Situational Awareness and Fault Warning for Smart Grids Combined with Deep Learning Technology: Application of Digital Twin Technology and Long Short Term Memory Networks

Yanjie Zhang*, Zhihui Kang
Hebi Institute of Engineering and Technology, Henan Polytechnic University, Hebi 458000, China
Email: zhangyanjie2323@126.com
*Corresponding author

Keywords: long short-term memory network, smart grid, deep learning, situational awareness, fault warning, digital twin technology

Received: January 8, 2025

In order to achieve effective perception of the power grid situation and accurate warning of operational faults, this study proposes a situation perception and fault warning method for smart grids based on deep learning technology. Firstly, using the digital twin smart grid platform as a carrier, build a smart grid digital twin situational awareness framework; Secondly, considering both dynamic and static security, intelligent grid situation evaluation indicators are selected; Then, comprehensively analyze the data of various indicators, evaluate the security situation of the power grid, and calculate the security situation assessment value of the power grid; Finally, a smart grid situational awareness model is built based on long short-term memory networks to achieve smart grid situational awareness and fault warning. A provincial-level smart grid big data information platform conducted experiments as the data source. After dividing the training and testing samples, 1000 iterations of learning were carried out to complete situational awareness and fault warning. The experiment was conducted to verify the accuracy, recall, F1 score, fault warning accuracy, fault command response time, and resource consumption of safety situation prediction results and actual values, as well as safety situation discrimination results. The experimental results show that the accuracy of this method for identifying the safety situation of smart grid operation is 98.72%, the recall rate is 98.95%, and the F1 score is 99.06%. This indicates that the comprehensive application performance of this method is good, and it can accurately and effectively perceive, predict, and analyze the safety situation of smart grid operation. At the same time, the maximum fault warning accuracy of this method is 99.82%, the minimum fault command response time is 0.083 s, and the minimum resource consumption is 118.57 MB, indicating that this method has a good power grid fault warning effect, which can accurately distinguish between normal operating conditions and critical states before faults and provide real-time and effective warnings.

Povzetek: Raziskava predstavi metodo za zaznavanje situacijskega zavedanja in napovedovanje napak v pametnih omrežjih, ki temelji na globokem učenju z uporabo omrežij dolgoročne kratkoročne pomnilnosti (LSTM) in digitalne tehnologije dvojčkov.

1 Introduction

The smart grid, known as the new era of “Grid 2.0”, is rooted in the solid foundation of integrated and high-speed bidirectional communication networks [1]. With cutting-edge sensing and measurement technology, precision equipment technology, advanced control strategies, and the comprehensive application of intelligent decision support systems, it is committed to achieving the reliability, safety, economy, high efficiency, environmental harmony, and worry free safety of power supply for users [2]. With the rapid development of the power industry, China has entered an era of “ultra-high voltage, large power grid”. However, the structure of the smart grid is relatively weak, and the failure rate of electrical equipment and lines is high. It has also experienced multiple large-scale power outages [3]. Therefore, it is necessary to timely and effectively prevent power outages in the power grid, predict the safe operation status of the smart grid, and perceive the safety situation of the smart grid.

Presekal et al. [4] proposes a hybrid deep learning model based on the perspective of smart grid network security situational awareness to achieve online network attack situational awareness. By combining deep convolutional neural networks to construct a basic perception network framework, a temporal data classification unit is constructed in the network architecture to detect anomalies in the input power grid situation data. However, this method has the problem of slow overall response speed to safety faults. Bai et al. [5] extensively explores effective security situational awareness methods and remote operation and maintenance technologies to enhance the overall defense capability of smart grid systems, ensure power supply continuity and reliability. Constructing a neural network model using radial basis functions to comprehensively process operational data of the power grid system. On this basis, linear discriminant analysis was introduced into the model to establish an efficient power grid anomaly situation detection model, effectively realizing the perception of smart grid operation trends. However, this method has certain room for improvement in the division of power grid operation risk thresholds. Gong et al. [6] proposes a network security situational awareness detection technology based on distributed data analysis, taking into account the characteristics of big data in intelligent power networks. By applying cross entropy function and linear units, the loss evaluation part of the neural network model was optimized, and an innovative smart grid operation situation awareness model was constructed by integrating improved linear unit structure. However, this method has the problem of low utilization of computational resources. Zhai et al. [7], an iterative algorithm that integrates Gaussian processes was designed to use the time series measured by the phasor measurement units of the actual power grid to verify the trend indicators of power grid operation online, in order to evaluate the stability level of smart grid operation. However, the overall safety situation awareness accuracy of this method needs to be improved.

Long Short Term Memory (LSTM), as a special variant of recurrent neural networks, is an efficient deep learning technique with strong sequential data processing capabilities, suitable for processing time-series data and predicting future situations. Due to the large and complex amount of data involved in the smart grid, including multiple dimensions and variables, many important states and changes may accumulate over time and affect future trends. Based on the above analysis, this study combines deep learning technology to propose a smart grid situational awareness method based on LSTM, and further designs a smart grid fault warning method with the aim of reducing the impact of operational faults.

2 Design of smart grid situation awareness and fault warning methods

2.1 Construction of smart grid digital twin situation awareness framework

Smart grid digital twin refers to the complete mapping of the physical entities of the smart grid in the digital world based on digital twin technology, forming a digital model that is synchronized and consistent with the real grid. This digital model can include all information about the equipment, lines, operating status, environmental factors, etc. of the power grid, achieving real-time monitoring and prediction of the power grid status.

The digital twin power grid essentially belongs to the form of a physical power grid coexisting with a virtual power grid in the information dimension, and the integration of virtual and real power grids [8, 9]. Therefore, the collection of smart grid situation indicator data can be based on the digital twin grid. On the basis of smart grid IoT data perception and multi-dimensional information transmission, real-time holographic simulation can be carried out through the digital Lisheng platform to make scientific decisions and intelligent control processes, and to achieve real-time prediction and analysis of the operation situation of the physical grid. The smart grid digital twin situational awareness framework is shown in Figure 1.

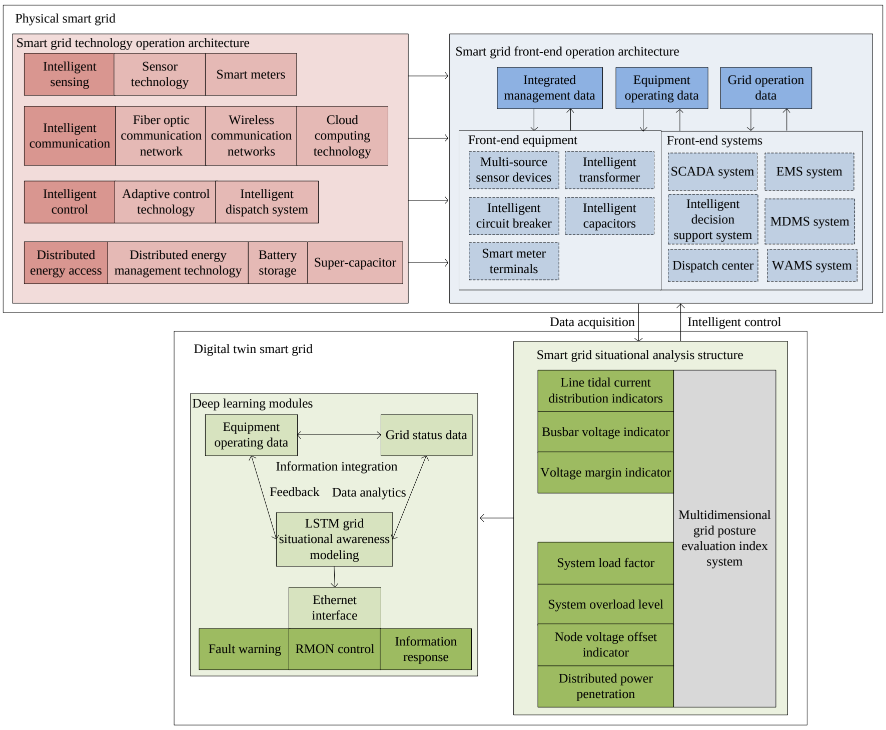

Physical smart grid
Smart grid technology operation architecture
Intelligent sensing
Sensor technology
Smart meters
Intelligent communication
Fiber optic communication network
Wireless communication networks
Cloud computing technology
Intelligent control
Adaptive control technology
Intelligent dispatch system
Distributed energy access
Distributed energy management technology
Battery storage
Super-capacitor
Smart grid front-end operation architecture
Integrated management data
Equipment operating data
Grid operation data
Front-end equipment
Multi-source sensor devices
Intelligent transformer
Intelligent circuit breaker
Intelligent capacitors
Smart meter terminals
Front-end systems
SCADA system
EMS system
Intelligent decision support system
MDMS system
Dispatch center
WAMS system
Data acquisition
Intelligent control
Digital twin smart grid
Deep learning modules
Equipment operating data
Grid status data
Information integration
Feedback
Data analytics
LSTM grid situational awareness modeling
Ethernet interface
Fault warning
RMON control
Information response
Smart grid situational analysis structure
Line tidal current distribution indicators
Busbar voltage indicator
Voltage margin indicator
System load factor
System overload level
Node voltage offset indicator
Distributed power penetration
Multidimensional grid posture evaluation index system

Figure 1: The digital twin situational awareness framework of smart grid

The smart grid digital twin situational awareness framework covers multiple elements. At the physical entity level of the smart grid, real-time collection of operational and management data and other status information of the smart grid and its internal devices, such as smart meter data and device operating parameters, is achieved through technology entities such as sensors and wireless communication. This information is transmitted to the digital twin through a digital twin link. The digital twin initially processes the received data and constructs a multidimensional power grid situation evaluation index system [10], including indicators such as line flow distribution and bus voltage. Then, based on this, the situation evaluation index data is filtered and generated, and transmitted to the deep learning unit through the digital twin data platform. The operational architecture of smart grid technology in the framework includes intelligent sensing, communication, control, and distributed energy access; The front-end operation architecture includes comprehensive management, equipment and power grid operation data related content, as well as various front-end devices and systems. The deep learning module uses LSTM situational awareness modeling and other methods to analyze data based on power grid status data, achieving intelligent power grid situational analysis functions. It also has response mechanisms such as fault warnings.

2.2 Selection of evaluation indicators for smart grid situation

Situational awareness refers to the recognition and understanding of environmental factors within a certain time and space range, and the prediction of future development trends. The situational awareness of smart grid is an important technical means to grasp the operation trajectory of the power grid. In the perception practice, it is necessary to first monitor and extract various factors related to the changes in power grid operation, in order to characterize the operation trajectory of the power grid.

However, a single indicator cannot effectively characterize the operational trajectory of the smart grid. Therefore, based on the analysis structure of the smart grid situation using the correlation indicators in Figure 1, this article considers both dynamic and static security aspects of the smart grid, and sets the line flow distribution index P_1', bus voltage index P_2', and active power margin P_3' as the dynamic security situation evaluation indicators for the smart grid, which can reflect the operating status of the smart grid; The system load rate Q_1', system overload degree Q_2', node voltage offset index Q_3', and distributed power penetration rate Q_4' are static security situation assessment indicators, which perceive and analyze the situation of the smart grid. Based on the evaluation indicators and related parameters, the data collection scope is clarified, and the corresponding collection indicator data is collected to comprehensively and effectively evaluate the operation trajectory of the smart grid.

Among them, the power flow distribution index of the line refers to the average difference between the maximum transmission capacity allowed by the line in the system and the active power flow of the line. This indicator reflects the stability of the system. The larger the indicator value, the farther the system is from the allowed maximum transmission capacity and the more stable the system is. The expression for this indicator and its associated parameters is Equations (1) and (2):

$$P_1' = V_z' \times \left(I_{za}'\right)^{\overline{Z}} \tag{1}$$

$$V_z' = \frac{I_{za} \times \cos\left(\hat{\theta}_z - \varphi_1\right)}{p_{za}'} \tag{2}$$

In the formula: V_z' represents the phasor of the bus phase voltage; za represents the branch bus in the smart grid, and I_{za}' represents the parallel double bus structure, which is the phasor of the line current generated during the load transfer process of the double bus; $\overline{Z}$ represents conjugate complex numbers; p_{za}' represents the active power of the three-phase AC line in the smart grid; $\hat{\theta}_z$ represents the voltage phase angle; φ_1 represents the phase angle of the current.

The amplitude of bus voltage refers to the average value of the voltage of the bus (excluding the bus connected to the generator) in the system. This indicator reflects the ability of the system bus to withstand voltage. The larger the indicator value, the stronger the system's ability to withstand voltage [11, 12]. Based on the known power flow distribution of the line, calculate the bus voltage amplitude index and related parameters according to the active power of the line where the parallel bus is located Equation (3):

$$P_2' = X'_{z,\gamma_1}\left(V_z' + V_a'\right)^2 + 2\left[R_{za}'\left(p_{za}' + q_{za}'\right)\right] \tag{3}$$

In the formula: X'_{z,γ_1} represents the reactance between the distributed generation unit of the smart grid and the busbar; γ_1 represents the power generation unit, which is a synchronous generator; V_a' represents the phase voltage phasor of the branch bus in the double bus structure; R_{za}' represents the resistance of the three-phase AC line in the smart grid; q_{za}' represents the reactive power of the three-phase AC line in the smart grid [13].

The active power margin refers to the average ratio of the difference between the maximum transmission capacity of the line in the system and the active power flow of the line in the current state. This indicator reflects the system's ability to withstand power disturbances, and the larger the indicator value, the stronger the system's ability to withstand power disturbances [14]. Given the layout of distributed generation units in the three-phase AC line of a smart grid, and based on clarifying the reactance parameters between the distributed generation units and the bus, calculate the active power margin of the smart grid system, as shown in Equations (4) and (5):

$$P_3' = P_2'\left(\lambda_1 - \lambda_2\right) \tag{4}$$

$$\lambda_2 = \sum_{m'=1}^{2} \sin \delta' \times \hat{E}\left(V_z' + V_a'\right)^{m'} - X'_{z,\gamma_1} \tag{5}$$

In the formula: λ_1 and λ_2 represent the total active power generation capacity and total active load demand of the smart grid system; m' represents the total layout of power generation units in the power grid system; δ' represents the power angle of the power generation unit, which is the phase difference between the excitation potential and the terminal voltage of the generator; $\hat{E}$ represents the electromotive force of the generator.

Based on the static security of the smart grid, the system load rate refers to the ratio of the sum of the transmission power of the system lines to the maximum transmission capacity allowed by the lines. This indicator reflects the probability of a major power outage in the system, and the higher the value of this indicator, the greater the probability of a major power outage occurring in the system [15]. The degree of system overload refers to the ratio of the number of overloaded lines to the total number of remaining lines when a component of the system fails. This indicator represents the degree of overload caused by system component failures. The larger the value of this indicator, the more lines the system deviates from normal state, and the greater the degree of overload of the system, making its state more dangerous. The expression for this indicator and its associated parameters is Equations (6) and (7):

$$Q_1' = \lambda_2 - \left\| C_{i'}' - \sqrt[\lambda_1]{\vec{p}_1} \right\| \tag{6}$$

$$Q_2' = \frac{n_2}{n_1\left(n_1 + 1\right)} \tag{7}$$

In the formula: $C_{i'}'$ represents the required capacity of the generator; n_1 and n_2 represents the total number of lines in the smart grid system and the total number of overloaded lines in the system; $\vec{p}_1$ represents the total load power of the power grid.

The node voltage offset index Q_3' refers to the sum of the difference between the node voltage of the current system and the node voltage under normal conditions. This indicator reflects the volatility of the system voltage. The larger the value of this indicator, the greater the deviation of the system voltage from the normal voltage, and the more dangerous the system is. Considering the diversity and nonlinear characteristics of this indicator, only its characterization features will be analyzed here.

In addition, the penetration rate of distributed power refers to an indicator that measures the proportion of distributed power in the smart grid, reflecting the scale of distributed power relative to the total load of the grid, and quantitatively reflecting the degree of penetration of distributed power in the entire smart grid system. As the penetration rate increases, the impact of fluctuations in distributed power sources on grid frequency will gradually increase. At this time, new frequency regulation strategies (such as the coordination of energy storage systems) are needed to ensure grid frequency stability. This indicator also verifies the matching degree between the connected distributed power sources and local loads. The expression for this indicator and its associated parameters is Equation (8):

$$Q_4' = \frac{\sum_{j=1}^{\tilde{n}} \vec{p}_j\left(\tilde{n} + 1\right)}{\vec{p}_1} \tag{8}$$

In the formula: $\tilde{n}$ represents the number of distributed power sources in the smart grid; j represents the index of the power supply unit; $\vec{p}_j$ represents the output power of a fixed sequence distributed power source.

2.3 Smart grid situation awareness and fault warning based on long short term memory networks

2.3.1 Smart grid situation assessment

By comprehensively analyzing various indicator data, evaluate the safety situation of the power grid and calculate the safety situation assessment value of the power grid. Firstly, in order to improve the convenience of indicator processing and eliminate errors in indicators, dynamic and static security indicators are regarded as an analytical subject, and each indicator is normalized Equation (9):

$$L = \left(k_o - \min\, k_o\right)\left(\max k_o - \min\, k_o\right)^{-1} \tag{9}$$

In the formula: L represents the comprehensive indicator of the power grid situation after the unified state analysis subject; k_o represents the o-th indicator value of the smart grid system.

Secondly, based on the Analytic Hierarchy Process, determine the weight coefficients of the corresponding indicator values in Equation (9). By repeatedly determining the weight coefficients of multiple indicators and multiplying the comprehensive indicator data with the corresponding weights, the smart grid security situation assessment value is calculated Equation (10).

$$\hat{\kappa} = \sum_{o=1}^{7} \varpi_o \times L \tag{10}$$

In the formula: $\hat{\kappa}$ represents the evaluation value of the security situation of the smart grid; ϖ_o represents the weight coefficient corresponding to specific indicator data.

According to the safety standards for the operation of smart grids, further refine the risk categories corresponding to the security situation assessment values of smart grids, and set reasonable warning thresholds for the comprehensive indicators of security situation. The threshold and risk level classification of smart grid security situation warning is shown in Table 1.

