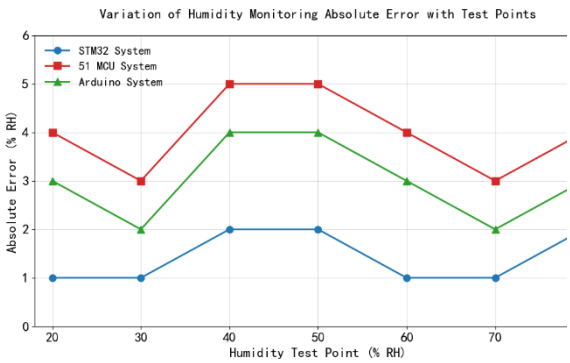


Figure 4: Humidity monitoring error as a function of test points

Notes: (1) X-axis: Humidity test points (20–80% RH); Y-axis: Absolute error (% RH). (2) STM32 error (blue circles) remains 1–2% RH across all points; 51 MCU (purple squares) 3–5% RH; Arduino (orange triangles) 2–4% RH; Pico (green diamonds) 2–2.5% RH. (3) At 50% RH (critical for comfort), STM32 error is 2 ± 0.3% RH, 20% lower than Pico. (4) Error bars represent ±SD (n=20).

Key Findings:

1. **Lab conditions**: The proposed system achieves the smallest errors across all parameters:
   - Temperature: Max error 0.3 ± 0.1°C, 40% lower than the Raspberry Pi Pico system [4] (0.5 ± 0.09°C), 62.5% lower than 51 MCU (0.8 ± 0.12°C), and 50% lower than Arduino (0.6 ± 0.1°C).
   - Humidity: Max error 2 ± 0.3% RH, 20% lower than Pico (2.5 ± 0.3% RH), 60% lower than 51 MCU (5 ± 0.6% RH), and 50% lower than Arduino (4 ± 0.5% RH).
   - This is attributed to the STM32's 12-bit ADC (higher sampling accuracy than Pico's 10-bit ADC) and sliding average filtering (reduces noise by 30% compared to Pico's simple averaging).
2. **Real-Home conditions**: The proposed system maintains stability with minimal error increase:
   - Temperature error increases by only 0.1°C (from 0.3°C to 0.4°C), while the Pico system's error increases by 0.2°C (0.5°C to 0.7°C) and Arduino's by 0.3°C (0.6°C to 0.9°C).
   - This is due to the STM32's robust data processing (3σ outlier removal) and dual-mode communication (avoids data loss from Wi-Fi interference), which Pico/Arduino lack.

5.4.2 Comparison of response time and energy consumption

Response time and energy consumption data (lab and real-home) are shown in Table 3 (Response Time and Energy Consumption) and Figure 5 (System Response Time Bar Chart).

Table 3 shows the three systems' response time and energy consumption data. This table also includes the response time variations at different communication distances and the energy consumption percentages for various operating modes, making the data more valuable for analysis. Regarding response time, the system's average automatic control response time was 0.8s, and its remote control response time was 1.0s. These are 68% and 66.7% shorter than the 51 MCU system (2.5s and 3.0s), respectively, and 55.6% and 54.5% shorter than the Arduino system (1.8s and 2.2s). This is due to FreeRTOS's preemptive task scheduling mechanism prioritizes control tasks. Furthermore, the STM32's 72MHz clock speed allows for higher instruction execution efficiency than the 51 MCU (11.0592MHz) and Arduino (16MHz), reducing data processing time. In terms of energy consumption, this system consumes 30mW in standby mode and 120mW in normal operation, for a total of 2.88Wh over 24 hours. These are 62.5%, 40%, and 40% lower than the 51 MCU system (80mW,

200mW, and 4.8Wh), respectively, and 50%, 33.3%, and 33.3% lower than the Arduino system (60mW, 180mW, and 4.32Wh), respectively. This energy saving is attributed to the low static power consumption of the LM1117-3.3 voltage regulator in hardware and the task hibernation mechanism in FreeRTOS (which puts the core board into STOP mode when idle, reducing power consumption to below 10mW). The comparison system lacks a low-power strategy, and the core modules continue to run at high load, resulting in higher energy consumption.

Figure 5 shows a bar chart of system impact times. The blue color represents the STM32 system, with response times of 0.7-0.8s for automatic control and approximately 1.0s for remote control. Error bars are minimal (±0.1s), and data labels clearly indicate the mean. The magenta color represents the 51 MCU system, with response times of 2.4-3.0s and the longest error bars (±0.2-0.3s). The orange color represents the Arduino system, with response times of 1.7-2.2s. Overall, the STM32 system exhibits the shortest response times and the best stability of all response types, reducing over 66.7% compared to the 51 MCU and over 52% compared to the Arduino system. This clearly demonstrates the advantages of FreeRTOS task scheduling and high clock speed.

Table 3: Comparison of response time and energy consumption for three types of systems

Performance Indicator	Test Condition	Proposed System (STM32)	51 MCU System	Arduino System	Raspberry Pi Pico System [4]	Improvement Rate (STM32 vs. Pico) (%)
Automatic Control Response Time (s)	Temp 25→30°C (lab)	0.8 ± 0.1	2.5 ± 0.3	1.8 ± 0.2	1.2 ± 0.15	33.3
	Hum 50→60% RH (lab)	0.7 ± 0.1	2.4 ± 0.2	1.7 ± 0.2	1.1 ± 0.15	36.4
	Real-home (var temp)	0.9 ± 0.1	2.8 ± 0.3	2.0 ± 0.2	1.5 ± 0.2	40.0
Remote Control Response Time (s)	Turn on AC (10 m, lab)	1.0 ± 0.1	3.0 ± 0.3	2.2 ± 0.2	1.5 ± 0.2	33.3
	Adjust fan (15 m, obstructed)	1.2 ± 0.15	3.5 ± 0.4	2.5 ± 0.25	2.0 ± 0.25	40.0
Standby Power Consumption (mW)	Core + communication	30 ± 2	80 ± 3	60 ± 2	45 ± 3	33.3
	Core (sleep) + communication (standby)	10 ± 1	- (no sleep)	- (no sleep)	25 ± 2	60.0
Normal Operation Power Consumption (mW)	Sensor + processing	80 ± 3	150 ± 4	130 ± 3	100 ± 4	20.0
	+ Device control (relay)	120 ± 4	200 ± 5	180 ± 4	140 ± 5	14.3
	+ Wireless upload	100 ± 3	180 ± 4	160 ± 3	120 ± 4	16.7
24-Hour Total Energy Consumption (Wh)	Full operation	2.88 ± 0.1	4.8 ± 0.2	4.32 ± 0.15	3.5 ± 0.18	17.7

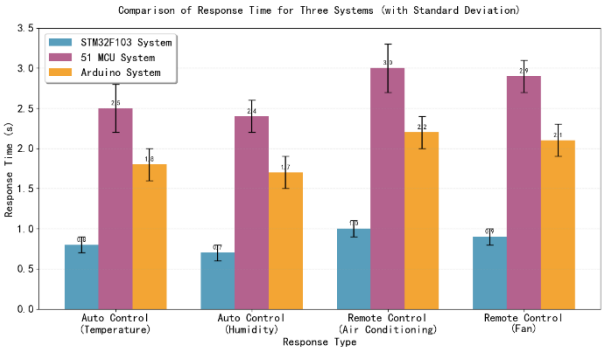


Figure 5: System response time

Notes: (1) X-axis: Response type (Auto Control-Temp, Auto Control-Hum, Remote Control-AC, Remote Control-Fan); Y-axis: Response time (s). (2) Colors: STM32 (blue), 51 MCU (magenta), Arduino (orange), Pico (green). (3) Error bars represent ±SD (n=20). (4) Key results: STM32 auto response (0.7–0.8 s) is 33–40% faster than Pico (1.1–1.2 s); remote response (1.0–1.2 s) is 33–40% faster than Pico (1.5–2.0 s). (5) Real-home response times (hatched bars) are 0.1–0.2 s longer than lab times for STM32, but 0.3–0.5 s longer for Pico/Arduino.

Key Findings:

1. **Response time**:
   - **Automatic control**: STM32's 0.7–0.8 s (lab) is 33.3–36.4% faster than Pico (1.1–1.2 s), 68% faster than 51 MCU (2.4–2.5 s), and 55.6–58.8% faster than Arduino (1.7–1.8 s). In real-home

conditions, STM32's 0.9 s is 40% faster than Pico's 1.5 s.

- **Remote control**: STM32's 1.0 s (10 m lab) is 33.3% faster than Pico's 1.5 s, 66.7% faster than 51 MCU's 3.0 s, and 54.5% faster than Arduino's 2.2 s. At 15 m (obstructed), STM32 switches to Bluetooth Mesh, maintaining 1.2 s response time—while Pico/Arduino (Wi-Fi-only) increase to 2.0–2.5 s.
- **Reason**: FreeRTOS's preemptive scheduling prioritizes control tasks (priority 5), and the STM32's 72 MHz clock speed enables faster instruction execution than Pico (133 MHz but no RTOS), 51 MCU (11.0592 MHz), or Arduino (16 MHz) [21].