Table 1: Threshold and risk level of smart grid security situation warning

Indicator warning thresholds	Risk class	Description of the type of situational risk
0-0.2	Safety status	——
0.21-0.5	Early warning status (low risk)	There are small fluctuations in the power of the distributed power sources, but the power supply is stabilizing the transmission efficiency of some lines slightly below optimal.
0.51-0.8	Dangerous state (medium risk)	Risk of overloading of transformers, small deviations from the normal range of voltage in some areas, abnormal intermittent changes in distributed power.
0.81-1.0	Emergency status (high risk)	Multiple key devices are close to or have reached their limit operating conditions, localized power outages, the grid control system is unable to accurately obtain information on the status of the devices, and there is a power deficit in the grid.

The division of the warning threshold for smart grid security situation in Table 1 above is based on a deep understanding of the operating characteristics of the power grid, statistical analysis of actual operating data, and comprehensive consideration of power grid security standards. Taking 0-0.2 as an example to represent the safety status, selecting 0.2 as the upper limit is based on statistical analysis of historical operating data and comprehensive consideration of power grid safety standards. This value ensures that the power grid can maintain safe and stable operation in most cases. If the threshold is set too loosely, it may reduce the sensitivity and accuracy of the warning system, thereby increasing the risk of power grid operation.

2.3.2 Analysis of long short term memory network structure

Long Short Term Memory (LSTM), as a special variant of recurrent neural networks, is a deep learning technique with strong sequential data processing capabilities, suitable for processing time-series data and predicting future situations [16, 17]. LSTM network introduces a unique gate structure (input gate, forget gate, and output gate) based on traditional recurrent neural networks, and then captures and models long-term dependencies in time series data through memory units and gate structures [18-21].

The integration of digital twin technology and LSTM significantly enhances the situational awareness and fault warning capabilities of smart grids through dynamic modeling and real-time updates. The digital twin technology constructs a virtual model of the power grid that can reflect the operating status in real time, while LSTM, as a time series model, excels at capturing long-term dependencies and complex nonlinear patterns in power grid data. This combination not only improves the accuracy of fault prediction, but also reduces false alarm rates, while enhancing adaptability, allowing it to dynamically adjust according to the real-time status of the power grid. Compared to traditional methods, LSTM performs better in processing time series data and can more effectively identify potential faults.

As for why other models (such as GRU or transformer models) are not considered, it is mainly due to their limitations in applicability and efficiency in smart grid scenarios. Although GRU is a simplified version of LSTM, its modeling ability is not as good as LSTM when dealing with complex time series data, especially in capturing long-term dependencies. Although the transformer model performs well in certain tasks, its computational complexity is high and it requires large-scale data for training, making it difficult to meet the real-time processing requirements of smart grids. In addition, LSTM has been widely applied in time series tasks, and its performance and stability have been fully verified. However, GRU and transformer models have relatively few applications in the field of smart grids, lacking sufficient practical support. Therefore, LSTM has become the preferred model in this scenario.

The data involved in the smart grid is massive and complex, containing multiple dimensions and variables, and many important states and changes may accumulate over time and affect future trends. LSTM, as a special variant of recurrent neural networks, has strong sequential data processing capabilities and is suitable for processing time-series data and predicting future situations. By introducing unique gate structures (input gate, forget gate, and output gate), LSTM can capture and model long-term dependencies in time series data, effectively improving the accuracy of situation prediction. In addition, the smart grid digital twin situational awareness framework constructed with digital twin technology can comprehensively and real-time monitor the status of the power grid, providing accurate and comprehensive input data for LSTM, further improving the accuracy of situational awareness and fault warning. Therefore, the article selects it as the main carrier for power grid situational awareness prediction, which is based on LSTM network for smart grid situational awareness. LSTM network structure is shown in Figure 2.

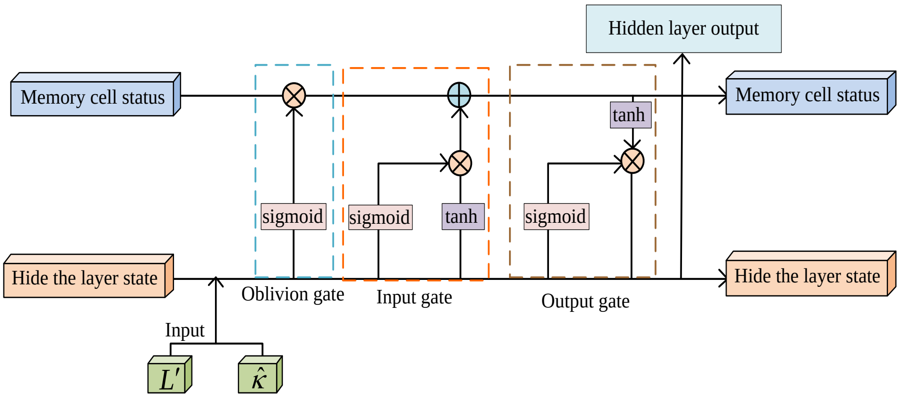


Figure 2: LSTM network architecture

In this study, the optimizer used in the LSTM network is the Adam optimizer, which can adaptively adjust the learning rate, accelerate convergence, and improve training efficiency. Set the learning rate to 0.5 to reduce overfitting. Set the batch size to 64 to balance memory usage and training speed. To better balance model complexity and learning ability, two hidden layers were chosen.

As shown in Figure 2, the basic unit of LSTM network includes three gates, namely input gate, forget gate, and output gate. Among them, the forget gate determines which information will be forgotten from the unit state, and its calculation Equation (11):

$$\hat{F}_1 = \sigma_1\left[\vec{\varpi}_1\left(y_{T-1}, L\right)\left(y_{T-1}, \hat{\kappa}\right) + b_1\right] \qquad (11)$$

In the formula: $\hat{F}_1$ represents the forget gate output at a specific time step; σ_1 represents the sigmoid activation function; $\vec{\varpi}_1$ represents the forgetting gate weight matrix; y_{T-1} represents the hidden state of the previous time step; T represents the current time step; b_1 represents the bias coefficient of the forget gate.

The input gate consists of two parts: the sigmoid layer and the tanh layer. The sigmoid layer determines which values will be updated, while the tanh layer creates a new candidate value vector. The calculation is Equation (12):

$$\begin{cases} \hat{F}_2 = \sigma_1\left[\vec{\varpi}_2\left(y_{T-1}, L\right)\left(y_{T-1}, \hat{\kappa}\right) + b_2\right] \\ \vec{H}_0 = \sigma_2\left[\vec{\varpi}_0\left(y_{T-1}, L\right)\left(y_{T-1}, \hat{\kappa}\right) + b_0\right] \end{cases} \qquad (12)$$

In the formula: $\hat{F}_2$ represents the input gate output at a specific time step; $\vec{\varpi}_2$ represents the weight matrix of the input gate; b_2 represents the bias coefficient of the input gate; $\vec{H}_0$ represents the candidate unit state at a specific time step; σ_2 represents the tanh activation function; $\vec{\varpi}_0$ represents the weight matrix of candidate unit states; b_0 represents the bias coefficient of the candidate unit state.

The output gate determines the output of the next hidden state, which is specifically represented as Equation (13):

$$\hat{F}_3 = \sigma_1\left[\vec{\varpi}_3\left(y_{T-1}, L\right)\left(y_{T-1}, \hat{\kappa}\right) + b_3\right] \qquad (13)$$

In the formula: $\hat{F}_3$ represents the output gate output at a specific time step; $\vec{\varpi}_3$ represents the output gate weight matrix; b_3 represents the bias coefficient of the output gate.

2.3.3 Implementation of smart grid situation awareness and fault warning

On the basis of laying out the LSTM model, it is necessary to train the network model and combine the trained model with the dynamic and static characteristics of the smart grid situation, and deploy it to the smart grid system [22, 23]. The specific model training steps are as follows:

Step 1: Input data partitioning. Using the normalized index data from the previous cycle (previous time step) as input and training data for LSTM, these normalized index data cover multiple dimensions of dynamic and static security of smart grids. Subsequently, these data are scientifically divided into training and testing sets. Typically, the training set is used for model training and learning, while the testing set is used for model performance validation, ensuring that the model can generalize to unseen data.

Step 2: Build an LSTM network. Based on the characteristics of the smart grid situation indicator data, the number of input layer feature types in the LSTM network structure is set to 7, and the number of hidden layers is set to 2, in order to balance the complexity and learning ability of the model. This network architecture design aims to efficiently extract key information from input data, laying a solid foundation for subsequent security situation assessment and fault warning.

Step 3: Initialize network parameters. After the LSTM network is built, the weights and biases in the network are randomly initialized to ensure that the model has sufficient diversity at the beginning of training, so as to gradually converge to the optimal solution in the subsequent learning process. This study sets the weight to 0.5 and the bias to 0.1.

Step 4: Model training. Using the training set data for forward propagation, calculate the loss function to measure the difference between the current model's predicted results and the true values. Subsequently, the weights and biases in the network are updated using backpropagation algorithm, gradually reducing the value of the loss function. This process is repeated in multiple iterations until the performance of the model on the training set reaches stability.

Step 5: Performance evaluation. Evaluate the performance of the trained LSTM model through a test set. The evaluation indicators include accuracy, recall, and F1 score, which can comprehensively reflect the model's ability in safety situation assessment and fault warning. Based on the evaluation results, continuously adjust the network structure and hyperparameters (such as learning rate, number of hidden units, etc.) to optimize the model performance. This process may require multiple iterations until the model performance reaches the predetermined accuracy and reliability standards [24-26].

Step 6: After the LSTM network model completes training and meets the predetermined performance standards, deploy it to the smart grid system. The model can receive real-time operation data of the smart grid and output evaluation scores from multiple dimensions including power flow distribution of grid lines, system load, and power penetration rate. These evaluation scores constitute the security situation assessment values of the smart grid, which can be compared with the preset alarm threshold to determine whether the operating situation of the smart grid is in a fault abnormal state.

Step 7: Based on the alarm threshold, corresponding level, and output evaluation value in Table 1, compare them to determine whether the operation status of the smart grid is in a fault abnormal state, and achieve smart grid situation perception and fault warning.

3 Experiments and results analysis

3.1 Experimental environment construction

In order to verify the feasibility and effectiveness of the method proposed in this article, real-time data from a provincial smart grid big data information platform in October was used as the experimental object, and a real-time data set size of 125 GB was collected. On this basis, the collected data is divided based on the power simulation system, generating a total of 120 power grid situation evaluation indicators including line flow distribution indicators, bus voltage indicators, active power margin, system load rate, system overload degree, node voltage offset indicators, distributed power source penetration rate, etc. (each data includes all power grid situation evaluation indicators). The specific number of generated indicators is as follows:

(1) Dynamic security situation assessment indicators: 15 data points on line flow distribution indicators; 12 pieces of bus voltage indicator data; 23 active power margin data;

(2) Static security situation assessment indicators: 18 system load rate data; 21 pieces of system overload degree data; 16 pieces of node voltage offset index data; 15 pieces of penetration rate data for distributed power sources.

In the experimental analysis process, the generated data will be used as samples for the security situation assessment of the smart grid system. Each piece of data corresponds to one sample, for a total of 120 samples. In order to ensure the generalization ability of the model and avoid overfitting, it is necessary to retain a portion of the data for testing. Therefore, this study selected approximately 70% (85 samples) as training samples and approximately 30% (35 samples) as testing samples.

For the sample data, first, normalization is performed to treat dynamic and static security indicators as one analysis subject. Normalize each indicator to eliminate indicator errors and form a unified comprehensive indicator of the power grid situation. Then, the Analytic Hierarchy Process is used to determine the weight coefficients of each indicator value. By repeatedly determining the weight coefficients of multiple indicators, the comprehensive indicator data is multiplied with the corresponding weights to calculate the results of the security situation assessment of the smart grid.

Preprocess the power grid situation evaluation index data in the test samples using the comprehensive index generation method described in the article. During training, randomly select a fixed number of samples in each iteration to form a small batch dataset for training until the preset number of iterations is reached. The configuration information of the software and hardware devices included in the experimental environment is shown in Table 2.

Table 2: Configuration information of experimental software and hardware equipment

Type of experimental equipment	Device model	Performance parameters/running version
Hardware equipment	PowerEdge R740 server	Processor: 20 cores, 40 threads, base frequency 2.3 GHz, max RWD 3.9 GHz. Memory: Supports up to 1.5 TB of DDR4 memory. Storage: Equipped with multiple hard disks 8 x 1TB SAS hard disks in a RAID array.
	KC705 FPGA development board	Logical Resources: The number of LCs is about 326,000 and the number of CLBs is about 40,750.

		Storage Resources: Total BRAM capacity is 18.5 Mb.
Software equipment	Linux operating system	Ubuntu 18.04 LTS
	Python programming language	Python 3.7
	TensorFlow deep learning framework	TensorFlow 2.3

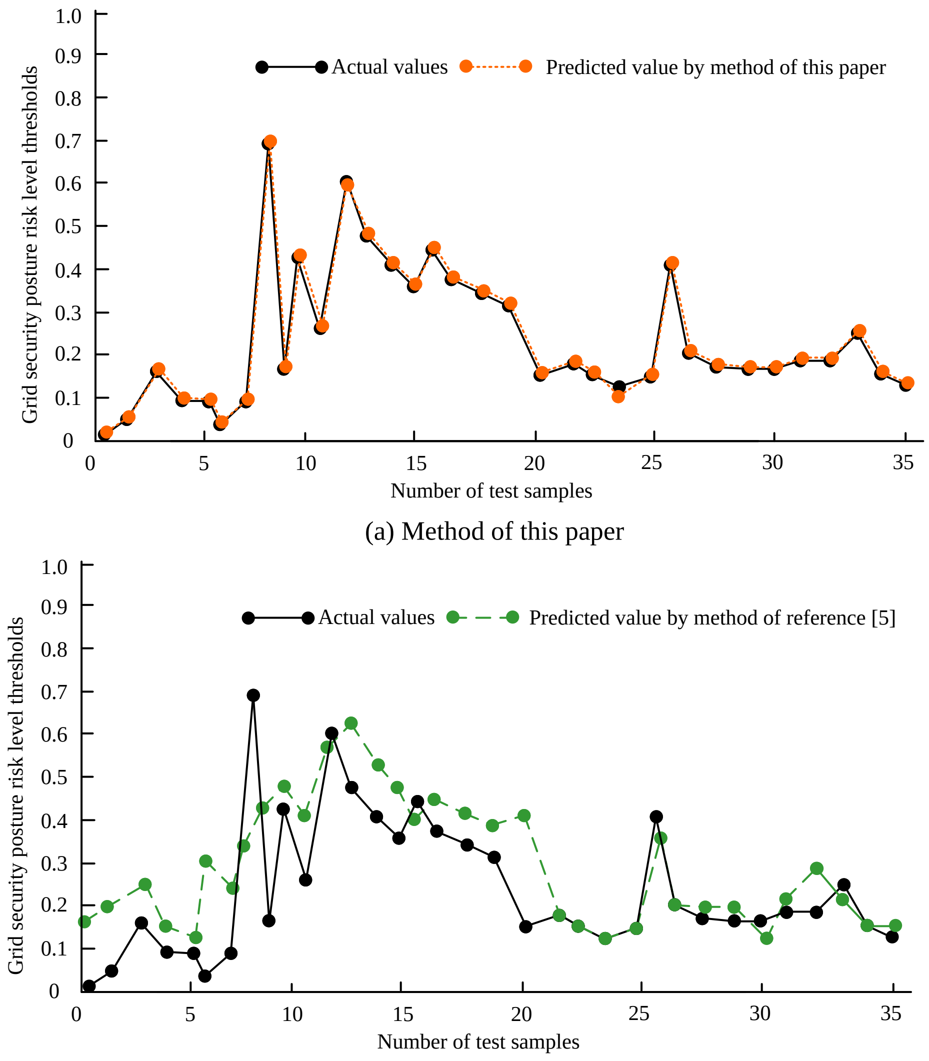
The experimental simulation parameters are shown in Table 3.

Table 3: Experimental simulation parameters

Simulation parameters	Parameter value
Input data time step	120
Output data time step	1
Number of hidden layers of LSTM network	2
Number of hidden units	128
Hidden layer activation function	tanh function
Learning rate	0.5
Number of iterations	1000

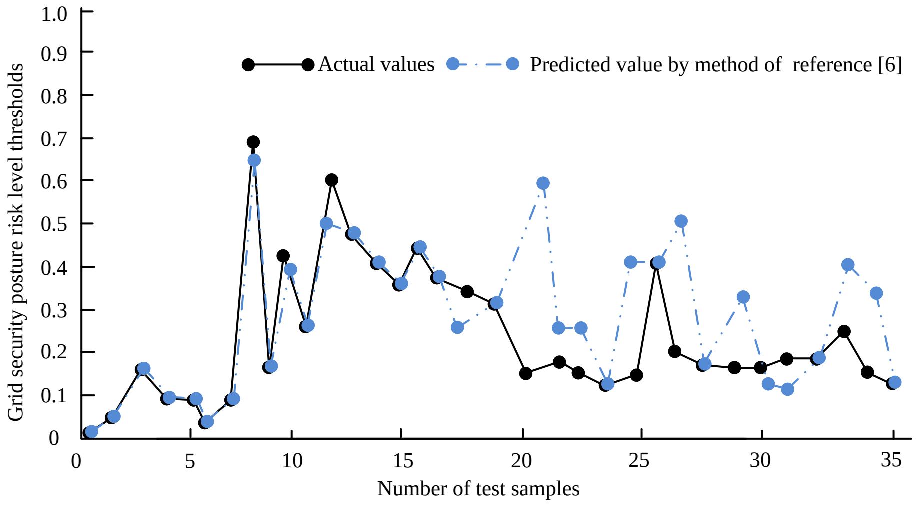
3.2 Testing the effect of smart grid security situation awareness

In order to verify the practical application effect of method of this paper in intelligent network security situational awareness and analysis, method of reference [5] and method of reference [6] were introduced as comparative methods, both of which were trained and learned 1000 times, and then predicted on the test samples. Based on the warning threshold and risk level of smart grid security situation in Table 1, the predicted value of the test sample is defined as the warning threshold range of smart grid security situation risk level. Compare the predicted values of the obtained test samples with the actual values to verify the effectiveness of the smart grid security situation prediction. The specific results are shown in Figure 3.