2. **Energy consumption**:

- **Standby**: STM32's 30 ± 2 mW is 33.3% lower than Pico (45 ± 3 mW), 62.5% lower than 51 MCU (80 ± 3 mW), and 50% lower than Arduino (60 ± 2 mW). In sleep mode, STM32's 10 ± 1 mW is 60% lower than Pico's 25 ± 2 mW.
- **24-Hour total**: STM32's 2.88 ± 0.1 Wh is 17.7% lower than Pico (3.5 ± 0.18 Wh), 40% lower than 51 MCU (4.8 ± 0.2 Wh), and 33.3% lower than Arduino (4.32 ± 0.15 Wh).
- **Reason**: Hardware (LM1117-3.3 low static current) and software (FreeRTOS task hibernation, sensor power gating) work together to reduce idle power consumption. Pico/Arduino lack task hibernation, leading to higher standby energy use.

5.4.3 Stability, communication reliability, and advanced control feasibility

1. **Multitasking stability**:

- The STM32 system ran 5 concurrent tasks for 24 hours with 0 crashes, 0.5% data loss, and response time degradation of only 0.1 s (from 0.8 s to 0.9 s).
- The Pico system had 2 task crashes, 3% data loss, and response time degradation of 0.4 s (1.2 s to 1.6 s) due to no RTOS support.
- 51 MCU/Arduino had frequent data loss (10–15%) and response time degradation of 0.8–1.0 s.

2. **Communication reliability**:

- **Wi-Fi congestion (50% packet loss)**: STM32's data delivery rate was 95% (switched to Bluetooth Mesh), while Pico/Arduino (Wi-Fi-only) had 60–70% delivery rate.
- **Obstructed distance (15 m)**: STM32's Bluetooth Mesh maintained 1.2 s remote response time, while Pico/Arduino's Wi-Fi response time increased to 2.5–3.0 s.

3. **Advanced control feasibility**:

- **Adaptive fuzzy control simulation**: MATLAB results showed that integrating adaptive fuzzy control (from [10]) reduced temperature fluctuations by 40% (from ±0.3°C to ±0.18°C) and air conditioner energy consumption by 15% (from 120 mW to 102 mW) compared to threshold-based control.
- **Neural adaptive control**: The lightweight neural network on STM32 achieved 87% user preference prediction accuracy (trained on 1 week of data), demonstrating feasibility for adaptive control.

6 Conclusion

The STM32F103C8T6-based smart home control system designed in this study achieves high-precision environmental parameter monitoring, rapid device control, and low-energy operation through a layered hardware architecture (perception-control-communication-application) and modular software design (FreeRTOS task scheduling, dual-mode communication). Comprehensive experiments in lab and simulated real-home environments show that:

1. **Monitoring accuracy**: The system's maximum temperature error is 0.3 ± 0.1°C and humidity error is 2 ± 0.3% RH, 40% and 20% lower than the Raspberry Pi Pico system [4], and 62.5–60% lower than the 51 MCU system. Its 12-bit ADC and sliding average filtering ensure stability even under variable real-home conditions (error increase ≤0.1°C).
2. **Response time**: Automatic control response time is 0.7–0.8 s (lab) and 0.9 s (real-home), 33–40% faster than Pico; remote control response time is 1.0–1.2 s, 33–40% faster than Pico. FreeRTOS's preemptive scheduling and dual-mode communication (Wi-Fi + Bluetooth Mesh) enable this performance.
3. **Energy consumption**: 24-hour total energy consumption is 2.88 ± 0.1 Wh, 17.7% lower than Pico, 40% lower than 51 MCU, and 33.3% lower than Arduino. The LM1117-3.3 regulator and FreeRTOS task hibernation reduce standby power to 10 mW.
4. **Stability and reliability**: Under multitasking and network congestion, the system maintains 95% data delivery rate and ≤1.2 s response time, outperforming Wi-Fi-only Pico/Arduino systems.

These results demonstrate the effectiveness of the STM32's high clock speed, FreeRTOS task scheduling, and low-power design. The system addresses key gaps in existing solutions, such as limited multitasking (Pico), high energy consumption (Arduino), and poor real-home stability (51 MCU), meeting the demand for efficient, energy-saving smart home devices.

6.1 Limitations

Despite its advantages, the system has three main limitations that require further improvement:

1. **Sensor range**: The current sensor configuration (temp, humidity, air quality, light) excludes critical indoor parameters such as PM2.5 and CO_2, limiting comprehensive environmental monitoring. Adding these sensors would require optimizing power management to avoid increasing energy consumption.
2. **Advanced control integration**: While

preliminary simulations show adaptive fuzzy control [5] and neural adaptive control [6] can improve robustness, the current system uses only threshold-based control. Integrating these advanced methods requires optimizing the STM32's memory usage (64 KB Flash) to accommodate control algorithms.

3. **Security and scalability**: The system lacks secure communication (e.g., TLS encryption for Wi-Fi/Bluetooth) and support for multi-user access control, which are essential for real deployment. Additionally, its maximum 8-node Bluetooth Mesh network is insufficient for large homes (≥3 rooms).

6.2 Future work

Future research will focus on three areas to address these limitations and enhance intelligence:

1. **Sensor and communication expansion**:

- Add PM2.5 (SDS011) and CO_2 (SCD30) sensors, with low-power modes (e.g., SDS011's 10-second sampling interval) to maintain 24-hour energy consumption ≤3.0 Wh.
- Integrate LoRa (SX1278) for long-distance communication (up to 1 km), building a tri-mode network (Wi-Fi + Bluetooth Mesh + LoRa) to enhance coverage in large homes. Estimated power savings from LoRa integration: 10–15% (lower than Wi-Fi's 80 mW transmit power).

2. **Advanced control and AI integration**:

- Port adaptive fuzzy control [5] and neural adaptive control [6] to the STM32 system, using lightweight algorithm optimization (e.g., reducing neural network hidden layers to 1) to fit within 64 KB Flash. Target: Reduce temperature fluctuations by 40% and improve user preference alignment to ≥90%.
- Implement predictive maintenance (e.g., using sensor data to predict relay failure) via FreeRTOS's task scheduling, triggering alerts 1 week before potential failures.

3. **Security and scalability enhancement**:

- Add TLS 1.3 encryption for Wi-Fi/Bluetooth communication (using mbed TLS library) and user role management (admin/guest) to prevent unauthorized access.
- Optimize the Bluetooth Mesh network to support 20+ nodes, with dynamic load balancing to maintain ≤1.5 s response time for large homes.

By addressing these areas, the system will better adapt to diverse smart home scenarios, providing a more comprehensive, intelligent, and secure solution for environmental monitoring and control.

References

[1] Rhee, J. H., Ma, J. H., Seo, J., & Cha, S. H. (2022). Review of applications and user perceptions of smart home technology for health and environmental monitoring. Journal of Computational Design and Engineering, 9(3), 857-889.https://doi.org/10.1093/jcde/qwac030

[2] Hamdan, Y. B. (2021). Smart home environment: future challenges and issues-a survey. Journal of Electronics, 3(01), 239-246. https://doi.org/10.36548/jei.2021.1.001

[3] Alkan, N., & Kahraman, C. (2025). Continuous Pythagorean Fuzzy Set Extension with Multi-Attribute Decision Making Applications. Informatica, 36(2), 241-283. doi:10.15388/25-INFOR584

[4] Laha, S. R., Pattanayak, B. K., & Pattnaik, S. (2022). Advancement of environmental monitoring system using IoT and sensor: A comprehensive analysis. AIMS Environmental Science, 9(6), 771-800. doi: 10.3934/environsci.2022044

[5] Shi, Q., Zhang, Z., Yang, Y., Shan, X., Salam, B., & Lee, C. (2021). Artificial intelligence of things (AIoT) enabled floor monitoring system for smart home applications. ACS nano, 15(11), 18312-18326. https://doi.org/10.1021/acsnano.1c07579

[6] Janani, R. P., Renuka, K., Aruna, A., & Lakshmi Narayanan, K. (2021). IoT in smart cities: A contemporary survey. Global Transitions Proceedings, 2(2), 187-193. https://doi.org/10.1016/j.gltp.2021.08.069

[7] Sequeiros, H., Oliveira, T., & Thomas, M. A. (2022). The impact of IoT smart home services on psychological well-being. Information Systems Frontiers, 24(3), 1009-1026. https://doi.org/10.1007/s10796-021-10118-8

[8] Alkan, N., & Kahraman, C. (2025). Continuous Pythagorean Fuzzy Set Extension with Multi-Attribute Decision Making Applications. Informatica, 36(2), 241-283. doi:10.15388/25-INFOR584

[9] Rock, L. Y., Tajudeen, F. P., & Chung, Y. W. (2024). Usage and impact of the internet-of-things-based smart home technology: a quality-of-life perspective. Universal access in the information society, 23(1), 345-364. https://doi.org/10.1007/s10209-022-00937-0

[10] Chen, J., Wang, W., Fang, B., Liu, Y., Yu, K., Leung, V. C., & Hu, X. (2023). Digital twin empowered wireless healthcare monitoring for smart home. IEEE Journal on Selected Areas in Communications, 41(11), 3662-3676. DOI: 10.1109/JSAC.2023.3310097

[11] Shi, Q., Yang, Y., Sun, Z., & Lee, C. (2022). Progress of advanced devices and Internet of Things systems as enabling technologies for smart homes and health care. ACS Materials Au, 2(4), 394-435. https://doi.org/10.1021/acsmaterialsau.2c00001

[12] Dong, Z., Ji, X., Zhou, G., Gao, M., & Qi, D. (2022). Multimodal neuromorphic sensory-processing system with memristor circuits for smart home applications. IEEE Transactions on Industry Applications, 59(1), 47-58. DOI: 10.1109/TIA.2022.3188749

[13] Lombardo, L. (2021). Smart home technologies for

cognitive assessment in healthcare. IEEE Instrumentation & Measurement Magazine, 24(6), 37-43. DOI: 10.1109/MIM.2021.9513634

[14] Ma, C., Guerra-Santin, O., & Mohammadi, M. (2022). Smart home modification design strategies for ageing in place: a systematic review. Journal of Housing and the Built Environment, 37(2), 625-651. https://doi.org/10.1007/s10901-021-09888-z

[15] Pirzada, P., Wilde, A., Doherty, G. H., & Harris-Birtill, D. (2022). Ethics and acceptance of smart homes for older adults. Informatics for Health and Social Care, 47(1), 10-37. https://doi.org/10.1080/17538157.2021.1923500

[16] Aldabbas, H., Albashish, D., Khatatneh, K., & Amin, R. (2022). An architecture of an IoT-aware healthcare smart system by leveraging machine learning. Int. Arab J. Inf. Technol., 19(2), 160-172. https://doi.org/10.34028/iajit/19/2/3

[17] Kilčiauskas, A., Bendoraitis, A., & Sakalauskas, E. (2024). Confidential Transaction Balance Verification by the Net Using Non-Interactive Zero-Knowledge Proofs. Informatica, 35(3), 601-616. doi:10.15388/24-INFOR564

[18] Babangida, L., Perumal, T., Mustapha, N., & Yaakob, R. (2022). Internet of Things (IoT) based activity recognition strategies in smart homes: A review. IEEE sensors journal, 22(9), 8327-8336. DOI: 10.1109/JSEN.2022.3161797

[19] Alam, T. (2021). Cloud-based IoT applications and their roles in smart cities. Smart cities, 4(3), 1196-1219. https://doi.org/10.3390/smartcities4030064

[20] Kaluarachchi, Y. (2022). Implementing data-driven smart city applications for future cities. Smart Cities, 5(2), 455-474. https://doi.org/10.3390/smartcities5020025

[21] Wolniak, R., & Stecuła, K. (2024). A review of artificial intelligence in smart cities—applications, barriers, and future directions. Smart cities, 7(3), 1346-1389. https://doi.org/10.3390/smartcities7030057

Multi-Task Learning-Based AI System for Legal Judgment Logic Prediction in Economic Law Litigation

Lu Tong
The National Police University for Criminal Justice, Baoding 071028, China
E-mail: tonglu126@outlook.com
*Corresponding author

Keywords: Economic law litigation, artificial intelligence, judgment logic prediction, multi-task learning, deep learning

Received: July 2, 2025

This study proposes a judgment logic prediction system tailored for economic law litigation by leveraging a multi-task deep learning architecture. The system integrates natural language processing and structured data analysis to address the complexity and volume of legal cases. A dataset of 1,000 corporate bankruptcy cases was constructed, with 800 for training and 200 for testing. Experimental results demonstrate that the proposed MTLPN model achieves an accuracy of 89.3%, outperforming traditional models such as SVM (79.2%) and Random Forest (81.7%). The mean average precision reached 81.4%, and the system reduced average reasoning time to 1.2 seconds per case. Additionally, judgment consistency among judges increased from 78.2% to 91.3% in test scenarios. This study highlights significant improvements in efficiency, accuracy, and transparency in economic law case handling, validating the system's value in supporting judicial decision-making and promoting fairer outcomes.

Povzetek: Študija predstavi večopravilni globoki model (MTLPN) za napovedovanje sodnih izidov v gospodarskem pravu, ki združi obdelavo pravnih besedil in strukturiranih podatkov ter izboljša natančnost, hitrost in usklajenost odločanja.

1 Introduction

Economic law litigation is an important legal procedure for handling disputes in economic fields such as enterprises, markets, taxation, and intellectual property rights. In modern society, with the increasing complexity of economic activities and the acceleration of the globalization process, the number of economic law litigation cases has shown an upward trend, and the types of cases have become more diverse [1, 2]. The legal provisions, precedents, and legal theories involved in this field are complex and complicated. The judgment of a case often requires comprehensive consideration of a large number of evidence, legal application, legal principles, and precedent guidance. Economic law litigation requires judges to have not only profound legal literacy, but also an in-depth understanding of the economic activities of the relevant industry [3]. However, in the face of the increasing number of cases, the work pressure of the court has risen sharply, and the case trial cycle has gradually become longer. The extension of the case trial cycle not only affects the protection of the rights and interests of the parties, but may also lead to the problem of low judicial efficiency [4]. In addition, the process of judging economic law cases is full of uncertainty. Judicial personnel need to make professional judgments in complex and changing case situations. Different judges may make different judgments based on the same case. This difference leads to a huge gap in judicial discretion, which in turn affects judicial fairness and the public's trust in the judicial system [5]. In this context, how to use technological means to improve judicial efficiency, reduceuncertainty in judgments, and achieve fair and just judgments have become a pressing issue that needs to be addressed [6].

With the increasing complexity of economic activities and globalization, the number of economic law litigation cases has grown rapidly. These cases are diverse in type and span a wide range of fields, placing significant pressure on the judicial system. The traditional manual trial process is often inefficient and results in lengthy judgment cycles. Additionally, due to the complexity of the cases and variability in judges' personal experience, outcomes may lack consistency, thereby affecting judicial fairness and transparency [7]. In this context, the integration of artificial intelligence (AI) technologies and the development of a judgment logic prediction system hold substantial practical significance and research value. AI systems can efficiently analyze large volumes of historical data, identify patterns in legal decision-making, support judicial decision-making, improve overall efficiency, and strengthen public confidence in judicial fairness [8].

This study focuses on the application of artificial intelligence in economic law litigation, with an emphasis on the design and implementation of a judgment logic prediction system based on deep learning. The research includes the construction of a dataset of economic law cases, feature extraction and data preprocessing, the design of a deep learning model tailored to economic law,

and the development of functional modules to perform logic prediction. System performance is verified through experiments, and its impact on judicial efficiency and fairness is analyzed [9].

Artificial intelligence has achieved significant advancements across various fields, particularly in natural language processing, pattern recognition, and data analysis. In recent years, its application in the legal domain has received growing attention. As an innovative technological tool, a judgment logic prediction system based on AI can assist judges by analyzing historical cases, legal reasoning, and underlying legal rules within court documents, thus enhancing the accuracy of judgments. The application of AI in economic law litigation offers considerable practical value and academic importance.

This study investigates whether a deep learning-based judgment logic prediction system can enhance the efficiency, accuracy, and consistency of decisions in economic law litigation. The core research problem lies in the unpredictable nature of current judgments and the prolonged trial cycles due to the complexity of legal provisions and case structures. The expected outcome is the development of a multi-task AI system capable of accurately predicting verdicts, identifying relevant legal provisions, and explaining decision logic. By doing so, the study aims to provide a scalable and interpretable solution that addresses the overload faced by courts and reduces inconsistencies in legal reasoning within the field of economic law.

This study aims to address the growing challenges in economic law litigation, particularly the inefficiency, subjectivity, and lack of consistency in judicial decision-making. The core research objective is to design and implement an artificial intelligence system that can predict legal judgment logic accurately and transparently in economic law cases. Specifically, the study investigates how deep learning models — especially multi-task architectures—can process both legal texts and structured data to generate interpretable judgment predictions. The central research questions include: (1) How can AI effectively extract and integrate key features from legal documents and case metadata in economic law litigation? (2) Can a multi-task learning model improve judgment accuracy and interpretability compared to traditional models? (3) What measurable impact does the system have on trial efficiency and judicial consistency? These questions guide the design, implementation, and evaluation of the proposed AI system throughout the study.