(a) Method of this paper

(b) Method of Bai et al. [5]



(c) Method of Gong et al. [6]

Figure 3: Test results of the perception effect of smart grid security situation (P-values<0.05)

As shown in Figure 3, using the method of this paper to predict the security situation of the smart grid for the test samples, the predicted results (i.e., the warning threshold corresponding to the risk level of the samples) are consistent with the actual values, and the overall fit is high. There is no situation where the predicted risk level deviates from the actual value. However, the overall fit between the predicted results obtained using the method of Bai et al. [5] and method of Gong et al. [6] and the actual values of the test samples is low, and there is a significant deviation between the predicted results and the actual values regardless of the risk threshold of the test samples. Although the method of Bai et al. [5] constructs a neural network model using radial basis functions and combines linear discriminant analysis to detect abnormal situations in the power grid, there are certain shortcomings in the division of risk thresholds for power grid operation. This may make it difficult for the model to accurately classify certain critical data points when judging the operation status of the smart grid, thereby affecting the overall prediction accuracy. Although the method of Gong et al. [6] combines the characteristics of big data in intelligent power networks and proposes a network security situational awareness detection technology based on distributed data analysis, there are still shortcomings in terms of computational resource utilization. This may limit the performance of the model when processing large-scale, high-dimensional data, resulting in a certain deviation between the predicted results and the actual values [27-29].

From this, it can be seen that using the method of this paper can more accurately capture the changing trends of the smart grid situation, predict the risk level of data security situation better, effectively achieve smart grid security situation awareness, and provide more reliable basis for the scheduling and operation of smart grids.

3.3 The effectiveness of identifying the safety situation of smart grid operation

In order to verify the effectiveness of the method of this paper in discriminating the operating situation of smart grids, based on the experimental environment in section 3.2, the accuracy of different methods in discriminating the operating situation of smart grids was analyzed. However, considering that different methods may have inconsistent dimensions in extracting data features. Therefore, when the experimental environment is unified into the same feature quantity (analyzed according to the percentage of training samples to the total sample size), accuracy, recall, and F1 score are used as evaluation indicators to analyze the accuracy of various methods for predicting the safety situation of smart grid operation. The specific results are shown in Table 4.

Table 4: The recognition effect of the safety situation of smart grid operation (P-values <0.05)

Grid situational awareness algorithm	Accuracy/%	Recall rate/%	F1 score/%
Method of this paper	98.72	98.95	99.06
Method of Bai et al. [5]	89.32	90.01	90.93
Method of Gong et al. [6]	90.26	91.74	92.18

According to Table 4, the accuracy of using the method of this paper for identifying the safety situation of smart grid operation is 98.72%, the recall rate is 98.95%, and the F1 score is 99.06%. This indicates that the algorithm has high risk prediction accuracy for smart grid safety situation operation data based on unified feature quantities, and its application is relatively stable. The obtained prediction accuracy numerical results are superior to those obtained by the method of Bai et al. [5] and method of Gong et al. [6]. Due to the inadequacy of the method of Bai et al. [5] in dividing the risk threshold for power grid operation, the model may have misjudgments or omissions in identifying the safety situation of smart grid operation, thereby reducing accuracy and recall. Meanwhile, this deficiency may also affect the performance of F1 scores. The shortcomings of the method of Gong et al. [6] in terms of computational resource utilization may limit the performance of the model when dealing with complex data. This may lead to poor performance of the model in feature extraction, classification prediction, and other aspects, thereby affecting the overall accuracy, recall, and F1 score.

From this, it can be seen that the overall performance of the method of this paper is good, which can accurately and effectively perceive, predict, and analyze the safety situation of smart grid operation.

3.4 Analysis of the effectiveness of smart grid fault warning

During the simulation testing process of smart grid situational awareness and fault warning, a total of 1000 iterations were executed. In order to further verify the effectiveness of smart grid fault warning, the number of iterations to be executed will be uniformly divided into 5 planning units. Each unit calculates the accuracy of fault warning, response time of fault command, amount of resources consumed (all average values during the iteration process) generated by method of this paper, method of Bai et al. [5], and method of Gong et al. [6] during 200 iterations to verify the effectiveness of intelligent power grid fault warnings. Among them:

(a) Accuracy of fault warning: This indicator is the core standard for measuring the performance of smart grid fault warning methods. It represents the proportion of correctly predicted faults and issuing warning signals. High accuracy of fault warning means that the fault warning method can accurately distinguish between normal operating conditions and critical states before faults, providing strong guarantees for the safe and stable operation of the power grid.

(b) Response time of fault command: This indicator reflects the speed at which smart grid fault warning methods issue warning instructions after detecting faults. A shorter response time for fault instructions means that the fault warning method can respond to faults faster, buying valuable time for subsequent fault handling.

(c) Amount of resources consumed: This indicator measures the computational resources and storage space required for the operation of smart grid fault warning methods. Lower resource consumption means that fault warning methods can operate in a more economical way, reducing operational costs.

The specific test results of the intelligent grid fault warning effect are shown in Table 5.

Table 5: Smart grid fault warning effectiveness (P-values <0.05)

Experimental indicators	Iterations/times	Method of this paper	Method of Bai et al. [5]	Method of Gong et al. [6]
Accuracy of fault warning/%	200	98.12	88.34	90.42
	400	98.45	88.65	91.27
	600	98.96	90.03	91.39
	800	99.16	91.26	91.87
	1000	99.82	91.38	92.08
Response time of fault command/s	200	0.096	0.089	0.098
	400	0.085	0.098	0.107
	600	0.074	0.105	0.112
	800	0.091	0.107	0.101
	1000	0.083	0.112	0.118
Amount of resources consumed/MB	200	125.36	152.65	150.16
	400	123.28	153.78	155.79
	600	120.54	153.96	160.24
	800	119.16	155.02	161.77
	1000	118.57	155.28	168.56

According to Table 5, as the number of iterations continues to increase, the accuracy of fault warning, response time of fault command, and amount of resources consumed generated by our method are generally superior to other methods. The maximum accuracy of fault warning is 99.82%, the minimum response time of fault instructions is 0.083 s, and the minimum amount of resources consumed is 118.57 MB, indicating that our method has a good power grid fault warning effect. Due to the inaccuracy of the method of Bai et al. [5] in dividing the risk threshold of power grid operation, the model may deviate in judging the fault state, thereby reducing the accuracy of fault warning. Meanwhile, this deviation may also affect the response time of fault command, making it difficult for the model to respond quickly after detecting a fault. The insufficient utilization of computational resources in the method of Gong et al. [6] may lead to performance degradation of the model when processing large amounts of data. This may result in a longer response time for the model during the fault warning process, while consuming more computing resources. This deficiency limits the efficiency and reliability of the model in practical applications.

From this, it can be seen that the method of this paper has strong understanding and analysis capabilities for the operation status of the power grid, high resource utilization, and can accurately distinguish between normal operation status and critical status before faults. At the same time, there is a good connection with the subsequent fault handling mechanism. After the fault command is output, it can quickly connect to the power grid system for early warning response, and the entire system can quickly respond to the warning.

4 Discussion

Based on the analysis of the above experimental results, it can be concluded that the method proposed in this paper has good performance in the fit between safety situation prediction results and actual values, safety situation discrimination, and fault warning, while the application effect of the two comparative methods is relatively inferior. Now use Table 6 to conduct a detailed analysis of the two comparison methods.

Table 6: Analysis of two comparative methods

Method	Specific process	Result	Limitations analysis
Method of Bai et al. [5]	A neural network model was constructed using radial basis functions to comprehensively process the operational data of the power grid system. Based on this, linear discriminant analysis was introduced into the model to establish an abnormal situation detection model for the power grid, which is used to perceive the trend of smart grid operation.	(a) The fit between the predicted results and the actual values is relatively low; (b) The accuracy of identifying the safety situation of smart grid operation is 89.32%, the recall rate is 90.01%, and the F1 score is 90.93%. In terms of numerical performance, it is inferior to the method of this paper; (c) The highest accuracy of fault warning is 91.38%, the minimum response time of fault command is 0.089 seconds, and the maximum amount of resources consumed can reach 155.28 MB. In terms of numerical performance, it is inferior to the method of this paper.	Although this method uses RBF to construct a neural network model and combines LDA to detect abnormal situations in the power grid, the RBF neural network has the problem of insufficient generalization ability when processing high-dimensional and complex data. LDA is difficult to fully capture the subtle changes in smart grid data in feature extraction and classification, which affects the perceptual accuracy of this method.
Method of Gong et al. [6]	By applying the cross entropy function and linear units, the loss evaluation part of the neural network model was optimized, and a fusion improved linear unit structure was constructed to achieve perception of the operation status of the smart grid.	(a) The fit between the predicted results and the actual values is relatively low; (b) The accuracy of identifying the safety situation of smart grid operation is 90.26%, the recall rate is 91.74%, and the F1 score is 92.18%. In terms of numerical performance, it is inferior to the method of this paper; (c) The highest accuracy of fault warning is 92.08%, the minimum response time of fault command is 0.098 s, and the maximum amount of resources consumed can reach 168.56 MB. In terms of numerical performance, it is inferior to the method of this paper.	Although this method optimizes the loss evaluation part of the neural network model through cross entropy function and linear unit, and constructs a model that integrates improved linear unit structure, it is still difficult to fully learn the intrinsic rules of the data when dealing with large-scale and high-dimensional data such as smart grids, resulting in prediction accuracy and reliability. Moreover, this method has shortcomings in terms of computational resource utilization, which limits the performance of the model when processing large-scale data and reduces the real-time performance of the method.

The method for this paper has adopted effective strategies to overcome the difficulties of situation awareness and fault warning in smart grids. Firstly, in response to the complexity and temporal nature of power grid data, a long short-term memory network model is adopted, which utilizes its powerful sequence data processing capabilities to effectively capture long-term dependencies in power grid data and improve the accuracy of situation prediction. Secondly, by constructing a smart grid digital twin situational awareness framework, comprehensive real-time monitoring of the power grid status has been achieved, providing a solid foundation for accurate early warning. In addition, the method for this paper also comprehensively considers the dynamic and static security of the power grid, selects indicators that comprehensively reflect the operation trajectory of the power grid, and further improves the accuracy of situational awareness and fault warning.

The innovative work of method for this paper is as follows: on the one hand, by combining digital twin technology and deep learning models, comprehensive mapping and real-time monitoring of the power grid status have been achieved, providing strong guarantees for the safe operation of the smart grid. On the other hand, by introducing LSTM networks, the shortcomings of traditional methods in processing time-series data have been effectively addressed, improving the accuracy and efficiency of situation prediction and fault warning. In addition, the method for this paper also proposes the principles and methods for selecting indicators for evaluating the situation of smart grids, providing new ideas for research in related fields.

In summary, the method for this paper has significant innovation and application value in the field of smart grid situational awareness and fault warning.

5 Conclusion

In summary, this article comprehensively introduces a smart grid situational awareness and fault warning method that combines deep learning technology. This method is based on the digital twin smart grid platform and constructs a smart grid digital twin situational awareness framework. By selecting situational evaluation indicators that can comprehensively reflect the dynamic and static security of the smart grid, real-time monitoring and prediction of the power grid status are achieved. The core lies in utilizing Long Short Term Memory (LSTM) networks for deep learning analysis of power grid data, effectively capturing long-term dependencies in the data, thereby accurately assessing the power grid safety situation and providing fault warnings. The experimental results show that this method exhibits high accuracy, high recall rate, and high F1 score in safety situation prediction, discrimination, and fault warning, with high accuracy of fault warning, short response time of fault command, and low amount of resources consumed. This article provides an efficient and reliable solution for the safe operation of smart grids, demonstrating the enormous potential and application value of deep learning technology in the field of smart grids.

In the next stage of work, we are considering exploring alternative deep learning architectures to achieve significant performance improvements in temporal data processing. At the same time, considering the scalability issues that current research may face when dealing with large-scale datasets, especially when dealing with test datasets exceeding 125 GB, efforts should be made to research and develop more efficient data processing algorithms and parallel computing technologies to alleviate potential limitations on computing resources and ensure smooth response to larger scale data challenges.

Funding

This work was supported by Henan Province Science and Technology Research Project: Research and application of machine learning in equipment fault diagnosis of smart grid (Grant: 242102240122).

References

[1] Zhang, M., Liu, Y., Cheng, Q., Li, H., Liao, D., & Li, H. (2024). Smart grid security based on blockchain and smart contract. Peer-to-Peer Networking and Applications, 17(4):2167-2184.

[2] Xia, Y., Zhang, X., Ge, H., Hao, S., & Zou, W. (2021). Optimal dispatching technology of distributed power generation based on situation awareness. American Journal of Electrical and Electronic Engineering, 9(1):7-11.

[3] Bondarenko, A. F., Govorun, M. N., & Satsuk, E. I. (2024). About the Brazilian power grid accident on August 15, 2023. Power Technology and Engineering, 58(3), 527-534.

[4] Presekal, A., Ştefanov, A., Rajkumar, V. S., & Palensky, P. (2023). Attack graph model for cyber-physical power systems using hybrid deep learning. IEEE Transactions on Smart Grid, 14(5), 4007-4020.

[5] Bai, J., Jiao, J., Han, M., Zhou, X., & Liu, C. Research on substation network security situational awareness strategy and equipment remote operation and maintenance. Applied Mathematics and Nonlinear Sciences, 9(1), 516-527.

[6] Gong, X., Wu, X., & Zhou, X. (2023). Deep learning-based security situational awareness and detection technology for power networks in the context of big data. Applied Mathematics and Nonlinear Sciences, 8(1), 2939-2956.

[7] Zhai, C., Nguyen, H. D., & Zong, X. (2022). Dynamic security assessment of small-signal stability for power grids using windowed online Gaussian process. IEEE Transactions on Automation Science and Engineering, 20(2), 1170-1179.

[8] Wang, P., Zhang, D., Gan, L., & Zhang, Y. (2024). Key technologies and applications of collaboration between digital power grid and Internet of Things. Digital Twins and Applications, 1(1):26-37.

[9] Han, J., Chen, Z., Hu, P., Li, H., Li, G., & Pi, T. (2023). Digital twin power grid oriented mobile edge network resource allocation model. IEEJ Transactions on Electrical and Electronic Engineering, 18(10):1682-1693.

[10] Steffen, A, & Tarik, R. (2023). Digital solutions for future grid complexity: The change in the grid forces the adoption of digital solutions to manage future complexity. Transformers Magazine, 10(SE2):44-47.

[11] Weng, L., Yang, L., Lei, Z., Huang, Z., & Chen, Y. (2024). Integrated bus voltage control method for DC

microgrids based on adaptive virtual inertia control. Journal of Power Electronics, 24(7):1163-1176.

[12] Geng, Q., Sun, H., Zhou, X., & Zhang, X. (2023). A storage-based fixed-time frequency synchronization method for improving transient stability and resilience of smart grid. IEEE Transactions on Smart Grid, 14(6), 4799-4815.

[13 Zhang, M., Liu, Y., Cheng, Q., Li, H., Liao, D., & Li, H. (2024). Smart grid security based on blockchain and smart contract. Peer-to-Peer Networking and Applications, 17(4):2167-2184.

[14] Li, Y., Ren, R., Huang, B., Wang, R., Sun, Q., Gao, D. W., & Zhang, H. (2022). Distributed hybrid-triggering-based secure dispatch approach for smart grid against DoS attacks. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 53(6), 3574-3587.

[15] Hisham, M., Abdellah, B., Farag, A., & Kenan, B. (2024). Grid frequency stabilisation under magnitude and generation rate constraints. International Journal of Modelling, Identification and Control, 44(2):172-180.

[16] Sanju, K., Neeraj, K., & Prashant, S. (2024). Comparative performance study of different filtering techniques with LSTM for the prediction of power consumption in smart grid. IETE Journal of Research, 70(4):3646-3663.

[17] Yang, S., Yuan, A., & Yu, Z. (2022). A novel model based on CEEMDAN, IWOA, and LSTM for ultra-short-term wind power forecasting. Environmental Science and Pollution Research International, 30(5):11689-11705.

[18] Ngamroo, I., & Surinkaew, T. (2023). Control of distributed converter-based resources in a zero-inertia microgrid using robust deep learning neural network. IEEE Transactions on Smart Grid, 15(1), 49-66.

[19] Diaba, S. Y., & Elmusrati, M. (2023). Proposed algorithm for smart grid DDoS detection based on deep learning. Neural Networks, 159(1), 175-184.

[20] Zhang, X., Li, C., Xu, B., Pan, Z., & Yu, T. (2022). Dropout deep neural network assisted transfer learning for bi-objective Pareto AGC dispatch. IEEE Transactions on Power Systems, 38(2), 1432-1444.

[21] Rashmi, B., Matushree, K., Anamika, Y., & Mohammad, P. (2024). Load forecasting model using LSTM for Indian state load dispatch centre. Electrica, 24(3):601-615.

[22] Wang, Y., Liu, Y., Wang, M., Dinavahi, V., Liang, J., & Sun, Y. (2024). Resilient smart power grid synchronization estimation method for system resilience with partial missing measurements. CSEE Journal of Power and Energy Systems, 10(3):1307-1319.

[23] Attia, H., Takruri, M., & Al-Ataby, A. (2024). Intelligent algorithm-based maximum power point tracker for an off-grid photovoltaic-powered direct-current irrigation system. Clean Energy, 8(3), 48-61.

[24] Liang, H., Qian, C., Yu, W., Griffith, D., & Golmie, N. (2024). Assessing deep learning performance in power demand forecasting for smart grid. International Journal of Sensor Networks, 44(1), 36-48.

[25] Agrawal, A., Das, N., Jain, S. K., & Kulhar, K. S. (2024). LSTM controllers for power quality improvement in grid connected hybrid wind-pv-battery based power supply system. In E3S Web of Conferences. EDP Sciences. 540, 10007.

[26] Zhang, Z., Qin, B., Gao, X., & Ding, T. (2023). CNN-LSTM based power grid voltage stability emergency control coordination strategy. IET Generation, Transmission & Distribution, 17(16), 3559-3570.