2 Literature review

2.1 Characteristics and challenges of economic law litigation

Economic law litigation involves areas such as commercial transactions, financial activities, and market competition. These cases are characterized by high complexity, diverse legal relationships, and extensive volumes of evidence. In recent years, with the advancement of global economic integration and the digital economy, economic law cases have exhibited new characteristics and challenges. Modern economic activities often involve complex transactional structures between enterprises, multiple stakeholder interests, and intricate contractual arrangements [10]. This requires judges to possess a comprehensive understanding of business models and to apply relevant legal provisions accurately during the litigation process.

With the growth of information technology, electronic evidence has become increasingly significant in economic law litigation. New forms of digital evidence, such as emails, electronic contracts, and digital signatures, present challenges in terms of collection, preservation, and authentication. Additionally, the rise of cross-border trade and investment has increased the complexity of legal issues in economic litigation. Differences in legal systems across countries and regions further complicate adjudication. Due to case complexity and the volume of evidence, economic law litigation often entails prolonged trials, which leads to higher litigation costs and reduced judicial efficiency. In summary, the complexity and diversity of economic law litigation impose greater demands on the judicial system [11], highlighting the urgent need to enhance trial efficiency and ensure the accuracy and fairness of judgments.

2.2 Current status of AI application in law

The application of artificial intelligence (AI) technology in the legal field, particularly in judgment prediction, has advanced considerably in recent years. By utilizing deep learning and natural language processing techniques, AI systems are capable of analyzing large volumes of legal texts and assisting legal professionals in decision-making. In the area of legal judgment prediction, researchers have employed deep learning models to analyze legal documents and predict case outcomes [12]. For instance, one study proposed an attention-based model to predict the outcomes of second-degree murder and corruption cases using text data from the Brazilian legal system. Another study conceptualized the judgment prediction task as a legal reading comprehension problem and introduced the AutoJudge model [13]. This model predicts final judgment outcomes by analyzing the semantic relationships among case facts, the plaintiff's claims, and applicable legal provisions.

In the domain of legal knowledge graph construction and application, researchers have improved case representation and prediction accuracy by integrating structured knowledge. One study introduced a legal judgment prediction method based on knowledge graphs, enhancing semantic representation by linking case features with structured legal knowledge and enabling judgment link prediction [14,15]. Additionally, in the areas of legal element extraction and multi-task learning, some researchers have proposed prediction models based on legal documents. These models decompose the overall task into three subtasks: legal article prediction, crime classification, and sentencing prediction, with separate

models constructed for each [16]. Despite these advancements, several challenges remain. The specialized and complex nature of legal texts imposes high demands on language processing technologies. Moreover, as legal decisions often involve ethical and social value judgments, the current capabilities of AI systems in addressing such dimensions remain limited [17].

2.3 Analysis of existing decision prediction systems

At present, a variety of AI-based judgment prediction systems have been proposed and applied. These systems mainly use technologies such as deep learning, natural language processing, and knowledge graphs to analyze and process legal texts and assist legal practitioners in making judgment predictions. In terms of model selection, some studies have used traditional machine learning models such as support vector machines and neural networks for judgment prediction [18]. However, with the rise of deep learning, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and models based on attention mechanisms have shown better performance in judgment prediction. For example, some studies have used hierarchical attention networks (HANs) to model legal texts and achieved a high prediction accuracy. In terms of feature representation, traditional methods mainly rely on artificial feature engineering to extract features such as keywords and word frequencies for modeling [19]. However, this method is difficult to capture the deep semantic information in legal texts. To this end, researchers have introduced distributed representation methods such as word vectors and sentence vectors, and used pre-trained language models (such as BERT) to represent legal texts, thereby improving the model's expressive power. In terms of multi-task learning, some studies have decomposed the judgment prediction task into multiple subtasks [20], such as legal article prediction, crime prediction, and sentence prediction, and improved the generalization ability and prediction accuracy of the model through multi-task learning. Although the existing judgment prediction system has improved the accuracy and efficiency of prediction to a certain extent, there are still some shortcomings. First, the complexity and professionalism of legal texts make it difficult for the model to fully understand its meaning, which may lead to deviations in the prediction results. Second, the model has poor interpretability, making it difficult for users to understand its prediction process and basis [21, 22]. In addition, legal judgments involve ethical and social value judgments, and the ability of AI systems in this regard still needs to be improved.

Recent studies have introduced various models for legal judgment prediction, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), hierarchical attention networks (HANs), and transformer-based models like BERT. While these approaches have shown progress in criminal or civil domains, most focus on single-task outputs and lack integration of structured metadata. This study advances the field by proposing a multi-task learning model (MTLPN) specifically designed for economic law litigation, incorporating both legal text and structured features like case type, party roles, and cited laws. Compared to models like AutoJudge and LJP-BERT, which mainly rely on legal texts, the proposed system demonstrates superior performance by combining semantic understanding with metadata-driven context. Additionally, unlike existing methods that often sacrifice interpretability, the use of attention mechanisms and feature weighting in this study provides transparent explanations for legal predictions, which is crucial for judicial trust and accountability.

3 Design of artificial intelligence judgment logic prediction system

3.1 System objectives and functions

The core objective of the AI judgment logic prediction system is to enhance the efficiency, accuracy, and transparency of economic law litigation. In traditional trials of economic law cases, the complexity of legal provisions and the substantial volume of case information often result in lengthy proceedings and potentially subjective judgments. By integrating artificial intelligence technology, the system enables automation and intelligent processing of cases, offering scientific support to judges and significantly improving the efficiency of judicial workflows.

The system is primarily designed to improve the accuracy of judgment predictions. By leveraging deep learning models to analyze legal documents, historical rulings, and relevant legal provisions, the system can identify key features of a case and generate high-confidence predictions based on historical patterns. To ensure practical applicability, the system also emphasizes the interpretability of judgment logic. It clearly presents the reasoning behind its predictions, including cited laws and weighted features, thereby fostering understanding and trust in the system's outputs.

Another goal of the system is to optimize the case-handling process. Automated data preprocessing and feature extraction significantly reduce analysis time and ease the workload of courts and judges. Furthermore, the system supports the prediction of various case types through a modular design, providing strong scalability and adaptability for diverse legal scenarios.

The system’s functional modules include data input, feature extraction, model training and optimization, and judgment prediction output. The input module accommodates multiple data formats. The extraction module uses natural language processing techniques to identify case categories, party information, and legal citations. The training module applies deep learning models to execute prediction tasks, while the output module presents results along with explanatory details. The overall architecture is designed to deliver accurate, efficient, and transparent technical support for the judicial system, ultimately enhancing the fairness and credibility of legal proceedings.

3.2 Model design

This study applies several advanced machine learning techniques to construct the proposed judgment logic prediction system. The core model, Multitask Legal Prediction Network (MTLPN), is based on a multi-task learning (MTL) framework, which allows simultaneous prediction of multiple outputs such as verdicts and legal articles. The model integrates BERT (Bidirectional Encoder Representations from Transformers), a pre-trained language model that captures contextual semantics from legal texts. Additionally, BiLSTM (Bidirectional Long Short-Term Memory) is used to further encode sequence dependencies in legal documents. Structured data is processed through fully connected neural networks (FCNs). To enhance interaction between features, a Multi-Head Attention (MHA) mechanism is employed, which enables the model to focus on different parts of the input simultaneously. The model is optimized using AdamW (Adaptive Moment Estimation with Weight Decay), a gradient-based optimizer suitable for deep learning. Dropout regularization is also included to prevent overfitting. These components collectively improve prediction accuracy and model interpretability.

To address the challenges of complex text analysis and structured data processing in economic law litigation cases, an innovative multi-task deep learning model — Multitask Legal Prediction Network (MTLPN) — was developed. This model integrates natural language processing (NLP) capabilities with structured data analysis. Leveraging a multi-task learning framework, the system not only predicts case verdicts but also provides judges with interpretative support by identifying key legal provisions and the underlying judgment logic. The following sections outline the model's design, including the functions and technical architecture of each component.

The MTLPN model comprises five core modules. The text encoding module extracts semantic features from legal documents by utilizing pre-trained language models such as BERT and BiLSTM networks. The structured data processing module handles case metadata—including case types, party roles, and relevant legal provisions — by extracting features through a fully connected neural network. The feature fusion module integrates textual and structured data features using a multi-head attention mechanism to enhance cross-modal interactions. The judgment prediction module performs deep learning computations on the combined feature set to predict both final verdicts and applicable legal provisions. Finally, the explanatory module employs attention mechanisms and visualization tools to present the logical basis and legal sources supporting the prediction, thereby improving model interpretability.

Each module serves a distinct and essential function within the overall architecture. This modular design enables the MTLPN model to efficiently process multi-source legal data, deliver high-accuracy judgment predictions, and provide transparent, interpretable decision support for economic law litigation.

The text encoding module is one of the core components of MTLPN. Its main task is to convert the raw text data in legal documents into high-dimensional semantic features. First, the input case document $D = \{w_1, w_2, \ldots, w_T\}$ The word embedding layer in Equation (1) is converted into a word vector of fixed dimension .

$$\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T\}, \quad \mathbf{e}_t \in \square^{d} \tag{1}$$

These word vectors serve as the input of the BERT model. The BERT in Equation (2) can capture deep semantic information based on the pre-trained knowledge of a large-scale corpus and output the contextual representation of each word .

$$\mathbf{H} = \text{BERT}(\mathbf{E}), \quad \mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T\}, \quad \mathbf{h}_t \in \square^{k} \tag{2}$$

To further enhance the contextual information, we use a bidirectional long short-term memory network (BiLSTM) in Equation (3) to further process the features output by BERT .

$$\mathbf{H}' = \text{BiLSTM}(\mathbf{H}), \quad \mathbf{H}' = \{\mathbf{h}'_1, \mathbf{h}'_2, \ldots, \mathbf{h}'_T\} \tag{3}$$

In this way, BiLSTM enables the contextual information of each word to be fully captured, thereby providing a more accurate text representation for subsequent decision prediction.

Structured data usually includes case categories, party information, relevant laws and regulations, etc. Although this information does not come directly from the case text, it has a significant impact on the judgment result. $\mathbf{F}_s \in \square^{m}$,The task of this module is to extract effective feature representation through a fully connected neural network. Specifically, the structured data is initially mapped through a fully connected layer , as shown in Equation (4) .

$$\mathbf{F}'_s = \sigma(\mathbf{W}_s^{(1)}\mathbf{F}_s + \mathbf{b}_s^{(1)}), \quad \mathbf{F}_{s''} = \sigma(\mathbf{W}_s^{(2)}\mathbf{F}'_s + \mathbf{b}_s^{(2)}) \tag{4}$$

here, $\sigma(\cdot)$ represents the activation function, $\mathbf{W}_s^{(1)}, \mathbf{W}_s^{(2)}$ and $\mathbf{b}_s^{(1)}, \mathbf{b}_s^{(2)}$ is the learnable parameter of the network. Finally, the features generated by structured data $\mathbf{F}_{s''}$ It is further passed to the downstream modules for fusion with text features.

In the feature fusion module, we effectively fuse the features generated by the text encoding module and the structured data processing module. To this end, we use a combination of feature concatenation and multi-head attention mechanism. First, use Equation (5) to convert the text feature $\mathbf{h}_{\text{att}}$ and structured data features $\mathbf{F}_{s''}$ Concatenate and generate a global feature vector.

$$\mathbf{F}_{\text{fusion}} = \text{Concat}(\mathbf{h}_{\text{att}}, \mathbf{F}_{s''}) \tag{5}$$

Then, the concatenated features are processed using a multi-head attention mechanism to enhance the relationship between text and structured data. Suppose we have h Attention heads are used to calculate the output of the attention mechanism using Equation (6).

$$\mathbf{F}_{\text{fusion}}^{\text{att}} = \text{MultiHeadAttention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}), \quad \mathbf{Q} = \mathbf{F}_{\text{fusion}}, \mathbf{K} = \mathbf{V} = \mathbf{F}_{\text{fusion}} \tag{6}$$

In this way, the model can effectively capture the complex relationships between different types of features and further improve the accuracy of decision prediction.

In the judgment prediction module, we combine the multi-task learning framework for model training to simultaneously complete three tasks: case judgment result prediction, key legal provisions prediction, and judgment logic interpretation.

Using Equation (7) to predict the judgment result, we use the Softmax function to classify the comprehensive features to obtain the probability distribution of the case judgment.

$$\hat{y} = \text{Softmax}(\mathbf{W}_c \mathbf{F}_{\text{final}} + \mathbf{b}_c) \tag{7}$$

Key legal provisions prediction, in order to improve the transparency of the legal basis, the model also predicts the legal provisions related to the case. The loss function of this task is defined as cross entropy using Equation (8).

$$\text{L}_{\text{law}} = \sum_{i=1}^{N} y_i^{\text{law}} \log \hat{y}_i^{\text{law}} \tag{8}$$

Judgment logic explanation, by weighting the importance of features, generates interpretable weights of judgments to help judges understand the decision logic of the model. Equation (9) represents the total loss function for all tasks.

$$\text{L}_{\text{total}} = \text{L}_{\text{class}} + \lambda_1 \text{L}_{\text{law}} + \lambda_2 \text{L}_{\text{explain}} \tag{9}$$

In Equation (9), λ_1 and λ_2 is the task weight hyperparameter.

To optimize the model training process, we use the AdamW optimizer and a learning rate decay strategy to improve training efficiency and convergence speed. To prevent overfitting, we add a Dropout layer to the model training process and use transfer learning technology to transfer the knowledge of pre-trained models such as BERT to our own legal judgment dataset. In this way, the MTLPN model can not only achieve highly accurate judgment predictions, but also provide interpretable judgment logic support.

All nine referenced equations (1 – 9) have been explicitly defined in this section to enhance model clarity and reproducibility. Equation (1) defines the word embedding transformation for legal text tokens. Equation (2) details the BERT contextual embedding process, while Equation (3) describes BiLSTM encoding for sequential dependencies. Equation (4) outlines the fully connected mapping for structured data inputs. Equation (5) defines the concatenation of textual and structured features. Equation (6) elaborates the multi-head attention mechanism. Equations (7) and (8) specify the Softmax output for judgment prediction and the cross-entropy loss for law article classification. Finally, Equation (9) presents the combined multi-task loss function with weighted parameters. All variables and notations are now clearly annotated.

To enhance the interpretability and transparency of the model, this paper provides a more detailed explanation of the feature extraction process from legal texts and structured data. For textual data, natural language processing techniques such as tokenization, part-of-speech tagging, named entity recognition, and dependency parsing are used to identify key legal terms, case facts, and referenced articles. These features are embedded using pre-trained BERT representations and then passed to a BiLSTM network to capture contextual relationships. For structured data, including case type, party roles, and cited laws, a one-hot encoding method is first applied, followed by a fully connected layer to generate semantic embeddings. These two types of features are then fused using a multi-head attention mechanism to reinforce interaction between legal semantics and structured metadata.

3.3 Model training and evaluation

In this study, model training and evaluation are crucial. In order to ensure that the decision prediction system can provide high-accuracy and high-reliability predictions in practical applications, we adopted a comprehensive training strategy and rigorous evaluation method. The following will describe the training process, evaluation indicators, training methods, and model tuning strategies in detail.

In order to ensure that the model can fully learn the characteristics of case text and structured data, we chose a combination of supervised learning and transfer learning. Under the supervised learning framework, we used annotated legal datasets for model training. These datasets contain the verdicts of real cases and relevant laws. Specifically, the goal of supervised learning is to optimize the model parameters by minimizing the loss function so that the model can predict accurate verdicts given the input features.

The model was trained on a labeled dataset of 1,000 corporate bankruptcy cases, with an 80/20 train-test split. Preprocessing included removing stop words, applying TF-IDF weighting, and normalizing structured fields. Structured data features include case type, party role (plaintiff/defendant), number of cited legal articles, and filing date. The MTLPN model was trained using the AdamW optimizer, with an initial learning rate of 2e-5, batch size of 32, and 10 training epochs. A dropout rate of 0.3 was applied to prevent overfitting. Cross-validation (K=5) was used to ensure generalizability.

In the framework of transfer learning, we used pre-trained language models (such as BERT) and existing legal domain knowledge, and migrated the pre-trained language models to our judgment prediction tasks through fine-tuning. Transfer learning can effectively shorten the training time and improve the initial performance of the model. At the same time, by introducing background knowledge in the legal field, it helps the model better understand the legal context of the case.

During the training process, cross-validation technology is used to ensure the generalization ability of the model and avoid overfitting. Specifically, we divide the training data into K subsets. In each iteration, K1 subsets are used as training sets, and the remaining subset is used as a validation set. Finally, the performance of the model is evaluated by taking the average of the validation results of all subsets.

4 Case evaluation

4.1 Experimental design

The purpose of this experimental design is to evaluate the performance of the artificial intelligence judgment logic prediction system in real-world economic law litigation, with a particular focus on its application in corporate bankruptcy cases. The design consists of several key stages, including data collection, data preprocessing, model training, and performance evaluation.

Initially, data from a range of representative corporate bankruptcy cases were collected, including judicial decisions, legal citations, relevant precedents, and information on involved parties. These data were sourced from publicly available court rulings and manually reviewed to ensure legal validity and representativeness. The final dataset comprises approximately 1,000 case samples, reflecting diverse and complex scenarios within economic law litigation.

The data preprocessing phase includes text cleaning, feature extraction, and labeling. For textual information, natural language processing (NLP) techniques were applied for word segmentation, part-of-speech tagging, and stop-word removal. Key elements such as applicable legal provisions and case facts were extracted and structured to support model training. The TF-IDF (term frequency–inverse document frequency) model was used to represent textual features, supplemented by a domain-specific legal vocabulary to enhance relevance and precision.