[27] Hou, C., Xu, N., & Liu, S. (2025). Design of online monitoring method for distribution IoT devices based on DBSCAN optimization algorithm. Informatica, 49(5):181-194.

[28] Zhang, Y., Gao, Y., & Zhao, Z. (2025). Research on operation and anomaly detection of smart power grid based on information technology using CNN+Bidirectional LSTM. Informatica, 49(7):157-164.

[29] Huang, Q., Xian, H., Mei, L., Cheng, X., & Li, N. (2025). Intelligent distribution network operation and anomaly detection based on information technology. Informatica, 49(9):123-134.

A Random Forest-Based Machine Learning Framework with PCA, SMOTE, and SHAP for Efficient and Interpretable Coronary Artery Disease Prediction

T. Aswani[1*], Dr. Jose Moses Gummadi[2], Dr.G. Sharada[3]
[1]Research Scholar, Department of Computer Science and Engineering, School of Engineering, Malla Reddy University, Hyderabad, Telangana, India
[2]Department of Computer Science and Engineering, School of Engineering, Malla Reddy University, Hyderabad, Telangana, India
[3]Department of Computer Science and Engineering, Malla Reddy College of Engineering and Technology(A), Hyderabad, Telangana, India
Email: 2232CS010021@mallareddyuniversity.ac.in, josemoses@gmail.com, gsharada8@gmail.com
[*]Corresponding author

Keywords: machine learning framework, coronary artery disease prediction**,** feature engineering**,** class imbalance handling**,** model interpretability

Received: January 8, 2025

Given that coronary artery disease (CAD) is a major global cause of morbidity and mortality, there is an urgent need for precise and scalable diagnostic tools. While conventional machine learning (ML) models such as XGBoost and Gradient Boosting have demonstrated good predictive performance, they suffer from limitations, including weak handling of class imbalance, redundant feature spaces, and lack of interpretability. This work proposes an optimized Random Forest-based framework for CAD prediction to address these gaps, integrating advanced feature engineering and optimization techniques. Specifically, dimensionality reduction is achieved using principal component analysis (PCA), class imbalance is handled through the Synthetic Minority Oversampling Technique (SMOTE), and hyperparameter optimization is performed via GridSearchCV, tuning parameters such as the number of estimators, maximum depth, and minimum samples split. Additionally, SHAP (Shapley Additive exPlanations) values enhance interpretability by illustrating the contribution of each feature to the model's predictions; for example, features such as chest pain type and cholesterol level are shown to influence CAD outcomes significantly. The proposed framework is evaluated on the UCI Heart Disease dataset comprising 303 samples. Experimental results demonstrate that the optimized Random Forest model achieves an accuracy of 95.0%, outperforming Gradient Boosting (93.08%) and XGBoost (92.4%) classifiers. This framework provides a clinically relevant, interpretable, and scalable solution for CAD prediction, bridging the gap between technical advancements and their practical deployment in healthcare environments.

Povzetek: Razvit je izboljšan okvir za napovedovanje koronarnih arterijskih bolezni, ki uporablja algoritem naključnih gozdov, PCA za zmanjšanje dimenzionalnosti, SMOTE za ravnotežje razredov ter analizo SHAP za povečanje interpretabilnosti modela, kar omogoča klinično relevantno napovedovanje.

1 Introduction

Since coronary artery disease (CAD) is a primary global source of morbidity and death, early and precise diagnostic methods are crucial. Recent advances in machine learning (ML) have influenced CAD prediction, providing an excellent option to integrate clinical, diagnostic, and imaging data. Using a range of models, such as Gradient Boost and XGBoost, has shown promising results in predictive performance in existing studies. However, such methods have severe issues, such as class imbalance, redundant features, and inadequate generalizability. Moreover, the lack of interpretability inherent in most state-of-the-art approaches impedes their uptake in clinical practice.

These issues have been the subject of recent studies. For instance, Gupta et al. [13] utilized SMOTE and augmented features for high accuracy. However, Hashemi et al. [15] proposed integrating genetic algorithms with the one-layer multi-layer perceptrons for better predictions. While these approaches have been promising, they still leave a massive gap in scaling traditional ML models that achieve the optimal trade-off between accuracy, scalability, and interpretability. This work strives to address this gap by building upon existing implementations of CAD prediction to provide an optimized framework using RForest with improved feature engineering and advanced hyperparameter tuning methods.

This study attempts to create a machine-learning framework that enhances CAD prediction by overcoming the main limitations of existing approaches. The proposed methodology comprises a combination of PCA for dimensionality reduction, SMOTE for addressing the class imbalance, and GridSearchCV for hyperparameter tuning, among other novelties, leading to improved predictive performance and robustness. SHAP values are used for interpretability to elucidate feature contribution and improve model clinical relevance.

This study aims to develop a framework based on machine learning (ML) that includes three fundamental challenges in coronary artery disease (CAD) prediction: (1) clinical datasets suffer from class imbalance, (2) redundancy of features may lead to overfitting and generalizability, and (3) the lack of interpretability of the model. More specifically, we propose that the inclusion of PCA for dimensionality reduction, SMOTE for balancing classes, and SHAP values for interpretability of features used in the model, in conjunction with hyperparameter optimization Random Forest models, will provide better overall prediction accuracy robustness, and relevant to clinical practice than other pre-existing models such as Grad Boost and XGboost.

Specifically, the main contributions of this paper comprise: (1) an optimized implementation of Random Forest for CAD prediction; (2) the enhanced use of feature engineering techniques to boost model quality; (3) ensuring better interpretability of the model, using SHAP values, which is one of the significant drawbacks of currently used ML-based models; and (4) comparison against state-of-the-art models to validate our approach. The paper is organized as follows. In Section 2, existing CAD prediction methods are discussed, and research gaps are defined through a complete literature review. Section 3 outlines the proposed approach involving data preparation steps, feature engineering, and model tuning methods. Section 4 shows the experimental findings and assesses how well the suggested framework compared to state-of-the-art models. Section 5 presents the study's findings, contributions, and limitations, with Section 5.1 devoted to constraints. Finally, Section 6 summarizes the paper and discusses the study's overall importance for improving CAD diagnosis and patient care and possible future directions.

2 Related work

This literature review highlights advancements in machine learning-based approaches for coronary artery disease (CAD) prediction and management. Bertsimas et al. [1] developed ML4CAD, a personalized CAD management system with an 81.5% AUC, using EMR data. Future work will include validating clinical trials, including socioeconomic characteristics, and improving generalizability. Varuna et al. [2] recommended applying a two-phase AI model to identify coronary artery disease with 96.2% accuracy. More studies will try to make it more generalizable and expand its use to include other illnesses. Gabriel Anbarasi [3], the BSOXGB model outperforms previous approaches with a CAD recognition success rate of 97.70% because of enhanced feature selection and hyperparameters. Testing with additional datasets will be part of future efforts. Huang et al. [4], the RF model predicts CHD remarkably effectively using CACS and clinical factors. Future work will include handling missing data and improving models. Manduchi et al. [5] demonstrated how well TPOT can detect SNPs linked to CAD, while it has issues with big datasets, runtime, and heterogeneous data.

Zahia et al. [6] used feature selection and data balance to develop a hybrid machine-learning model that detects CAD with 98.34% accuracy. Jahmunah et al. [7] introduced a GaborCNN model for ECG-based CAD diagnosis with a 98.5% accuracy rate and potential for faster, more effective clinical use. Arian et al. [8] predicted myocardial function improvement after CABG using LGE-CMR images radiomics and machine learning, with encouraging findings. Umar Khan et al. [9] suggested a signal processing technique for CAD prediction that uses ECG data and SVM. It achieves 95.5% accuracy and suggests deep learning for future advancements. Nasarian et al. [10], a hybrid feature selection approach for CAD, is presented in this work, which achieves excellent accuracy by utilizing a variety of classifiers and balancing strategies. Expanding datasets and investigating evolutionary algorithms are the goals of future research.

Abdar et al. [11] proposed a novel machine-learning approach for CAD identification with an accuracy of 93.08%. Additional preprocessing methods, algorithms, and evolutionary approaches will be investigated in further studies. Li et al. [12] improved risk group categorization by creating a framework for risk stratification with machine learning assistance to streamline CAD diagnosis. Ongoing research might develop these techniques further. Gupta et al. [13] said that the C-CADZ system outperformed earlier techniques in achieving 97.37% accuracy for CAD diagnosis utilizing FAMD and SMOTE. Future research might improve multi-class categorization and the handling of class imbalance. Varun et al. [14], a deep neural network diagnosed CAD with 96.2% accuracy using Gaussian noise to reduce overfitting; future work will concentrate on extending to other ailments. Hashemi et al. [15] employed genetic algorithms and machine learning to predict CAD with 94.71% accuracy; deep learning advancements will be the main focus of future work.

Nesaragi et al. [16] presented a tensor-based machine learning system that achieves 96.62% accuracy in CAD identification using heart rate data. Further research will improve this approach. Saruladha and Swathy [17] examined AI and data mining strategies for predicting CVD, emphasizing the need for more information and customized approaches. Huang et al. [18] AI accelerates productivity and improves the accuracy and efficiency of computed tomography angiography (CCTA), a technique used to diagnose computer-aided design (CAD). Khozeimeh et al. [19] Active Learning with Ensemble of Classifiers (ALEC) improves the diagnosis of CAD by lowering the risks and costs associated with invasive angiography. Qiao et al. [20] suggested that ML-based FFRCT may improve CAD diagnosis and decision-making compared to invasive angiography; nevertheless, more validation is needed.

Alizadehsani et al. [21] presented a high-accuracy machine-learning approach to identifying individual cases of coronary artery stenosis while resolving model uncertainty. Omkari and Shaik's [22] TLV model uses ensemble voting and machine learning to achieve high accuracy in CAD diagnosis on large datasets, which makes it perform better than previous methods. Braun et al. [23], a non-invasive, precise, and economical CAD screening technique, is provided by cardiography, which combines vector cardiography with machine learning. Wishart et al. [24] provided a cost-aware feature selection technique to identify coronary heart disease with excellent accuracy and AUC using fewer features. Wang et al. [25] provided a two-level stacking machine learning model for CHD diagnosis. Although it produces high accuracy, the dataset's amount and the parameters' values are limited.

Ahmad et al. [26] LR, KNN, SVM, and GBC approaches are examined in this study and shown to be less accurate than Extreme Gradient Boosting with GridSearchCV in predicting cardiac disease. Yan et al. [27] created a machine learning-based system that uses XGBoost for high classification accuracy and customized patient advice to diagnose coronary artery stenosis. Cheung et al. [28] provided a 2D UNET model for accurately segmenting coronary arteries on CTCA photos using the least computer resources. Spadarella et al. [29] Radiomics and machine learning deliver promising advancements to cardiovascular imaging despite ongoing issues with study consistency and model interpretability. Benjamins et al. [30] state that the capacity to identify myocardial ischemia and the necessity for early revascularization is improved by combining machine learning with CTA and clinical data, albeit further validation is needed.

Molenaar et al. [31] Artificial intelligence in invasive coronary angiography is advancing, even if more multicenter datasets and external validation are required for broader applications. Muhammad and Algehyne [32] enhanced the C4.5 algorithm, which was used to create a fuzzy-based expert system for CAD in Nigeria that produced high dependability and accuracy of 94.55%. Hagan et al. [33] highlighted different training costs and accuracy across datasets when comparing machine learning techniques for diagnosing cardiovascular disease. Brandt et al. [34] assessed CT-derived fractional flow reserve (CT-FFR), which may reduce the requirement for invasive angiography to identify substantial CAD in patients with severe aortic stenosis. Liu et al. [35] evaluated a machine-learning model for severe CAD prediction using routine data to reduce invasive procedures and improve diagnosis accuracy.

Table 1: Summary of existing machine learning models for CAD prediction

Study (Author, Year)	**Dataset Used**	**Methodology / Model**	**Performance (Accuracy)**	**Interpretability Addressed**	**Class Imbalance Handling**	**Dimensionality Reduction**
Bertsimas et al. (2020) [1]	EMR Data	ML4CAD (Multiple Models)	81.5% (AUC)	No	Not explicitly addressed	Not addressed
Varuna et al. (2023) [2]	Custom Dataset	Two-phase AI Model (Deep Learning)	96.2%	No	Not explicitly addressed	Not addressed
Gabriel et al. (2023) [3]	Public Dataset	BSOXGB (XGBoost + Feature Selection)	97.7%	Partial (XGBoost + SHAP support but not emphasized)	Not mentioned	Feature Selection (not PCA)
Zahia et al. (2020) [6]	Clinical Dataset	Hybrid ML Model with Feature Selection	98.34%	No	Balancing techniques used	Feature Selection (not PCA)
Jahmunah et al. (2021) [7]	ECG Data	GaborCNN (Deep Learning)	98.5%	No	Not addressed	Not addressed
Abdar et al. (2019) [11]	UCI Cleveland Dataset	Hybrid ML Approach	93.08%	No	Not specified	Not specified

Gupta et al. (2021) [13]	Z-Alizadeh Sani Dataset	C-CADZ (FAMD + SMOTE + Classifier)	97.37%	No	SMOTE used	Feature Aggregation (FAMD)
Hashemi et al. (2024) [15]	Public Dataset	Genetic Algorithm + Optimized MLP	94.71%	No	Not mentioned	Genetic Algorithm (not PCA)
Wang et al. (2020) [25]	Public Dataset	Stacking Ensemble Model	90.0%	No	Not specified	Not addressed
Benjamins et al. (2021) [30]	Clinical + CTA Data	XGBoost	92.4%	Not focused	Not specified	Not addressed
Proposed Study (2024)	UCI Heart Disease Dataset (303 samples)	Optimized Random Forest + PCA + SMOTE + SHAP	95.0%	Yes (SHAP used explicitly)	SMOTE applied	PCA applied (8 components)

Militello et al. [36] showed that integrating radiomic and clinical variables enhances the prediction of coronary artery disease compared to utilizing clinical data alone. Nilashi et al. [37] demonstrated how incremental machine learning techniques, especially fuzzy support vector machines (SVM), improve the precision of heart disease detection while cutting down on computation time. Raparelli et al. [38] integrated a variety of characteristics and used machine learning to separate obstructive from non-obstructive CAD; nevertheless, more enormous datasets are required for validation. Yang et al. [39] utilized enhanced LightGBM and focal loss, and the HY_OptGBM model improved early CHD diagnosis with a 97.8% AUC. Cherradi et al. [40] suggested that KNN and ANN-based diagnostic systems outperformed earlier techniques with higher accuracy for predicting atherosclerosis. Significant learning and data mining techniques have obtained up to 60% to 90% prediction accuracy by selecting features through an effective model and addressing the class imbalance problem. Sadri Alija et al. [42] used a supervised learning model and a wrapper-based feature selection component to improve student performance prediction over imbalanced datasets. Harjinder Kaur et al. [43] proposed a prediction framework based on academic performance analysis using machine learning algorithms focusing on the early detection of underperformers. Hua Huang [44] proposed a two-stage feature selection method and enhanced machine learning classifiers for text data classification. These studies demonstrate that optimized feature selection and balanced data learning are critical elements of predictive modeling. A comparative summary of key machine learning studies on CAD prediction is provided in Table 1. Other features include datasets, methodologies, performance, interpretability focus, class imbalance handling, and dimensionality reduction techniques. It also shows the novelty of the proposed approach while addressing the limitations of other existing works and providing robustness and clinical applicability through PCA, SMOTE, hyperparameter tuning, and SHAP interpretability as the final methodologies.

The studies surveyed revealed a range of machine learning models employed for CAD prediction, such as Random Forest, SVM, and hybrid models, resulting in notable accuracy. Feature selection, class balancing, and deep learning — all of these techniques lead to better performance. In future studies, we aim to expand the generalizability, work with larger datasets, as well as incorporate clinical, radiomic, and socioeconomic variables to make the CAD diagnosis robust.

3 Proposed framework

Figure 1 overviews the methodology for predicting coronary artery disease, which represents a systematic process adopted to implement a robust and accurate machine-learning framework. In the first step, the raw dataset was preprocessed. Feature Scales were standardized using the StandardScaler to normalize the data so that each feature contributed equally during the model's training. PCA was applied to the data, so the dimensionality of the data cube was reduced to eight principal components, which provide the best data features without raising the computation price and the possibility of overfitting.

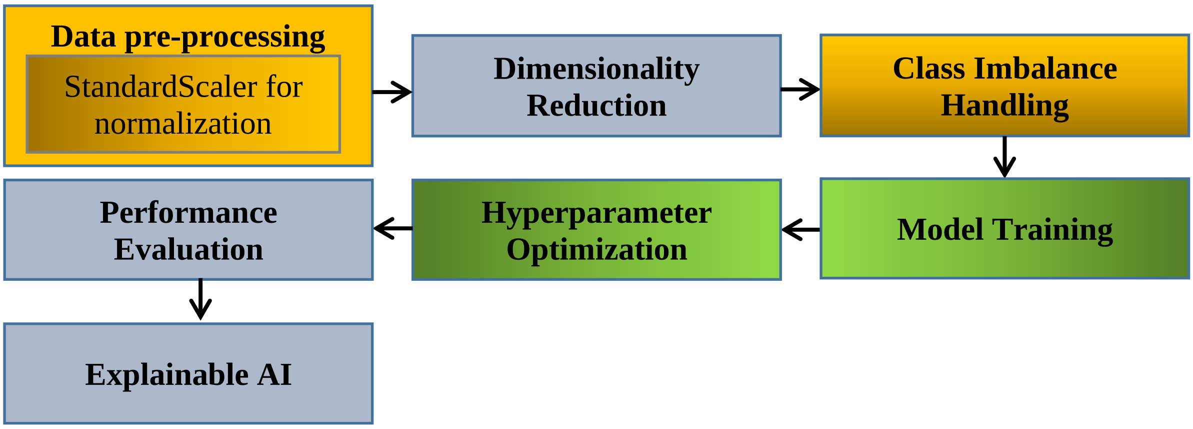


Figure 1: Proposed methodology for coronary artery disease prediction

To rebalance the target, we used the Synthetic Minority Oversampling Technique (SMOTE). Variable. Using this technique, synthetic samples were created for the minority class, which provided a balanced dataset for the model to learn patterns for both classes more effectively. The cleaned and balanced dataset was subsequently chosen for its robustness and appropriateness for high-dimensional data analysis and utilized to train a random forest classifier. Hyperparameter optimization was performed with GridSearchCV by searching over the Using a 5-fold cross-validation technique, the optimized model produced dependable and generalizable findings for the number of estimators, the maximum tree depth, and the least number of samples required to divide a node.