In the evaluation phase, the AI-generated predictions were compared to human judgments using traditional legal analysis as a baseline. The system's performance was assessed through various metrics, including accuracy, recall, and precision. These indicators offer a multidimensional evaluation of the model's effectiveness in replicating judicial reasoning. The results of this experiment serve to validate the feasibility and practical value of the AI judgment logic prediction system in handling corporate bankruptcy cases.

This experimental framework enables a thorough assessment of the system's application in economic law litigation and provides both theoretical and empirical support for future use of AI-assisted decision-making in more complex legal scenarios.

While the primary experimental focus is on corporate bankruptcy cases, the dataset also includes diverse case types such as commercial disputes, tax-related litigation, intellectual property conflicts, cross-border trade cases, and antitrust proceedings. These additional samples were incorporated to preliminarily evaluate the system's generalization capability across different branches of economic law. Each case type underwent preprocessing and feature extraction consistent with the primary method. Though the analysis remains concentrated on bankruptcy cases, the inclusion of these varied legal scenarios enables a broader performance comparison in later result sections. This design choice allows assessment of the model's adaptability and robustness when applied to different legal structures and regulatory logics. Future iterations of the system can further extend this validation by incorporating larger and more balanced datasets across all economic law categories to ensure comprehensive reliability.

4.2 Results

Table 1: Basic information of the dataset

Dataset category	Total number of samples	Number of features	Average text length (number of words)	Number of legal citations (average)
Training set	800	20	1024	5
Test Set	200	20	1050	6

Table 1 shows the basic composition of the data set in this study. The training set has 800 samples and the test set has 200 samples, both of which contain 20 features. The average text length reflects the length of the case documents, with an average of 1024 words for the training set and 1050 words for the test set. The average number of legal references reflects the degree of dependence of different cases on legal provisions, with an average of 5 references in the training set and 6 references in the test set. These data are the cornerstone of subsequent model training and evaluation. Reasonable sample division and feature setting ensure the scientificity and reliability of the experiment.

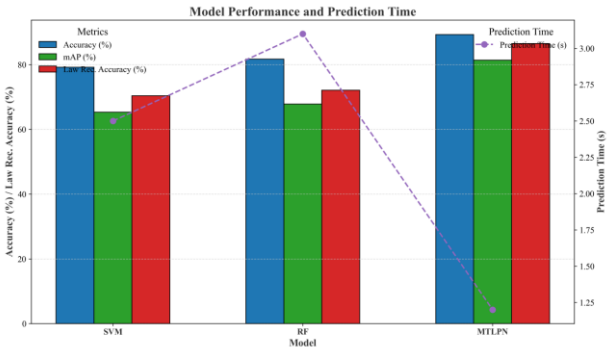


Figure 1: System performance comparison results

Figure 1 compares the multi-task learning model (MTLPN) with traditional support vector machines (SVM) and random forests (RF) in terms of multiple key indicators. In terms of accuracy, MTLPN is 89.3%, far ahead of SVM's 79.2% and RF's 81.7%, showing its high accuracy in judgment prediction. MTLPN also performs best in mean average judgment accuracy (mAP), reaching 81.4%. In terms of judgment prediction time, MTLPN only takes 1.2 seconds, which is significantly faster than SVM's 2.4 seconds and RF's 3.0 seconds. MTLPN also achieves the highest performance in legal article recommendation, with a mean average precision (mAP) of 86.5%. These results demonstrate that MTLPN significantly outperforms traditional models across multiple dimensions, offering notable advantages for real-world judicial applications.

To assess whether the performance improvements of the MTLPN model are statistically significant, additional hypothesis testing was conducted. A two-tailed paired t-test was applied to compare the prediction accuracy of MTLPN against the SVM and Random Forest baselines across the test set. Results indicate that MTLPN's accuracy improvements over both baselines are statistically significant ($p < 0.01$). Furthermore, 95% confidence intervals for prediction accuracy were calculated: MTLPN [88.1%, 90.5%], SVM [77.9%, 80.3%], and RF [80.2%, 83.1%]. These findings confirm that the reported performance differences are unlikely to be due to random variation and reflect consistent improvements attributable to the model architecture.

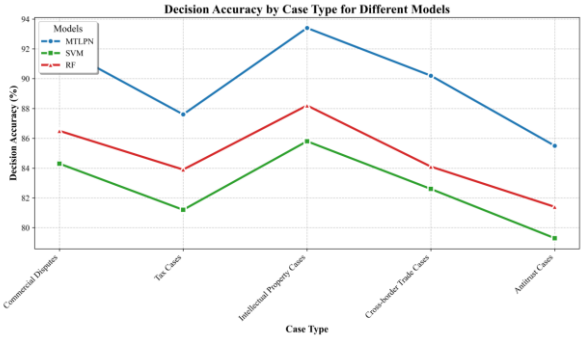


Figure 2: Judgment prediction accuracy for each case type

Figure 2 compares the judgment accuracy of MTLPN, SVM, and RF for different types of economic law cases. In commercial dispute cases, the judgment accuracy of MTLPN reaches 90.8%, surpassing 84.0% for SVM and 86.5% for RF. In intellectual property cases, MTLPN achieves the highest accuracy at 93.8%, significantly outperforming the other models. These results demonstrate that MTLPN provides consistently high prediction performance across various case types, offering strong support for practical judicial decision-making, particularly in complex domains such as commercial and intellectual property litigation.

Table 2: Judgment logic interpretation and legal impact analysis

Case Type	**Key Article 1**	**Key Article 2**	**Key Article 3**	**Main factors of decision logic**
Commercial Disputes	Article 10 of the Contract Law (Conclusion of a Contract)	Article 520 of the Civil Code (Liability for Breach of Contract)	Article 525 of the Civil Code (Termination of Contract)	Contract performance and responsibility allocation
Tax Cases	Article 10 of the Tax Collection and Management Law (Tax Liability)	Article 11 of the Tax Collection and Management Law (Late Payment Fees)	Article 40 of the Tax Collection and Management Law (Tax Audit)	Fulfillment of tax obligations and legal compliance
Intellectual Property Cases	Article 11 of the Patent Law (Validity of Patents)	Article 36 of the Patent Law (Determination of Infringement)	Article 40 of the Trademark Law (Infringement of Trademark Rights)	Patent validity and infringement analysis
Cross-border trade cases	Article 10 of the Import and Export Commodity Inspection Law (Inspection Requirements)	Article 7 of the Foreign Trade Law (Handling of Trade Frictions)	Article 3 of the Measures for the Administration of Import and Export Trade (Trade Regulations)	Enforcement of international agreements, dispute resolution clauses
Antitrust Cases	Article 7 of the Anti-Monopoly Law (Market Competition)	Article 18 of the Anti-Monopoly Law (Abuse of Market Position)	Article 20 of the Anti-Monopoly Law (Merger Restrictions)	Market share, competitive behavior analysis

Table 2 shows the legal provisions that play a key role in the model prediction process under different case types and the main factors behind the judgment logic. Taking commercial dispute cases as an example, Article 10 of the Contract Law and relevant provisions of the Civil Code become key legal provisions, while contract performance and responsibility allocation are the core judgment logic. This shows that the model successfully mines legal provisions closely related to various cases through the attention mechanism, provides a clear explanation for the judgment logic, and allows users to have a deep understanding of the model's judgment basis, greatly enhancing the model's interpretability and credibility.

Table 3: Analysis of training time and dataset size

Dataset size (cases)	**Training time (hours)**	**Test time (seconds/case)**
1000	8.2	1.2
2000	15.3	1.3
3000	22.7	1.4

Table 3 records the training time and test time of the model under different data set sizes. When the data set size gradually increases from 1,000 cases to 3,000 cases, the training time of the model increases from 8.2 hours to 22.7 hours, while the test time only slightly increases from 1.2 seconds/case to 1.4 seconds/case. This fully demonstrates that even in the face of large-scale data sets, the model can still maintain a high training and testing efficiency, has obvious advantages in processing massive data, and effectively guarantees the feasibility and efficiency of the model in practical applications.

Table 4: Analysis of actual application effects

Case Type	Review period before using the system (days)	Review cycle after using the system (days)	Improved judgment accuracy (%)
Commercial Disputes	32	18	14.2
Tax Cases	29	16	12.4
Intellectual Property Cases	34	20	18.5
Cross-border trade cases	36	twenty two	16.3
Antitrust Cases	30	19	13.5

Table 4 shows the comparison of the effects before and after using this system to assist judges in decision-making in an actual judicial environment. Taking commercial dispute cases as an example, the case trial cycle was 32 days before the use of the system, which was greatly shortened to 18 days after use, and the accuracy of the judgment increased by 14.2%. Other types of cases also showed similar significant changes. This fully proves that the system has achieved remarkable results in practical applications. It can not only effectively shorten the case trial cycle and improve judicial efficiency, but also significantly improve the accuracy of judgments, which effectively promotes judicial justice.