After that, the dataset was split into a 70:30 training-testing set, with the testing set used to assess the model's performance and the training set used to build the model. The performance metrics were based on accuracy, confusion matrix, and classification report (precision, recall, and F1-score). The SHAP (SHapley Additive exPlanations) framework was employed to improve model interpretability. SHAP values calculated the contribution of each feature to predictions, and a summary plot graphically described feature importance, revealing insights on the key predictors of coronary artery disease. The best model was serialized using joblib and exported to a. pkl file, making it suitable for clinical decision support systems deployment. This pipeline, outlined in Figure 1, integrates robust preprocessing techniques complemented by class balancing, hyperparameter tuning, and interpretability to create a strong and transparent framework for practical deployment.

3.1 Machine learning models

The performance of several machine learning models was compared in this study to predict coronary artery disease. Using the KNN technique, because it is a simple yet effective instance-based learning approach, we use the majority class observed among their closest neighbors to classify similar data points. This algorithm can find various local patterns which are the most relevant to the dataset. Furthermore, The SVM's capacity to identify the best hyperplane for dividing data points into distinct classes was also utilized. SVM is beneficial for high-dimensional data and creating robust decision boundaries. The Decision Tree Classifier was also one of the models evaluated (selected due to interpretability and simplicity). That's why this algorithm makes a tree structure that splits the data repeatedly using the value of a particular feature. We also used random Forest, a type of ensemble learning that builds numerous decision trees to increase precision and decrease overfitting. Random Feature Selection and Bootstrap Methodology The above methodology is inherent in Random Forest and is used to contribute to the robustness and reliability of the model partially. These various models gave the study a comprehensive assessment of different classification techniques. The performance of all models was analyzed and compared to the data for identification of coronary artery disease.

3.2 Data preprocessing

Data was preprocessed to prepare for machine learning modeling. We used the StandardScaler to standardize each feature, assigning a standard deviation of one and a mean of zero to the data. This resolution ensured that broader-scale features did not disproportionately affect the model. No null values remained. They were one-hot encoded to prepare categorical variables for use in K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees, and Random Forest models. This preprocessing step was crucial so that all models performed consistently.

First, the dataset was preprocessed to handle missing and categorical data encoding. In particular, categorical variables (e.g., types of chest pain and thalassemia) were preprocessed using one-hot encoding to convert them to numeric format. Feature scaling was executed with StandardScaler to have all features with a zero-mean and one-variance distribution. MinMaxScaler, RobustScaler, and other scaling techniques were tried but again showed a minuscule impact on performance, so the best choice was to use StandardScaler, especially before PCA.

3.3 Dimensionality reduction

Feature dimensionality was reduced using Principal Component Analysis (PCA) after the data were scaled, retaining 95% of the data variance. This threshold was empirically derived; retention of 90% variance was found to exclude clinically relevant features dangerously, and retention of 99% yielded trivial variance benefit with increased model complexity. By eliminating duplicate and correlated features, PCA helped to prevent overfitting and enhanced the efficiency of our models. While PCA is a linear combination of features and thus could impact direct clinical interpretability, the SHAP values were calculated on features before PCA was performed, which ensures the interpretation of feature contributions.

3.4 Handling class imbalance

Instead of a class imbalance (majority class), the target variable showed a high-class imbalance. For the minority classes, synthetic samples were made using the Synthetic Minority Oversampling Technique (SMOTE). This resulted in a balanced distribution required for all four machine learning models to be trained relatively and practically. SMOTE stopped SVM and KNN, Decision Tree, and Random Forest models from learning patterns only on majority classes, thus enhancing the accuracy of their predictions.

SMOTE handled imbalance classes after the one-hot encoding and feature scaling process but before applying PCA transformation. The dataset was also imbalanced, with 55% majority (non-CAD) and 45% minority (CAD) samples. The SMOTE generated synthetic samples for the minority class, balancing the classes' ratios to 50:50, ensuring that the classifier learned the same amount from both classes and improving recall and F1-score.

3.5 Model training

The preprocessed and balanced dataset trained four models (i.e., KNN, SVM, Decision Tree, and Random Forest). On the contrary, KNN predicted a class of data points using a majority vote among the nearest neighbors, so it followed the local pattern of the data. This is due to SVM's capability to work with high-dimensional data by identifying an optimal hyperplane to distinguish between. The Decision Tree model, which segmented the data based on feature values, provided an interpretable structure for decision-making. The Random Forest was an ensemble of decision trees trained using bootstrapping and random feature selection to reduce variance while increasing accuracy. Each of these models was trained separately to provide a complete evaluation.

An A 5-fold cross-validation technique was incorporated during hyperparameter tuning and model evaluation to reduce the chance of overfitting. The dataset was randomly divided into a training set (70% of the data) and a testing set (30% of the data), and all works used a fixed random seed value of 42 to guarantee the replicability of the results. Using a confusion matrix, we used professional metrics to assess models, including accuracy, precision, recall, F1-value, and analysis. To account for class imbalance in the predictive model, cross-validation was used to select the optimal parameters for the model by maximizing the F1 score, which balances performance in both classes.

The training set was utilized for model construction and hyperparameter tuning. The testing set measured the model performance using data never used during training. The testing set can thus be viewed as providing an unbiased performance result for a particular application. Performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix analysis were evaluated. GridSearchCV yielded an exhaustive hyperparameter search, but RandomSearchCV and Bayesian optimization could have also been applied. Due to the moderate size of the dataset with a well-defined parameter space, GridSearchCV was favored for its systematic search strategy without introducing an excessive computational overhead.

Initial experiments included KNN to establish baseline performance, leveraging its ability to capture local data patterns effectively. However, the final optimized model employed Random Forest, chosen for its robustness, superior generalization, and ability to handle feature interactions, which proved essential for CAD prediction.

3.6 Hyperparameter optimization

This way, hyperparameter optimization with GridSearchCV was carried out to obtain each model's peak performance. KNN: suitable number of neighbors and SVM: tunning parameters like kernel and regularization strength. The maximum depth and split criteria were fine-tuned for the Decision Tree model. The number of estimators, maximum depth, and minimum samples for the split were optimized for Random Forest. This led to a 5-fold cross-validation, in which the optimized parameters were replicated across all models, resulting in accurate and generalizable results.

The hyperparameter tuning of the Random Forest model was applied using GridSearchCV with a 5-fold cross-validation strategy. The explored range of parameter values was: number of estimators [50, 100, 200, 300], maximum depth [4, 6, 8, 10, None], minimum samples split [2, 5, 10], and the minimum samples leaf [1, 2, 4]. A random seed of 42 guaranteed that query results could be reproduced. By restricting maximal depth and tuning minimum examples of leaves, Random forests, in their nature, enforced regularization, which kept away overfitting. The F1-score has been in the first place in choosing the best hyper-parameters because we have imbalanced classes; we need to balance precisely and recall.

3.7 Explainable AI (SHAP)

We implemented the SHAP (SHapley Additive exPlanations) framework to interpret the predictions from ML models. The significance of each attribute in predictions was evaluated using SHAP values, which describe the reasoning of KNN, SVM, Decision Tree, and Random Forest. Global interpretation plots (summary plots) were created to examine the relative contributions of features across all the models. This also improved transparency and made the models' predictions explainable, allowing them to be used in a clinical setting, where it is essential to know why a decision is made.

3.8 Proposed algorithm

The proposed Intelligent Coronary Artery Disease Prediction (ICADP) algorithm uses four optimized Random Forest, SVM, KNN, and Decision Tree machine learning models. It combines advanced preprocessing, class balancing, and interpretable predictions via SHAP. Without loss of generality, this algorithm generates a robust, fair, and interpretable predictor that is highly meaningful in clinical settings toward promoting informed healthcare decision-making.

```
Algorithm: Intelligent Coronary Artery Disease Prediction (ICADP)
Input: Dataset (X, Y), Models M = { Random Forest, SVM, Decision Tree, and KNN}, Parameters P_m for m in M
Output: Optimized Models M*, Metrics for each model, Predictions Y_pred

1. Preprocess X: Normalize features, handle missing values
2. Reduce Dimensions:
   X_PCA ← PCA(X, retain 95% variance)
3. Handle Class Imbalance:
   (X_balanced, Y_balanced) ← SMOTE(X_PCA, Y)
4. Split Data:
   (X_train, Y_train), (X_test, Y_test) ← Train-Test-Split(X_balanced, Y_balanced)
5. Train and Optimize Models:
   For each model m in M:
      m* ← GridSearchCV(m, P_m, cv=5)
      m*.fit(X_train, Y_train)
6. Evaluate Models:
   For each optimized model m* in M*:
      Y_pred_m ← m*.predict(X_test)
      Metrics_m ← Evaluate(Y_test, Y_pred_m)
7. Interpret Results:
   For each m* in M*:
      SHAP_values_m ← SHAP(m*, X_test)
8. Return M*, Metrics_m, Y_pred_m, SHAP_values_m
```

Algorithm 1: Intelligent coronary artery disease prediction (ICADP)

For accurate profiling of coronary artery disease, the ICADP algorithm systematically utilizes various ML models to implement and predict CAD effectively. The next step is to preprocess the dataset to fit it into the machine learning format. The features are standardized with StandardScaler, which gives them a mean of zero and a standard deviation of one. So, this step removes bias from different scales among the features to ensure consistency. It also handles missing values and encodes categorical variables to guarantee that the information aligns with the models. A PCA is applied to this initial dataset to reduce dimensionality. Retaining only the eight principal components with the highest variance helps the algorithm retain significant data while discarding redundancy, streamlining the feature space, and reducing the likelihood of overfitting. The Synthetic Minority Oversampling Technique balances the dataset's classes. To combat this, SMOTE creates artificial samples for the minority class and, in turn, gets a better-balanced dataset to ensure that all models learn better for both class patterns. This preprocessed, balanced data is split into training and testing different subsets, with a ratio of 70 to 30 to guarantee sufficient data for model evaluation and training.

Next, four machine learning models were used: Random Forest, Decision Tree, Support Vector Machine, and K-Nearest Neighbors (KNN). These are optimized individually through hyperparameter tuning with the help of GridSearchCV. We are running hyperparameter tuning with a different grid search space for each model for tuning settings like the number of KNN neighbors, SVM kernel type, Decision Tree maximum depth, and Random Forest number of estimators. The optimization employs a 5-fold cross-validation strategy, resulting in reliable and generalizable results for every model.

After these steps have been optimized, the selected models are trained with the balanced training data and evaluated with the testing dataset. Predictions are obtained from each model, and performance metrics such as F1-score, confusion matrix, recall, accuracy, and precision are calculated. These metrics give a complete analysis of the effectiveness of each model and a comparison of performance. The Shapley Additive Explanations, or SHAP framework, is used to improve the interpretability of the algorithm. Computing SHAP values abstracts how much each feature has contributed to the predictions and, thus, provides a comprehensive view of all four models' prediction logic. We generate summary plots to visualize how features rank globally in importance, establishing a transparent basis to support transferable clinical uses of our interpretable framework. Ultimately, the ICADP algorithm produces the tuned models, those model's performance metrics, and the SHAP-based interpretations. This end-to-end process also guarantees robustness, accuracy, and interpretability for the employed, leading to generalizability that renders the framework well-suited for real-world coronary artery disease prediction scenarios.

3.9 Dataset details

The dataset [41] that is used to predict coronary artery disease consists of 303 samples with 14 attributes, i.e., Age, Sex, Chest Pain Type (cp), Resting Blood Pressure (trtbps), Cholesterol Level (chol), Fasting Blood Sugar (FBS), Resting Electrocardiographic Result (restecg), Maximum Heart Rate Achieved (thalachh), Exercise Induced Angina (exng), ST Depression Induced by Exercise (oldpeak), Slope of Peak Exercise ST Segment (slp), No of Major Vessels (caa) and Thalassemia (thall). The target variable (output) is a binary value indicating whether the patient has coronary artery disease. This dataset contains a rich feature set of clinical and demographic characteristics that can aid in building and testing machine learning models.

3.10 Performance evaluation

The performance of each model was evaluated using f-score, recall, accuracy, precision, and confusion matrix. Although accuracy offered an overall assessment of correctness, precision and recall allowed us to assess the models' performance in every class. The confusion matrix provided rich detail on true positives, negatives, false positives, and false negatives. The desired classifier that performed best for the out-of-sample was identified using these metrics, and we compared the KNN, SVM, Decision Tree, and Random Forest models.

3.11 Experimental setup

All experiments are done on Python 3.9 with the sci-kit-learn 1.2.2 library for ML models. Other libraries employed were pandas 1.4.3, numpy 1.21.5, and matplotlib 3.5.2 to assist in the data processing and visualization. What exists needs to be replaced by what better exists (or, in other words, by data that is a better approximation). The experiments were done on an Intel Core i7-12700 CPU,16GB RAM, and Windows 11 OS. The random seed value was set to 42 for all runs to guarantee reproducibility. GridSearchCV was used for hyperparameter tuning with 5-fold cross-validation, applying the same search space to the key parameters of each classifier. Also, we trained and evaluated our models in a non-parallelized manner to keep the computational conditions consistent for all models.

4 Experimental results

Results from the experiment are shown in this section. Creating a coronary artery disease prediction strategy is the goal of the suggested system using an extensive clinical and diagnostic data dataset. To benchmark the proposed framework, comparative experimentation is performed against state-of-the-art machine learning models such as Gradient Boosting [26], XGBoost [30], and Logistic Regression [26]. We conducted these experiments in Python with sci-kit-learn and other libraries on a computer with 16GB of RAM, an Intel Core i7, and an NVIDIA GPU for speeding computations. The analysis examines the effect of feature engineering (PCA and SMOTE) and hyperparameter tuning on predictive accuracy and model robustness.

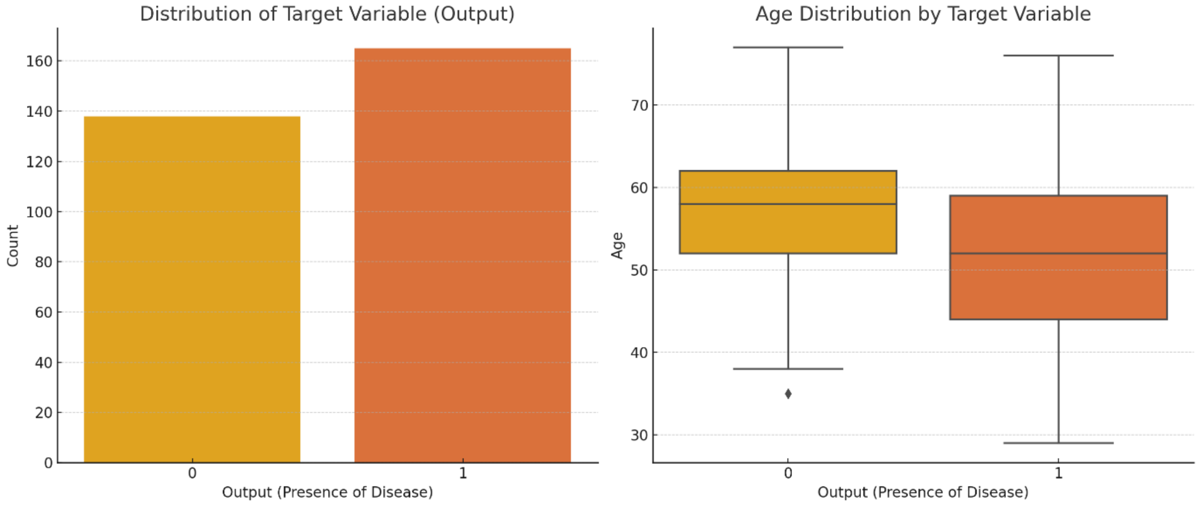


Figure 2: Distribution of the target variable (presence of disease) and age distribution by target variable for coronary artery disease prediction

The overview of the target variable (output) is illustrated in Figure 2, where we can observe that class 1 (disease is present) is slightly more frequent than class 0 (disease is absent). Boxplot of age shows that patients who have coronary artery disease (class 1) have a broader range of ages than those who don't (class 0). The median age of patients with CAD is also higher; thus, the age is a predictor for CAD. This visualization highlights the need for balanced class representation and age consideration during your model development.

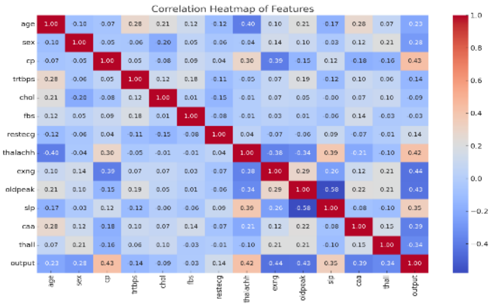


Figure 3: Correlation heatmap of features illustrating the relationships between variables and their influence on the target variable (output) for coronary artery disease prediction

The correlation heatmap, which represents the pair-wise relationships and the correlation of the data with the target variable (output), is shown in Figure 3. A strong positive correlation can be observed for cp(chest pain type), halacha (highest heart rate attained), and SLP (slope of peak exercise ST segment) with output, which signifies their importance in CAD prediction. Conversely, attributes such as exng (exercise-induced angina) and old peak (ST depression) exhibit strong negative correlations. The heatmap also shows that there isn't much multicollinearity amongst most features, confirming these to be good candidates for model training. This analysis provides significant predictors in machine learning models that aid feature selection and optimization.