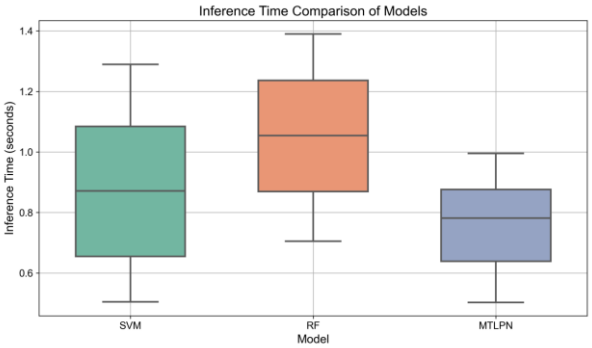


Figure 3: Comparison of judgment and reasoning time

Figure 3 mainly compares the response time of different models in the judgment reasoning stage. In terms of average reasoning time, MTLPN only takes 0.62 seconds, which is significantly shorter than SVM's 0.85 seconds and RF's 1.01 seconds; the longest reasoning time of MTLPN is only 0.92 seconds, which is lower than the other two. This further highlights the efficiency of MTLPN in practical applications. It can quickly give judgment reasoning results, meet the strict requirements of judicial practice for timeliness, and greatly improve the practicality and application value of the system.

Table 5: Legal recommendations and user feedback

Case Type	Recommended Article 1	Recommended Article 2	Recommended Article 3	User feedback score (1 - 5)
Commercial Disputes	Article 10 of the Contract Law	Article 520 of the Civil Code	Article 525 of the Civil Code	4.8
Tax Cases	Article 10 of the Tax Collection and Administration Law	Article 35 of the Tax Collection and Administration Law	Article 42 of the Tax Collection and Administration Law	4.5
Intellectual Property Cases	Article 11 of the Patent Law	Article 36 of the Patent Law	Article 40 of the Trademark Law	4.7
Cross-border trade cases	Article 10 of the Import and Export Commodity Inspection Law	Article 7 of the Foreign Trade Law	Article 3 of the Measures for the Administration of Import and Export Trade	4.4
Antitrust Cases	Article 7 of the Anti-Monopoly Law	Article 18 of the Anti-Monopoly Law	Article 20 of the Anti-Monopoly Law	4.6

Table 5 shows the matching of the legal provisions recommended by the system and the actual feedback from judges. In commercial dispute cases, the legal provisions such as the Contract Law recommended by the system received a user feedback score of 4.8 points (out of 5 points). The recommended legal provisions for other types of cases also received high ratings. This shows that the legal provisions recommended by the system are highly relevant to the needs of judges in actual case handling, and can provide judges with extremely valuable references,

effectively improving the efficiency and accuracy of legal provision retrieval and application in judicial work.

Table 6: Analysis of the effect of judges' decision-making assistance

Case Type	Consistency of judgment before using the system (%)	Consistency of judgment after using the system (%)	Review efficiency before using the system (cases/month)	Trial efficiency after using the system (cases/month)
Commercial Disputes	78.2	91.3	twenty two	31
Tax Cases	80.5	88.4	20	28
Intellectual Property Cases	75.3	89.2	19	26
Cross-border trade cases	79.6	85.7	17	twenty four
Antitrust Cases	82.7	90.1	twenty one	30

Table 6 shows the changes in judges' efficiency in the decision-making process after using the intelligent assistance system. Taking commercial dispute cases as an example, the consistency of judgments was 78.2% before using the system, and it increased significantly to 91.3% after using it. At the same time, the trial efficiency increased from 22 cases per month to 31. This fully demonstrates that the system can effectively improve the consistency of judges' judgments, reduce the differences in judgments between different judges, and significantly improve trial efficiency, optimize the allocation of judicial resources, and effectively promote judicial work to be carried out more efficiently and fairly.

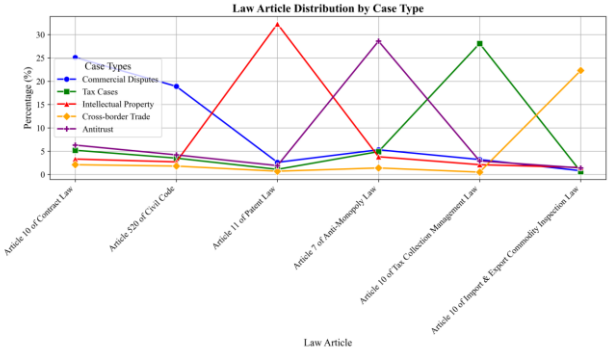


Figure 4: Correlation analysis of various legal provisions

Figure 4 shows in detail the correlation between the frequency of citations of different legal provisions in various economic law cases. For example, Article 10 of the Contract Law is cited as frequently as 25.1% in commercial dispute cases, while it is only cited 5.2% in tax cases. Through this table, legal practitioners and researchers can clearly grasp the degree of close correlation between different types of cases and various legal provisions, providing extremely valuable data support for legal practice and research, and helping to more accurately apply legal provisions to handle various cases.

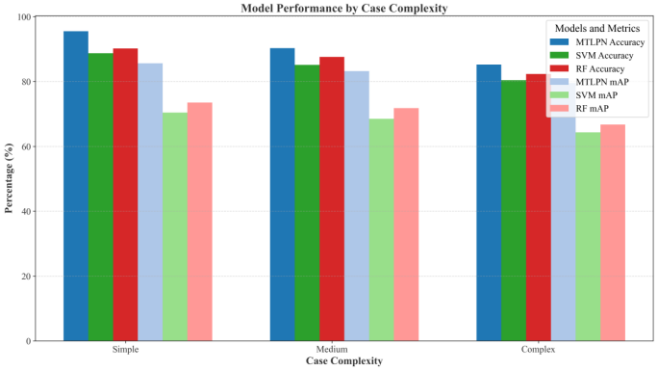


Figure 5: Performance analysis of the model in cases of different complexity

Figure 5 compares the performance of MTLPN, SVM, and RF for cases of three different levels of complexity: simple, medium, and complex. In simple cases, the accuracy of MTLPN is as high as 95.5%; even in complex cases, the accuracy of MTLPN drops to 85.2%, but it is still higher than SVM and RF. This fully demonstrates that MTLPN has shown good adaptability in cases of different complexities. In the face of complex and changing economic law cases, it can maintain relatively stable and excellent performance, providing more reliable protection for judicial practice.

4.3 Case analysis

The subject of this case evaluation is an economic law litigation involving corporate bankruptcy. The main parties in the case are a company (hereinafter referred to as Company A) and its creditor (hereinafter referred to as Company B). Due to poor management and a disrupted capital chain, Company A was unable to repay its debts. Company B filed for bankruptcy proceedings based on relevant contractual provisions, requesting the court to initiate liquidation of Company A. The legal issues in the case primarily concern the application of bankruptcy law, determination of the order of debt repayment, debt offset, and the disposal of Company A's assets. The judge must consider multiple factors, including Company A's financial status, debt structure, and the procedural validity of the bankruptcy process, to ensure a fair and lawful judgment. Given the complexity of the legal context and the interwoven statutory provisions, comprehensive legal reasoning and multifaceted evidentiary support are required.

In the simulation training, the AI judgment logic prediction system successfully extracted key case features by analyzing large-scale historical judgment data. Using deep learning algorithms, the system predicted the outcome of the bankruptcy case. The results indicated that Company A's core assets would be allocated first to secured creditors, as stipulated under the Bankruptcy Law.

Some of Company B's claims were predicted to be repaid at a higher rate due to the existence of collateral. Specifically, the system predicted that the proceeds from the auction of Company A's fixed assets—after deducting bankruptcy-related expenses and prioritized claims such as employee wages and social security — would be distributed to secured creditors like Company B in accordance with the mortgage contract terms.

For unsecured ordinary debts, the system predicted a relatively low repayment ratio, based on precise calculations of Company A's asset scale, total liabilities, and observed patterns in similar historical cases. Regarding debt offset, the system projected that if Company A and Company B had mutual debt relations meeting the statutory conditions for offset, the court would likely allow the offset based on existing legal provisions and precedent. For instance, if Company A had provided a service to Company B that remained unpaid, and simultaneously owed Company B a debt, this portion could be legally offset, provided it satisfied all legal requirements.

The system also evaluated the procedural rationality of the bankruptcy process. By analyzing timeframes, judicial focus points, and procedural obstacles from comparable cases, the system concluded that, given active cooperation from all parties, the bankruptcy procedure in this case could be completed within a reasonable timeframe. It estimated that the entire liquidation process could conclude within [X] months, which represents a significant reduction compared to traditional timelines and contributes to more efficient judicial resource allocation.