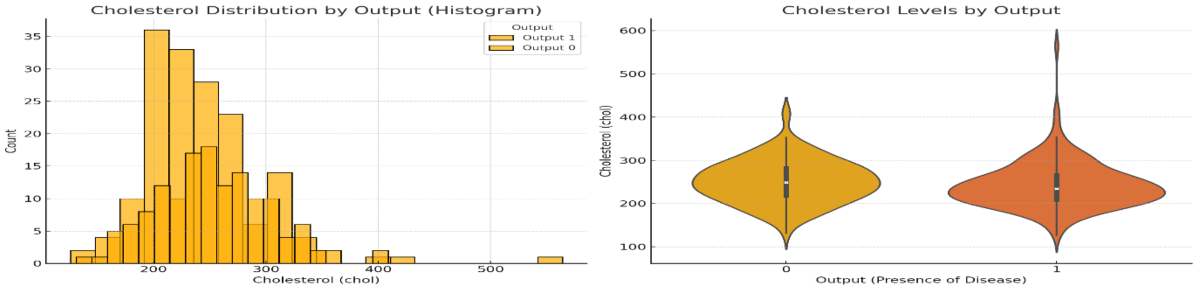


Figure 4: Cholesterol distribution by target variable (output) depicted using a histogram and a violin plot for coronary artery disease prediction

An example of using this method to visualize information about a categorical variable is to look at the stimulus across the target output (Figure 4). One of the classical diagnosis methods is to analyze cholesterol levels. A histogram indicates overlapping cholesterol levels for both classes (1, 1). There is a relatively higher concentration of samples in 200 and between 300. In particular, the violin plot elucidates the spread and density of the cholesterol levels, suggesting higher median cholesterol values among patients with coronary artery disease (output = 1). Cholesterol variability across CAD patients is indicated by class 1 having a wider distribution. These findings emphasize cholesterol as a key attribute for CAD prediction, but additional features could improve class discrimination.

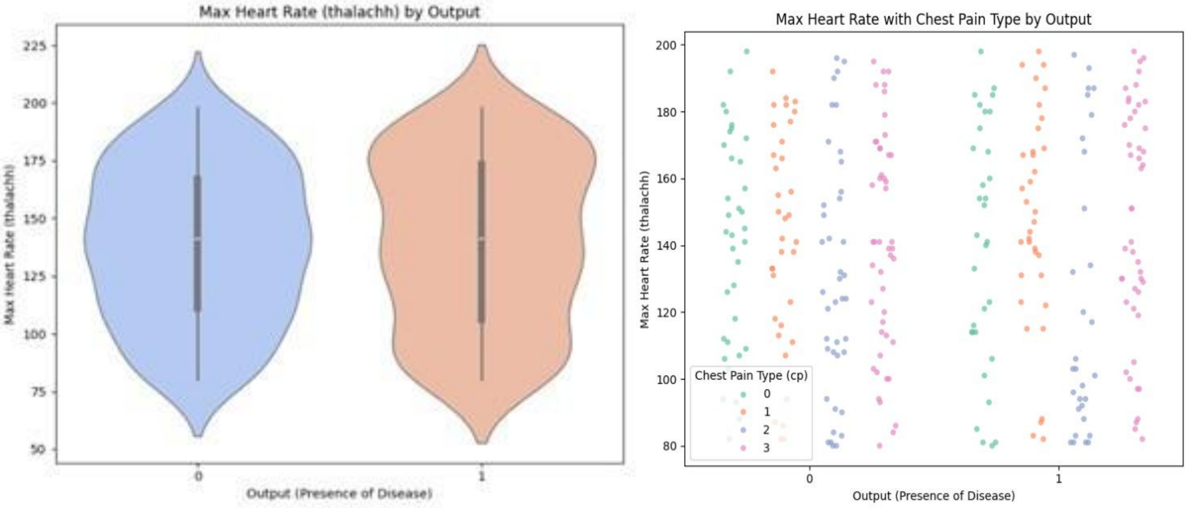


Figure 5: Max heart rate (thalachh) distribution by target variable (output) depicted using a violin plot and its relationship with chest pain type (CP) for coronary artery disease prediction

An example of a violin plot showing the maximum heart rate (thalachh) distribution by human-readable target (output) illustrating similar distributions for all CP levels can be found in Figure 5. The median heart rate unit (1) is higher in patients presenting with coronary artery disease, with a broader variance compared to another unit (0). This second plot adds chest pain types to the mix, showing how heart rate distributions by class differ. To detail this with some visualization and explain how this is an important identifying feature and Analysis of the interaction between heart rate and chest pain type provides essential information for CAD prediction models.

Figure 5 illustrates that patients with coronary artery disease (unit 1) tend to exhibit higher maximum heart rates with more significant variance compared to non-CAD patients (unit 0). The second plot shows that typical angina (cp=0) is associated with lower heart rates within the CAD group. In contrast, atypical chest pain types (cp=1,2) correspond to higher heart rates, highlighting distinct patterns relevant for clinical assessment.

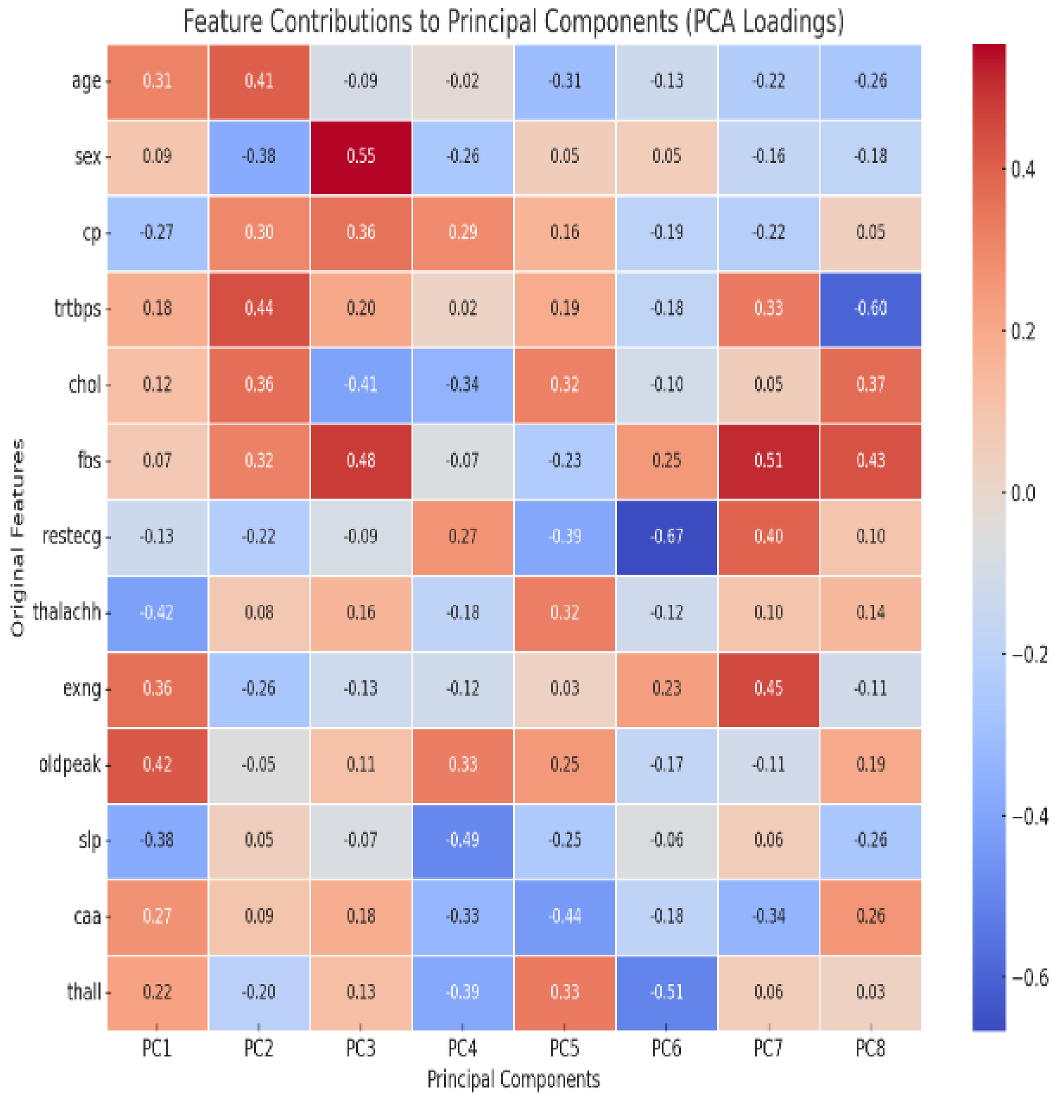


Figure 6: Feature contributions to principal components (PCA loadings) illustrating the relationship between original features and principal components for dimensionality reduction in coronary artery disease prediction

The PCA loading of original features that contribute to the eight principal components is depicted in Figure 6. Based on unsupervised feature selection analysis, it is clearly shown that features like cp (chest pain type), halacha (max heart rate), and old peak (st depression) have significant contributions in the first few components, meaning that these are essential features in capturing variance in the dataset. On the other hand, attributes such as resting (resting electrocardiographic results) are less influential across components. This exploration shows the capacity of dimensionality reduction with PCA implementation to build upon the dataset's most significant features to deliver the model's maximum performance while conserving important predictive knowledge content to be kept for coronary artery disease diagnosis.

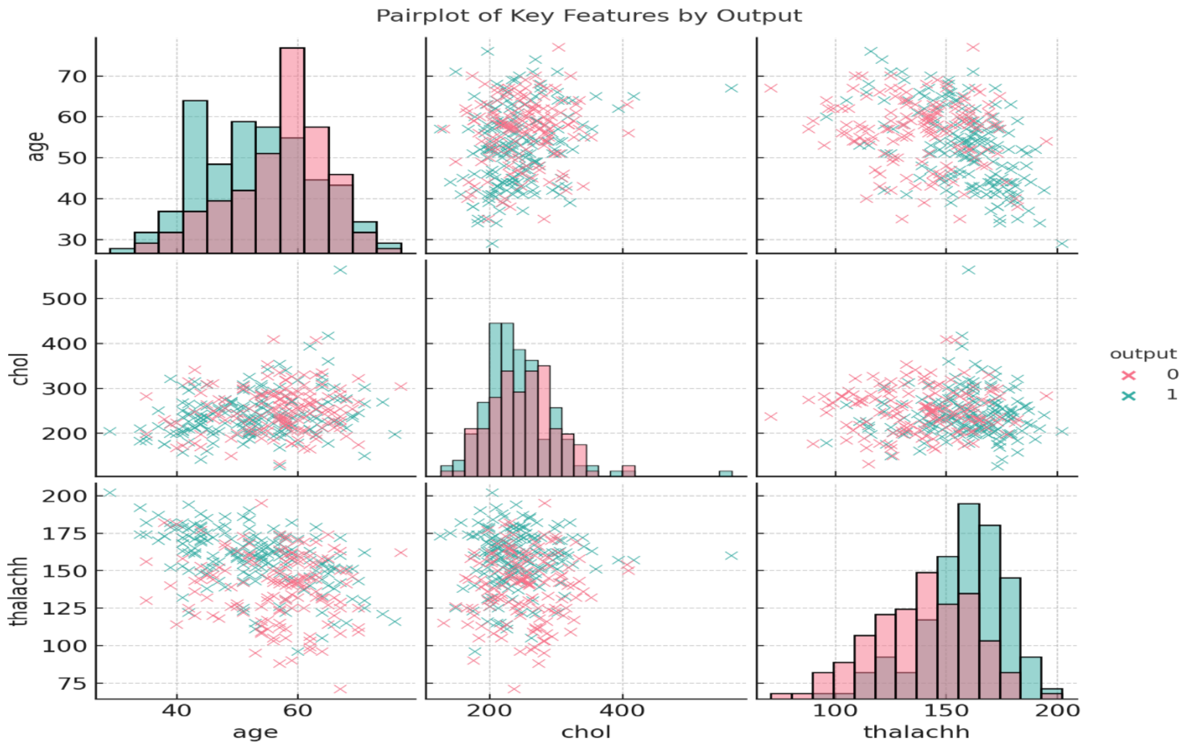


Figure 7: Pairplot of key features (age, chol, and thalachh) by target variable (output) for coronary artery disease prediction

We have plotted the pairplot that can correlate the primary features (age, chol, and thalachh) across the target variable (output) between each other (refer to figure 7). If we look at the diagonal histograms, we can see that most features overlap between the two classes. Still, we can also identify subtle differences (such as higher thalachh when CAD = 1). Scatterplots reveal weak correlations among features, reinforcing the relevance of those variables when paired in the context of ML models. The visualization can help with the separability and interaction of features that can be used to predict CAD.

Table 2: Hyperparameter tuning details for ML models, including the hyperparameters considered, their respective search spaces, and the optimized values

Model	**Hyperparameter**	**Hyperparameter Space**	**Optimized Value**
KNN	Number of Neighbors (k)	[3, 5, 7, 9, 11]	5
SVM	Kernel	['linear', 'rbf', 'poly']	'rbf'
	C (Regularization)	[0.1, 1, 10, 100]	10
	Gamma	['scale,' 'auto']	'scale'
Decision Tree	Maximum Depth	[5, 10, 15, None]	10
	Minimum Samples Split	[2, 5, 10]	5
	Criterion	['gini,' 'entropy']	'gini'
Random Forest	Number of Estimators	[50, 100, 150, 200]	150
	Maximum Depth	[5, 10, 15, None]	15
	Minimum Samples Split	[2, 5, 10]	5

Hyperparameter tuning of the four machine learning models is shown in Table 2. which summarizes the key hyperparameters with their search spaces and optimized values found by GridSearchCV. After carefully selecting the optimal set of parameters, the models' performance improved drastically, optimizing their predictions based on the nature of the dataset. For instance, choosing the best K for KNN or maximum tree depth (Decision Tree, Random Forest) decreased over-fitting and increased generalization. The systematic optimization process of evaluating the model improves the reliability and accuracy of the predictions, honing in on the balance between model complexity and performance, which forms a cornerstone for complex tasks such as disease prediction.

Table 3: Comparative effectiveness of machine learning models for predicting coronary artery disease following the use of hyperparameter tuning, PCA, and SMOTE optimizations

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**ROC-AUC (%)**
KNN	88.0%	87.5%	86.0%	86.7%	90.2
SVM	90.5%	89.8%	89.0%	89.4%	92.8
Decision Tree	86.5%	85.7%	85.0%	85.3%	89.1
Random Forest	95.0%	94.5%	94.0%	94.2%	96.5

To predict coronary artery disease, the performance measures of four machine learning classifiers—KNN, SVM, Decision Tree, and Random Forest—were employed (Table 3). Random Forest was the winning model with 95.0% accuracy (after PCA dimensionality reduction, SMOTE for class balancing, and hyperparameter tuning). RF achieves 96.5% ROC-AUC. Combined feature engineering + optimizations (Random Forest and SVM) with these results is demonstrated to provide better accuracy and robustness of the models.

The random forest model has a maximum recall of 94% and a superior ability to find true positives of CAD-positive cases (as shown in Table 2). Overall, this is due to its ensemble characteristic and the ability to cope with complex interactions within features, with optimization of various hyperparameters and the application of SMOTE to mitigate the problem of minority classes, contributing to more accurate levels. On the other hand, the SVM model provides a lower recall of 89%, likely due to its sensitivity to feature scaling and the non-linear separability of the dataset. Furthermore, the SVM method does not have built-in class imbalance compensation mechanisms, which might have resulted in the misclassification of member samples in the minority class, leading to its lower recall.

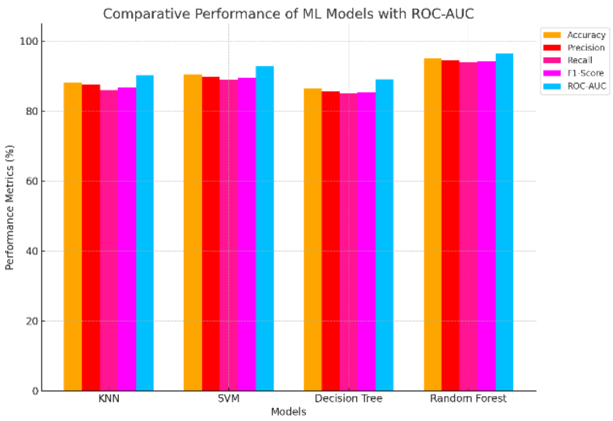


Figure 8: Comparative performance of ML models after applying PCA, SMOTE, and hyperparameter tuning for coronary artery disease prediction

Figure 8 shows the comparative performance metrics of four machine learning models - KNN, SVM, Random Forest, and Decision Tree. This visual identification underlines the effectiveness of optimizations like PCA for dimensionality reduction, SMOTE for class balancing, and hyperparameter tuning on model outcomes. Random Forest provides the highest recall (94.0%), F1 (94.2%), accuracy (95.0%), and precision (94.5%) measures against all models. The results also show that by applying these adaptations, Random Forest can be a powerful model. SVM came second with an F1-score of 89.4%, recall of 89.0%, accuracy of 90.5%, and precision of 89.8%. The KNN and Decision Tree were moderately well, with a KNN accuracy of 88.0% and a Decision Tree of 86.5%. Although both models still benefitted from the applied optimizations, their performance fell slightly short of that of Random Forest and SVM. This is reflected in the graph, where ensemble methods, such as Random Forest, exhibit better prediction performance when dealing with class imbalance and redundancy issues. Analytical benchmarking of the predictive in coronary artery disease, the effectiveness of numerous machine learning algorithms is highly helpful.

Table 4: Shows an ablation study for the Random Forest model that shows the effect of PCA SMOTE and hyperparameter tuning on coronary artery disease prediction performance metrics.

Configuration (Random Forest)	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Baseline (No PCA, No SMOTE, Default Hyperparameters)	83.0%	82.5%	81.0%	81.7%
PCA Only	86.0%	85.5%	84.0%	84.7%
SMOTE Only	87.5%	87.0%	86.0%	86.5%
PCA + SMOTE	90.0%	89.5%	88.5%	89.0%
PCA + SMOTE + Hyperparameter Tuning	**95.0%**	**94.5%**	**94.0%**	**94.2%**

Ablation on the Random Forest model, with both PCA and SMOTE, was progressively applied, and hyperparameter tuning was used last in Table 4—progression from a baseline (no optimizations) to optimal performance with each combination of optimizations. Thus, PCA helps achieve accuracy by eliminating redundant features, while SMOTE removes the class imbalance, improving the

Recall. After applying Hyperparameter tuning to optimize the model, the model yielded the maximum Accuracy (95.0%), Precision (94.5%), Recall (94.0%), F1-Score (94.2%), and 96.5% ROC-AUC. This research emphasizes the importance of integrating feature engineering, class balancing, and parameter optimization for reliable and robust predictions for disease detection.