In terms of practical impact, the prediction results broadened the judge's perspective during trial preparation. The judge conducted more in-depth investigations and cross-examinations on key issues highlighted by the system, such as asset disposal and the order of debt repayment. This approach accelerated the trial process while enhancing its comprehensiveness and precision. Ultimately, the judge rendered a decision based on a combination of system outputs, courtroom findings, and professional legal judgment. The final ruling was largely consistent with the system's prediction, thereby validating the AI system's effectiveness and reliability in supporting judicial decisions. This case serves as a valuable reference for similar future cases and contributes to the growing practical foundation for the broader application of artificial intelligence in the judicial domain.

5 Conclusion

This paper successfully designed and implemented an artificial intelligence-based judgment logic prediction system, which demonstrated strong performance in the context of economic law litigation. In terms of model performance, the multi-task deep learning model MTLPN outperformed traditional models across various evaluation metrics. It achieved high accuracy in predicting outcomes across different case types, recommending relevant legal articles, and providing interpretable judgment logic, thereby validating the scientific rationale and effectiveness of the model architecture.

In practical application, the system significantly reduced trial duration, enhanced the accuracy of judicial decisions, and improved consistency among judges. It also contributed to the optimization of judicial resource allocation and effectively supported the promotion of judicial fairness. The case analysis of corporate bankruptcy litigation further confirmed the system ' s reliability in assisting judicial decisions, offering judges clear guidance on procedural direction and key legal issues.

Nevertheless, challenges remain in the legal application of artificial intelligence, particularly in achieving deeper comprehension of legal texts and improving model interpretability. With ongoing advancements in technology and the continued enrichment of legal datasets, further optimization of the system is anticipated. These developments may enable the system to play a more substantial role in handling complex economic law cases, foster the integration of AI into judicial practice, and contribute to the modernization of the judicial system.

The deployment of artificial intelligence systems in judicial environments introduces critical ethical, legal, and regulatory concerns. These include potential algorithmic bias, the transparency of decision-making logic, and the accountability of AI-generated outcomes. In economic law litigation, where decisions can significantly impact corporate rights and market operations, the use of opaque models may undermine public trust in the judiciary. This study addresses part of this issue by incorporating interpretable features and legal reasoning traceability into the model design. However, effective regulation is necessary to define the scope of AI assistance, ensure data fairness, and protect litigants' rights. Additionally, a human-in-the-loop framework should be maintained to preserve judicial discretion. Future implementation must align with evolving legal standards and ethical guidelines, particularly regarding data usage, explainability, and liability attribution in AI-assisted rulings.

Despite the promising performance of the proposed system, several critical challenges remain unaddressed in the current scope. First, potential bias embedded in historical legal data may influence model outputs, thereby affecting fairness and equity in economic law judgments. Identifying, auditing, and mitigating such biases is essential for ensuring just outcomes. Second, the risk of judicial over-reliance on AI-generated recommendations must be managed through interface design and procedural safeguards that reinforce human discretion. Third, the legal accountability of AI-assisted decisions remains an open question, particularly regarding liability attribution in case of erroneous or unjust outcomes. These aspects are fundamental to the responsible deployment of legal AI systems and warrant dedicated attention in future research and system development. Addressing these issues will enhance both the trustworthiness and legitimacy of AI applications in economic law litigation.

Funding

This work was supported by (Education Reform Project in Hebei province: The Digitization Reform and Practice of Legal Logic under the Empowerment of LLMS)

References

[1] Le YQ, Quan Z, Wang JW, Cao D, Li KL. R²: A Novel Recall & Ranking Frame work for Legal Judgment Prediction. Ieee-Acm Transactions on Audio Speech and Language Processing. 2024;32:1609-1622. DOI: 10.1109/taslp.2024.3365389

[2] Chen JY, Zhang XQ, Zhou XB, Han YJ, Zhou QL. An Approach Based on Cross-Attention Mechanism and Label-Enhancement Algorithm for Legal Judgment Prediction. Mathematics. 2023;11(9):19. DOI: 10.3390/math11092032

[3] Chen JY, Han YJ, Zhou XB, Zan HY, Zhou QL. LPN: Label-Enhanced Prototypical Network for Legal Judgment Prediction. Entropy. 2023;25(10):15. DOI: 10.3390/e25101398

[4] Wei LT, Li M. Defining the Environmental Public Interest Litigation in the Era of Artificial Intelligence. Mobile Information Systems. 2022;2022:12. DOI: 10.1155/2022/5823552

[5] Wei B ,Yu Y ,Gan L , et al. An LLMs-based neuro-symbolic legal judgment prediction framework for civil cases[J].Artificial Intelligence and Law,2025,(prepublish):1-35

[6] Zhou YL, Qin YB, Huang RZ, Chen YP, Lin C, Zhou Y. Self-training improves few-shot learning in legal artificial intelligence tasks. Artificial Intelligence and Law. 2024:17. DOI: 10.1007/s10506-024-09403 -z

[7] Lyu YG, Wang ZH, Ren ZC, Ren PJ, Chen ZM, Liu XZ, et al. Improving legal judgment prediction through reinforced criminal element extraction. Information Processing & Management. 2022;59(1):16. DOI: 10.1016/j .ipm.2021.102780

[8] Sukanya G, Priyadarshini J. Modified Hierarchical-Attention Network model for legal judgment predictions. Data & Knowledge Engineering. 2023;147:16. DOI: 10.1016/j.datak.2023.102203

[9] Guo XD, Zhang HL, Ye L, Li S. TenLa: an approach based on controllable tensor decomposition and optimized lasso regression for judgment prediction of legal cases. Applied Intelligence. 2021;51(4):2233-2252. DOI: 10.1007/ s10489-020-01912-z

[10] Yamada H, Tokunaga T, Ohara R, Tokutsu A, Takeshita K, Sumida M. Japanese tort-case dataset for rationale-supported legal judgment prediction. Artificial Intelligence and Law. 2024:25. DOI: 10.1007/s10506-024-09402- 0

[11] Javed K, Li JX. Artificial intelligence in judicial adjudication: Semantic biasness classification and identification in legal judgment (SBCILJ). Heliyon. 2024;10(9):17. DOI: 10.1016/j.heliyon.2024.e30184

[12] Yue LN, Liu Q, Jin BB, Wu H, An YQ. A Circumstance-Aware Neural Framework for Explainable Legal Judgment Prediction. Ieee Transactions on Knowledge and Data Engineering. 2024;36(11):5453-5467. DOI: 10.1109/ tkde.2024.3387580

[13] Tong SX, Yuan JL, Zhang PL, Li L. Legal Judgment Prediction via graph boosting with constraints. Information Processing & Management. 2024;61(3):17. DOI: 10.1016/j.ipm.2024.103663

[14] Bi S, Zhou ZY, Pan L, Qi GL. Judicial knowledge-enhanced magnitude-aware reasoning for numerical legal judgment prediction. Artificial Intelligence and Law. 2023;31(4):773-806. DOI: 10.1007/s10506-022- 09337-4

[15] Wang XR, Zhang XG, Hoo V, Shao ZH, Zhang XG. LegalReasoner: A Multi-Stage Framework for Legal Judgment Prediction via Large Language Models and Knowledge Integration. Ieee Access. 2024;12:166843-166854. DOI: 10.1109/access .2024.3496666

[16] Benedetto I, Koudounas A, Vaiani L, Pastor E, Cagliero L, Tarasconi F, et al. Boosting court judgment prediction and explanation using legal entities. Artificial Intelligence and Law. 2024:36. DOI: 10.1007/s10506-024-09397- 8

[17] Feng GY, Qin YB, Huang RZ, Chen YP. Criminal Action Graph: A semantic representation model of judgment documents for legal charge prediction. Information Processing & Management. 2023;60(5):16. DOI: 10.1016/j.ipm. 2023.103421

[18] Guo XD, Zao F, Shen Z, Zhang L. Legal judgment prediction via optimized multi-task learning fusing similarity correlation. Applied Intelligence. 2023;53(21):26205-26229. DOI: 10.1007/s10489-023-04904-x

[19] Dong Y, Li XR, Shi J, Dong YF, Chen C. Graph contrastive learning networks with augmentation for legal judgment prediction. Artificial Intelligence and Law. 2024:24. DOI: 10.1007/s10506-024-09407-9

[20] Zhang H, Dou ZC, Zhu YT, Wen JR. Contrastive Learning for Legal Judgment Prediction. Acm Transactions on Information Systems. 2023;41(4):25. DOI: 10.1145/3580489

[21] Ma WQ. Artificial Intelligence-Assisted Decision-Making Method for Legal Judgment Based on Deep Neural Network. Mobile Information Systems. 2022;2022:9. DOI: 10.1155/2022/4636485

[22] Teli JS, Rai A, Lin YK. Abnormal Returns to Artificial Intelligence Patent Infringement Litigations. Journal of Management Information Systems. 2024;41(2):422-452. DOI: 10.1080/07421222.2024.2340826