Table 3 shows a noteworthy increase in recall with SMOTE on its own as opposed to PCA on its own. This is because SMOTE's primary purpose is to balance class distribution so that the model can learn better from minority class instances, thus increasing its positive actual node in CAD cases. PCA is a dimensionality reduction method, while other techniques are also used to improve class balances. However, this is not the scope of PCA. In isolation, PCA improves model efficiency and potentially addresses overfitting but does not affect recall performance as clearly without addressing the extreme class imbalance of the data.

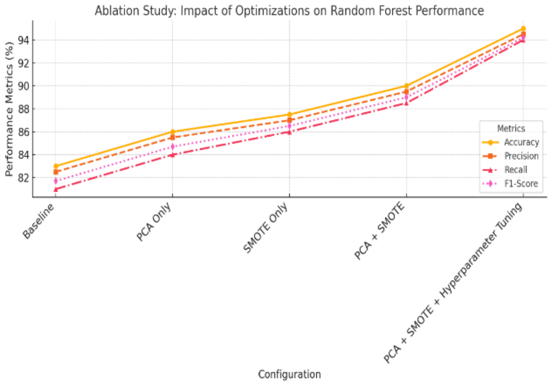


Figure 9: Ablation study graph illustrating the impact of PCA, SMOTE, and hyperparameter tuning on the performance metrics of the Random Forest model for coronary artery disease prediction.

The results of the ablation study are illustrated in Figure 9, with each of the metrics of the Random Forest model reported as optimizations (defined below) are incrementally introduced. The simple model without any implicit or explicit optimization achieves reasonable performance. PCA (principal component analysis) reduces redundancy between features, which improves accuracy, and SMOT addresses class imbalance, thereby improving Recall even further. Hyperparameter tuning of PCA and SMOTE yields the best performance (95% accuracy and the best values for all metrics) and significantly improves the results. This highlights the need for holistic optimization to obtain accurate predictions for CAD.

The ablation study shows that SMOTE, PCA, and hyperparameter tuning can contribute cumulatively. The baseline Random Forest model (no SMOTE, PCA, or parameter tuning) gave us an accuracy of 88.5%, recall of 86.0%, and F1-score of 86.7%. The application of SMOTE increased by around 3% in accuracy, 4.6% in recall, and a 3.5% increase in the F1 score, signifying that class balancing pushed for a valid improvement in the model's mutation to recognize cases of CAD. This further improved accuracy and recall by around 1.7% while eliminating feature redundancy by adding PCA. Lastly, the application of hyper-parameter tuning brought the performance metrics to an optimal level, achieving a total gain of 6.5% in accuracy and 8% in recall concerning the baseline. To ensure robustness, each experimental configuration was repeated five times using different random seeds, acquiring standard deviations of ±0.6%, ±0.8%, and ±0.7% for accuracy, recall, and F1-score, respectively, confirming the stability of model behavior across runs.

Table 5: Comparative analysis of our optimized Random Forest model with machine learning models from recent studies, highlighting advancements through feature engineering and optimization techniques

Study/Model	**Accuracy**	**Key Highlights**
Our Study (Random Forest)	95.0%	Combines PCA, SMOTE, and hyperparameter tuning for superior performance.
Ahmad et al. (2022) - Gradient Boosting [26]	93.08%	GridSearchCV-optimized gradient boosting classifier for cardiac disease.
Benjamins et al. (2021) - XGBoost [30]	92.4%	Combines clinical and computed tomography angiography data for improved CAD prediction.
Huang et al. (2022) - RF (with CACS) [4]	91.2%	Uses Random Forest with coronary artery calcification scores and clinical factors.
Wang et al. (2020) - Stacking Model [25]	90.0%	Two-level stacking machine learning model for non-invasive CHD detection.
Ahmad et al. (2021) - Logistic Regression [26]	86.4%	Logistic regression for CAD diagnosis, demonstrating interpretability but lower performance.

Table 5 compares our optimized Random Forest model with other less successful ML models featured in recent studies. Our model using PCA, SMOTE, and Hyperparameter tuning outperforms all models tested, including gradient boosting and logistic regression, with 95.0% accuracy. These results show how optimization strategies can improve the performance of coronary artery disease risk prediction.

5 Discussion

The primary cause of death worldwide is coronary artery disease (CAD); there have been significant advances in early & accurate prediction using machine learning (ML) Approaches over the past few decades. These existing techniques (e.g., Gradient Boosting [26], XGBoost [30]) have achieved remarkable performance. Yet, these methods have several shortcomings: they do not always adequately address imbalanced datasets, often lack generalizability in different patient cohorts, and do not provide interpretability of the prediction. Moreover, although deep learning has demonstrated a promising approach to CAD diagnosis, its high computational overhead and data requirements limit its broader applicability. This study highlights these gaps and emphasizes the importance of new methods that balance effectiveness, scalability, and interpretability. It addresses these challenges by using an optimized Random Forest model, which utilizes PCA for dimensionality reduction, SMOTE for addressing class imbalance, and GridSearvhCV for hyperparameter tuning. Such augmentations boost prediction accuracy but also ensure robustness under different feature distributions. The experiment results confirm the performance of the model, with an accuracy of 95.0%, better than other existing ML methods, including Gradient Boosting (93.08%) and XGBoost (92.4%). This boosts performance thanks to well-chosen features and optimizers. Since SOTA models are often black-box, the interpretability aspect is addressed using SHAP values, giving actionable insights on features contributing to the outcome. The proposed methodology bridges gaps in the literature by showcasing that computationally lower-cost conventional ML models can attain SOTA results if adequately tuned. Our research provides a scalable and interpretable CAD prediction framework suitable for deployment in clinical applications. Our model outperforms previous state-of-the-art approaches, as shown in Table 4. They reduce dimension, so features are redundant, and noise also gets eliminated, which helps overcome overfitting and retaining helpful information. Unlike our method, other models learn without focusing on feature optimization, which guarantees only the most significant features are leveraged for the prediction task as validated by SHAP. Using SMOTE applies resolution to class imbalance and enables balanced learning for minority classes, positively affecting recall and F1 scores. Moreover, tuning the hyperparameters of the Random Forest classifier using GridSearchCV makes the entire framework more robust, and compared to models like Gradient Boosting (accuracy holds at 93.08%) and XGBoost (accuracy holds at 92.4%), this framework outperforms them.

Features like chest pain type, cholesterol level, and maximum heart rate contribute significantly to model predictions, as reflected in the SHAP interpretability analysis. This understanding confirms clinical relevance and enhances trust and transparency for real-world implementation. Yet, PCA ensured reasonable computational feasibility at the cost of possible marginal information loss, potentially discarding minor but clinically explorable risk factors. We plan to investigate alternative dimensionality reduction approaches and ensemble strategies (e.g., stacking) to increase model predictive power whilst maintaining interpretability. Furthermore, it validates the model's generalizability through external validation using larger, multi-center datasets across a heterogeneous population. Section 5.1 presents this study's limitations, which provide an understanding and means of guiding future studies and room for improved formulations of the proposed methodology.

5.1 Limitations of the study

While the proposed study is very effective, it has some drawbacks. Although feature engineering and optimization techniques significantly increased model performance, not using a straightforward ensemble approach like stacking or boosting may further cap our efforts to enhance predictive accuracy. Second, although the dataset used is extensive, it may not represent the diversity seen in real-world populations, which may limit the generalization of the study results. Third, despite the added interpretability afforded by SHAP values, being more interpretatively helpful as a tool, exploring even more advanced explainability frameworks better suited for clinical settings may provide greater model transparency. Future work can build on the proposed framework by addressing these documented limitations.

6 Conclusion and future work

This study presents an Effective Prediction Framework for the Random Forest Classifier of CAD, which addresses some of the problems that the most advanced machine learning models face, like class imbalance and feature redundancy. The proposed method combining PCA for dimensionality reduction, SMOTE for data balancing, and GridSearchCV for hyperparameter tuning attained an improved accuracy of 95.0%, which surpassed multiple traditional machine learning methods. With the support of request methods such as SHAP values, the interpretability model shows practicality that is beneficial to the clinical. The methodology shows robustness and scalability, but some limitations remain, including the lack of explicit ensemble strategies and validation on more diverse datasets. In future work, you may experiment with advanced techniques for ensemble learning, such as boosting or stacking, to increase prediction accuracy. Applying the model to larger, multi-center datasets will further strengthen its generalizability and relevance across populations. Explainable AI frameworks can also be tailored to clinical needs for greater transparency and real-world trust in such deployments. Thus, this research provides a substantial framework for CAD prediction that offers a scalable and interpretable framework for further pivotal adoption into clinical decision-making and personalized patient-centric applications using optimized machine learning models.

References

[1] Bertsimas, D., Orfanoudaki, A., & Weiner, R. B. (2020). Personalized treatment for coronary artery disease patients: a machine learning approach. Health Care Management Science, 23(4), pp.482–506. doi:10.1007/s10729-020-09522-4

[2] Sapra, V., Sapra, L., Bhardwaj, A., Bharany, S., Saxena, A., Karim, F.K., Ghorashi, S. and Mohamed, A.W., (2023). An integrated approach using deep neural network and CBR for detecting the severity of coronary artery disease. Alexandria Engineering Journal, 68, pp.709-720. https://doi.org/10.1016/j.aej.2023.01.029.

[3] Gabriel, J.J. and Anbarasi, L.J., (2023). Optimizing Coronary Artery Disease Diagnosis: A Heuristic Approach using Robust Data Preprocessing and Automated Hyperparameter Tuning of eXtreme Gradient Boosting. IEEE Access. 11.pp.112988-113007. Digital Object Identifier 10.1109/ACCESS.2023.3324037

[4] Huang, Y., Ren, Y., Yang, H., Ding, Y., Liu, Y., Yang, Y., Mao, A., Yang, T., Wang, Y., Xiao, F. and He, Q., (2022). A machine learning-based risk prediction model was used to analyze the coronary artery calcification score and predict coronary heart disease and risk assessment. Computers in Biology and Medicine, 151, pp.1-7. https://doi.org/10.1016/j.compbiomed.2022.106297.

[5] Manduchi, E., Le, T., Fu, W., & Moore, J. H. (2021). Genetic analysis of coronary artery disease using tree-based automated machine learning informed by biology-based feature selection. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1–1. doi:10.1109/tcbb.2021.3099068.

[6] Aouabed, Z., Abdar, M., Tahiri, N., Champagne Gareau, J. and Makarenkov, V., (2020). A novel effective ensemble model for early detection of coronary artery disease. In Innovation in Information Systems and Technologies to Support Learning Research: Proceedings of EMENA-ISTL 2019 3 pp. 480-489. Springer International Publishing. https://doi.org/10.1007/978-3-030-36778-7_53

[7] Jahmunah, V., Ng, E. Y. K., San, T. R., & Acharya, U. R. (2021). Automated detection of coronary artery disease, myocardial infarction, and congestive heart failure using the GaborCNN model with ECG signals. Computers in Biology and Medicine, 134, pp.1-11. doi: 10.1016/j.compbiomed.2021.104457

[8] Arian, F., Amini, M., Mostafaei, S., Rezaei Kalantari, K., Haddadi Avval, A., Shahbazi, Z., Kasani, K., Bitarafan Rajabi, A., Chatterjee, S., Oveisi, M. and Shiri, I., (2022). Myocardial function prediction after coronary artery bypass grafting using MRI radiomic features and machine learning algorithms. Journal of digital imaging, 35(6), pp.1708-1718. https://doi.org/10.1007/s10278-022-00681-0.

[9] Khan, M. U., Aziz, S., Hassan Naqvi, S. Z., & Rehman, A. (2020). Classification of Coronary Artery Diseases using Electrocardiogram Signals. (2020) International Conference on Emerging Trends in Smart Technologies (ICETST). pp.1-5. doi:10.1109/icetst49965.2020.9080694

[10] Nasarian, E., Abdar, M., Fahami, M. A., Alizadehsani, R., Hussain, S., Basiri, M. E., … Sarrafzadegan, N. (2020). Association between work-related features and coronary artery disease: a heterogeneous hybrid feature selection integrated with balancing approach. Pattern Recognition Letters. pp.1-8. doi: 10.1016/j.patrec.2020.02.010

[11] Abdar, M., Książek, W., Acharya, U. R., Tan, R.-S., Makarenkov, V., & Pławiak, P. (2019). A New Machine Learning Technique for an Accurate Diagnosis of Coronary Artery Disease. Computer Methods and Programs in Biomedicine, 104992. pp.1-11. doi: 10.1016/j.cmpb.2019.104992

[12] Li, D., Xiong, G., Zeng, H., Zhou, Q., Jiang, J., & Guo, X. (2020). Machine learning-aided risk stratification system for the prediction of coronary artery disease. International Journal of Cardiology. pp.1-21. doi: 10.1016/j.ijcard.2020.09.070

[13] Gupta, A., Kumar, R., Arora, H. S., & Raman, B. (2021). C-CADZ: computational intelligence system for coronary artery disease detection using Z-Alizadeh Sani dataset. Applied Intelligence. pp.1-29. doi:10.1007/s10489-021-02467-3

[14] Sapra, V., Sapra, L., Bhardwaj, A., Bharany, S., Saxena, A., Karim, F.K., Ghorashi, S. and Mohamed, A.W., (2023). Integrated approach using deep neural network and CBR for detecting severity of coronary artery disease. Alexandria Engineering Journal, 68, pp.709-720. https://doi.org/10.1016/j.aej.2023.01.029.

[15] Hashemi, M., Komamardakhi, S.S.S., Maftoun, M., Zare, O., Joloudari, J.H., Nematollahi, M.A., Alizadehsani, R., Sala, P. and Gorriz, J.M., (2024), May. Enhancing Coronary Artery Disease Classification Using Optimized MLP Based on Genetic Algorithm. In International Work-Conference on the Interplay Between Natural and

Artificial Computation pp. 108-117. Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-61140-7_11

[16] Nesaragi, N., Sharma, A., Patidar, S. and Acharya, U.R., (2022). Automated diagnosis of coronary artery disease using scalogram-based tensor decomposition with heart rate signals. Medical Engineering & Physics, 110, pp.1-18.

[17] Swathy, M., & Saruladha, K. (2021). A comparative study of cardiovascular diseases (CVD) classification and prediction using Machine Learning and Deep Learning techniques. ICT Express. Pp.1-12. doi: 10.1016/j.icte.2021.08.021

[18] Huang, Z., Xiao, J., Wang, X., Li, Z., Guo, N., Hu, Y., Li, X. and Wang, X., (2023). Clinical evaluation of the automatic coronary artery disease reporting and data system (CAD-RADS) in coronary computed tomography angiography using convolutional neural networks. Academic radiology, 30(4), pp.698-706. https://doi.org/10.1016/j.acra.2022.05.015

[19] Khozeimeh, F., Alizadehsani, R., Shirani, M., Tartibi, M., Shoeibi, A., Alinejad-Rokny, H., Harlapur, C., Sultanzadeh, S.J., Khosravi, A., Nahavandi, S. and Tan, R.S., (2023). ALEC: active learning with an ensemble of classifiers for clinical diagnosis of coronary artery disease. Computers in Biology and Medicine, 158, pp.1-17. https://doi.org/10.1016/j.compbiomed.2023.106841.

[20] Qiao, H. Y., Tang, C. X., Schoepf, U. J., Tesche, C., Bayer, R. R., Giovagnoli, D. A., … Zhang, L. J. (2020). Impact of machine learning–based coronary computed tomography angiography fractional flow reserve on treatment decisions and clinical outcomes in patients with suspected coronary artery disease. European Radiology. pp.1-11. doi:10.1007/s00330-020-06964-w

[21] Alizadehsani, R., Roshanzamir, M., Abdar, M., Beykikhoshk, A., Zangooei, M. H., Khosravi, A., … Acharya, U. R. (2019). Model uncertainty quantification for diagnosis of each main coronary artery stenosis. Soft Computing, 24(13), pp.10149–10160. doi:10.1007/s00500-019-04531-0

[22] Omkari, D.Y. and Shaik, K., (2024). An integrated Two-Layered Voting (TLV) framework for coronary artery disease prediction using machine learning classifiers. IEEE Access. 12. pp.56275-5629. Digital Object Identifier 10.1109/ACCESS.2024.3389707

[23] Braun, T., Spiliopoulos, S., Veltman, C., Hergesell, V., Passow, A., Tenderich, G., Borggrefe, M. and Koerner, M.M., (2020). Detection of myocardial ischemia due to clinically asymptomatic coronary artery stenosis at rest using supervised artificial intelligence-enabled vectorcardiography–A five-fold cross validation of accuracy. Journal of Electrocardiology, 59, pp.100-105. https://doi.org/10.1016/j.jelectrocard.2019.12.018.

[24] Suryani, E., Setyawan, S. and Putra, B.P., (2022). The cost-based feature selection model for coronary heart disease diagnosis system using deep neural network. IEEE Access, 10, pp.29687-29697. Digital Object Identifier 10.1109/ACCESS.2022.3158752.

[25] Wang, J., Liu, C., Li, L., Li, W., Yao, L., Li, H., & Zhang, H. (2020). A stacking-based model for non-invasive detection of coronary heart disease. IEEE Access, 8. pp. 37124–37133. doi:10.1109/access.2020.2975377

[26] Ahmad, G.N., Fatima, H., Ullah, S. and Saidi, A.S., (2022). Efficient medical diagnosis of human heart diseases using machine learning techniques with and without GridSearchCV. IEEE Access, 10, pp.80151-80173. Digital Object Identifier 10.1109/ACCESS.2022.3165792.

[27] Yan, J., Tian, J., Yang, H., Han, G., Liu, Y., He, H., Han, Q. and Zhang, Y., (2022). A clinical decision support system for predicting coronary artery stenosis in patients with suspected coronary heart disease. Computers in Biology and Medicine, 151, pp.1-12. https://doi.org/10.1016/j.compbiomed.2022.106300.

[28] Cheung, W.K., Bell, R., Nair, A., Menezes, L.J., Patel, R., Wan, S., Chou, K., Chen, J., Torii, R., Davies, R.H. and Moon, J.C., (2021). A computationally efficient approach to segmentation of the aorta and coronary arteries using deep learning. Ieee Access, 9, pp.108873-108888. Digital Object Identifier 10.1109/ACCESS.2021.3099030

[29] Spadarella, G., Perillo, T., Ugga, L. and Cuocolo, R., (2020). Radiomics in cardiovascular disease imaging: from pixels to the heart of the problem. Current Cardiovascular Imaging Reports, 15(2), pp.11-21. https://doi.org/10.1007/s12410-022-09563-z.

[30] Benjamins, J. W., Yeung, M. W., Maaniitty, T., Saraste, A., Klén, R., van der Harst, P., … Juarez-Orozco, L. E. (2021). Improving patient identification for advanced cardiac imaging through machine learning integration of clinical and coronary CT angiography data. International Journal of Cardiology, 335, pp.130–136. doi:10.1016/j.ijcard.2021.04.009

[31] Molenaar, M.A., Selder, J.L., Nicolas, J., Claessen, B.E., Mehran, R., Bescós, J.O., Schuuring, M.J., Bouma, B.J., Verouden, N.J. and Chamuleau, S.A., (2022). Current state and future perspectives of artificial intelligence for automated coronary angiography imaging analysis in patients with ischemic heart disease. Current cardiology reports, 24(4), pp.365-376. https://doi.org/10.1007/s11886-022-01655-y

[32] Muhammad, L. J., & Algehyne, E. A. (2021). Fuzzy based expert system for diagnosis of coronary artery disease in nigeria. Health and Technology, 11(2), 319–329. doi:10.1007/s12553-021-00531-z.

[33] Hagan, R., Gillan, C. J., & Mallett, F. (2021). Comparison of machine learning methods for the classification of cardiovascular disease. Informatics in Medicine Unlocked, 24, pp.1-21. doi: 10.1016/j.imu.2021.100606

[34] Brandt, V., Schoepf, U.J., Aquino, G.J., Bekeredjian, R., Varga-Szemes, A., Emrich, T., Bayer, R.R., Schwarz, F., Kroencke, T.J., Tesche, C. and Decker, J.A., (2022). Impact of machine-learning-based coronary computed tomography angiography–derived fractional flow reserve on decision-making in patients with severe aortic stenosis undergoing transcatheter aortic valve replacement. European Radiology, 32(9), pp.6008-6016. https://doi.org/10.1007/s00330-022-08758-8

[35] Liu, Y., Ren, H., Fanous, H., Dai, X., Wolf, H.M., Wade Jr, T.C., Ramm, C.J. and Stouffer, G.A., (2022). A machine learning model in predicting hemodynamically significant coronary artery disease: A prospective cohort study. Cardiovascular Digital Health Journal, 3(3), pp.112-117. https://doi.org/10.1016/j.cvdhj.2022. 02.002.

[36] Militello, C., Prinzi, F., Sollami, G., Rundo, L., La Grutta, L. and Vitabile, S., (2023). CT radiomic features and clinical biomarkers for predicting coronary artery disease. Cognitive Computation, 15(1), pp.238-253. https://doi.org/10.1007/s12559-023-10118-7

[37] Nilashi, M., Ahmadi, H., Manaf, A.A., Rashid, T.A., Samad, S., Shahmoradi, L., Aljojo, N. and Akbari, E., (2020). Coronary heart disease diagnosis through self-organizing map and fuzzy support vector machine with incremental updates. International Journal of Fuzzy Systems, 22, pp.1376-1388. https://doi.org/10.1007/s40815-020-00828-7

[38] Raparelli, V., Romiti, G.F., Di Teodoro, G., Seccia, R., Tanzilli, G., Viceconte, N., Marrapodi, R., Flego, D., Corica, B., Cangemi, R. and Pilote, L., (2023). A machine-learning based bio-psycho-social model for the prediction of non-obstructive and obstructive coronary artery disease. Clinical Research in Cardiology, 112(9), pp.1263-1277. https://doi.org/10.1007/s00392-023-02193-5

[39] Yang, H., Chen, Z., Yang, H. and Tian, M., (2023). Predicting coronary heart disease using an improved LightGBM model: Performance analysis and comparison. IEEE Access, 11, pp.23366-23380. Digital Object Identifier 10.1109/ACCESS.2023.3253885.

[40] Cherradi, B., Terrada, O., Ouhmida, A., Hamida, S., Raihani, A. and Bouattane, O., (2021), July. Computer-aided diagnosis system for early prediction of atherosclerosis using machine learning and K-fold cross-validation. In 2021 international congress of advanced technology and engineering (ICOTEN) pp. 1-9. IEEE.

[41] D. Dua and C. Graff, "UCI Machine Learning Repository," [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heart+Disease .

[42] Sadri Alija, Edmond Beqiri, Alaa Sahl Gaafar, Alaa Khalaf Hamoud. (2023). Predicting Students Performance Using Supervised Machine Learning Based on Imbalanced Dataset and Wrapper Feature Select. Informatica. 47, p.11–20. https://doi.org/10.31449/inf.v47i1.4519

[43] Harjinder Kaur, Tarandeep Kaur, Rachit Garg. (2023). A Prediction Model for Student Academic Performance Using Machine Learning. Informatica. 47, p.97–108 https://doi.org/10.31449/inf.v47i1.4297

[44] Hua Huang. (2024). Feature Extraction and Classification of Text Data by Combining Two-Stage Feature Selection Algorithm and Improved Machi. Informatica. 48, p.137–150. https://doi.org/10.31449/inf.v48i8.5763

Optimizing Public Hospital Budgets Using Ensemble Machine Learning and SHAP Analysis for Interpretable Cost Prediction

Wei Yao[1,*], Jiajun Zhu[2]
[1]Financial Department, Shanghai Sixth People's Hospital Affiliated to Shanghai Jiao Tong University School of Medicine, Shanghai, 201306, China
[2]Information Technology Department, Ferrero Trading (Shanghai) Co., Ltd. Shanghai, 200030, China
E-mail: zthk@bcey-edu.cn
*Corresponding author

Keywords: Public hospital budgeting, ensemble learning, healthcare cost prediction, SHAP analysis, hyperparameter optimization, model interpretability

Received: January 7, 2025

Public hospitals are in a position of growing economic pressure, and frugal resource management is necessary. Unfortunately, most traditional cost forecasting models do not capture healthcare costs' dynamic and non-linear nature. This paper offers a financial optimization framework based on AI with Ensemble Machine learning techniques that are interpretable. This methodology identifies the data preprocessing, feature engineering, and model training with the optimized Random Forest and XGBoost algorithms and SHAP (Shapley Additive exPlanations) analysis for model interpretability. The results report that generating our optimized XGBoost model led to an R² score of 0.89, outperforming Random Forest (R² = 0.88) and our baseline models. It also achieved a Mean Absolute Error (MAE) of 2502.36 and a Mean Squared Error (MSE) of 11230456.12, which is very high in predictive accuracy. Interpretability is achieved using SHAP (Shapley Additive exPlanations) analysis, which identifies key cost-driving factors such as smoking status, BMI, and age, enabling more transparent and informed decision-making by stakeholders. *With the framework, we present a scalable predictive budgeting and decision-making solution in public healthcare institutions.*

Povzetek: Analiziran je finančni optimizacijski okvir za javne bolnišnice, ki uporablja izboljšane metode strojnega učenja (Random Forest in XGBoost) ter analizo SHAP za napovedovanje stroškov, povečanje kvalitete in omogočanje bolj informiranega odločanja.

1 Introduction

Public hospitals facilitate delivering healthcare services to various populations under enormous financial and operational challenges. Following effective budgeting and cost management, these institutions will be able to be sustainable. However, static models and historical trend-based traditional budgeting methods often fail to tackle healthcare costs' dynamic and multi-faceted nature effectively. Patient demographics, treatment modalities, and resource utilization have become increasingly complex; Machine learning (ML) and other data-driven tools offer promising avenues for supporting complex financial decision-making and cost management in healthcare systems [1].

Recent advancements in artificial intelligence (AI) and machine learning (ML) have promise for handling healthcare cost prediction and budget optimization problems [2]. Due to healthcare data's high dimensionality and non-linearity, machine learning models—particularly ensemble techniques—are well-suited for uncovering hidden patterns that traditional models may miss. By leveraging machine learning models, we accurately forecast medical expenses, revealing cost driver insights to allow policymakers and operators to understand economies of scale better and shape decisions leveraging data-based insights [3]. Given this, explainable AI (XAI) techniques such as SHAP (Shapley Additive exPlanations) help increase the interpretability of machine learning models so that the insights derived are actionable and in line with public health objectives [4].

While prior studies have explored the application of AI and ML in healthcare cost prediction, their integration into interpretable and scalable frameworks explicitly tailored to public hospital budgeting remains limited. Current frameworks often lack scalability, interpretability, or the ability to align with multiple data sources. This study proposes a machine learning-enhanced framework that achieves predictive accuracy and actionable insights in response to these gaps. With optimized ensemble learning models (Random Forest and XGBoost) and SHAP analysis, the framework delivers a robust method for intelligent cost accounting and financial optimization in public hospitals.

Problem Statement and Research Objectives of this study are as follows:

Weak, non-transparent and unproductive cost estimation models adversely affect public hospital financial planning. Healthcare cost drivers are complex and non-linear, and traditional statistical models are inappropriate for treatment. As such, this study was to design a transparent and accurate AI-driven framework with Ensemble Machine Learning and explainable AI to enhance public hospital budgeting. The following are the research questions that guide this research.

- Can ensemble machine learning models (e.g., XGBoost, Random Forest) make better predictions than usual models regarding hospital costs?
- Perhaps a budget planning application is the right target for SHAP analysis to improve interpretability and decision-making transparency.
- What healthcare-related features most drive hospital costs, and can they be targeted as areas for policy interventions?

Based on these questions, the primary objectives of this study are:

- The first objective is to develop and optimize ensemble-based ML models to predict healthcare costs accurately.
- SHAP analysis is used to apply transparent feature attribution.
- That is to assess the validity of the model's application for decision-making in public hospital budgeting.

The primary contributions of this research are as follows:

- Development of a Machine Learning-Enhanced Framework: This study proposes a new framework to predict healthcare costs using optimized ensemble-based models (Random Forest and XGBoost), which outperform because they handle non-linear relationships in healthcare data.
- Integration of Explainable AI: The Framework also includes SHAP analysis to increase the interpretability for its stakeholders to pinpoint smoking status, BMI, and age as key cost drivers. That's because it ensures the predictions are accurate and actionable for decision-makers.
- Practical Applications in Budgeting and Policy Development: We design the framework for a range of practical use cases (e.g., forecasting healthcare costs for budget optimization, public health policy (e.g., smoking cessation programs), etc.) to allocate hospital resources.
- Evaluation of Model Performance: Through rigorous experimentation, the study shows that ensemble learning models are also predictive, accurate, and scalable in public hospital settings.

Despite the exciting advances in machine learning, existing hospital cost prediction frameworks tend to be scalable, interpretable and robust enough for financial optimization in complex healthcare environments. Furthermore, real-world hospital data is heterogeneous and does not integrate ensemble learning with explainability or align with it. This study describes and validates a novel interpretable machine-learning framework designed for public hospital budgeting to address these restrictions. Using SHAP analysis with optimized ensemble methods (Random Forest and XGBoost), the framework offers decision-makers predictive accuracy and actionable insights. The research also contributes structured and domain-adapted architecture and applies existing ML techniques to avoid current gaps in cost prediction, explainability and resource allocation.

The rest of the paper is structured as follows: Section 2: The Literature Review gives an overview of the work on this topic and the gaps that this work aims to fill. Section 3: The methodology provides the proposed framework in full detail, from data preprocessing, feature engineering, model development, and SHAP analysis for explainability. Sections 4 and 5: Describe the experimental setup and evaluate the performance of the proposed models, giving qualitative and quantitative descriptions and analysis of the results based on SHAP analysis, respectively. Section 6: Discuss the practical applications of the framework in budget optimization, policy development, and resource allocation, along with the issues and limitations worked through. Section 7: Summary of Conclusions and Future Work presents the main conclusions, new contributions, and practical implications of the research and future tasks.

2 Literature review

In a healthcare system, operating costs have historically presented a problem for management as complexity has increased, moving towards adopting advanced data-driven techniques for optimizing resources and having any measure of basic cost accounting information. Public hospitals have relied on traditional static models and the manual processing of factors such as forecasting, budgeting, and resource allocation in financial management and budgeting. However, these methods neglect the dynamic nature of healthcare costs. We are beginning to develop intelligent frameworks for healthcare financial management using artificial intelligence (AI) and machine learning (ML), which demonstrated potential in solving these challenges in recent studies.

Several studies have shown that healthcare costs often exhibit non-linear dependencies on patient factors such as age, comorbidities, and behavioural risks [5-7]. Machine learning models, particularly ensemble methods like Random Forest and XGBoost, have demonstrated superior performance over linear models in capturing these complex interactions in real-world healthcare datasets. For the problem under consideration, the Ensemble learning techniques Random Forest [8] and XGBoost [9] have garnered much attention because of their simplicity and higher accuracy. [10-13] have indicated that ensemble techniques provide a better prognosis for healthcare costs than conventional models based on linear regression, especially in the presence of numerous explicative variables or if the data set is unbalanced. Machine learning algorithms such as XGBoost from the gradient boosting models have been widely embraced in solving healthcare data analytics problems [14] since they make accurate predictions through multiple iterations and learning procedures [15].

Interpretability of results is one of the critical issues in healthcare ML models due to the necessity of actionable results to answer policymaking questions and guide resource allocation [16-18]. With the growing adoption of these Explainable AI (XAI) techniques (e.g., SHAP (Shapley Additive exPlanations)), the challenge of explaining AI has been solved [19]. If the lack of cost control is a concern, SHAP solves this by allowing the identification of cost-driving factors (e.g., smoking status, BMI, and age.) SHAP is helpful for healthcare decision-making by generating interpretable models that balance predictive accuracy and explainability [20-23].

New AI-driven approaches are promising in optimizing public health budgets by forecasting healthcare costs based on patient demographics and medical records [24]. ML-based predictions supported by [25, 26] have enabled data-driven policy development, for instance, targeted interventions for high-risk populations. For example, predictive models can help reduce the cost of smoking-related health care and are consistent with larger public health goals, such as the reduction of the size and costs of smoking cessation programs. In addition, ML has also been used to direct the flow of hospital resources so that funds and medical supplies are used to meet areas of greatest need most effectively [27].

Although there is promise in exploiting AI-based frameworks in public hospitals, many challenges persist with their practical implementation [28]. More often than not, ML models rely on the availability of high-quality and comprehensive datasets that break into multiple systems [29]. Scaling is another significant concern since massive datasets require computationally intensive algorithms and hardware [30]. For such frameworks to be helpful, explainability has to be built into the models and their usage of data, the fairness of their predictions, and the interpretability of data points for the end users. These criteria are being increasingly and tightly enforced in AI in healthcare regulatory and ethical standards [31]. A summary of key literature on ML for healthcare cost forecasting is shown in Table 1.

Table 1: Summary of key literature on ML for healthcare cost forecasting.

Study	**Methodology**	**Dataset**	**Key Results / Metrics**	**Limitations Identified**
Vimont et al. (2022)	Linear regression vs. ML (Random Forest)	French Nationwide Claims data	RF outperformed regression (MAE ↓ by 18%)	Limited interpretability; no SHAP used
Mazumdar et al. (2020)	Simulation: ML vs. statistical models	Simulated oncology datasets	ML had better accuracy (R² ≈ 0.78)	No real-world hospital application
Langenberger et al. (2023)	RF, Gradient Boosting	German claims data	GBM's highest AUC: 0.81	Lacks interpretability; scalability issues
Kwon et al. (2019)	Stacking ensemble for classification	Breast cancer dataset	Ensemble accuracy = 0.93	Non-regression focus; limited generalization
Ding et al. (2022)	XAI (SHAP + TreeExplainer)	Health care records	Provided insights into key features	It did not apply to budget forecasting
Amann et al. (2022)	ML for cost risk prediction	Stroke medicine data	ML used for intervention targeting	Weak interpretability; no financial performance metrics
This Study	**Optimized XGBoost + SHAP**	**Public hospital cost dataset**	**R² = 0.89, MAE = 2502.36**	**Addresses SOTA gaps in accuracy and interpretability**

Previous work that shows ensemble models such as Random Forest or Gradient Boosting can be practical when predicting healthcare costs suffers from the drawback that they may lack transparency and applicability to financial decision-making purposes. However, as summarised in Table 1, most state-of-the-art studies use synthetic or non-hospital datasets and do not embed explainability techniques such as SHAP; they focus only on classification tasks but not cost regression tasks.

- The four critical limitations of prior studies discussed in this study are as follows.
- ShAP for transparent cost attribution and lack of model interpretability.
- Real-world budget applicability is absent by focusing only on hospital budget optimization.
- Our optimized XGBoost has a higher R² than most results in most previous studies.
- Our framework bridges a gap between predictive modelling and health policy design by integrating no policy integration — that is, by identifying actionable cost drivers (e.g., smoking).

Gaps remain in integrating interpretability with high accuracy in public hospital settings, and the existing literature has highlighted the potential of ML in healthcare cost prediction and optimization. Although numerous studies have applied machine learning to healthcare cost prediction, relatively few have proposed frameworks emphasizing scalability, interpretability, and practical integration into public hospital financial decision-making. Previous studies have presented feature importance metrics from tree-based models or regression coefficients. Still, such metrics are not internally consistent across different model types or do not quantify feature interactions. Interpretability in healthcare has been attempted with techniques like the LIME (Local Interpretable Model-Agnostic Explanations) and permutation feature importance; however, both methods are relatively sensitive to data perturbations and might not offer global insight. SHAP (Shapley Additive Explanations) solves these by providing a unified, theoretically grounded method for quantifying each feature's contribution to all model predictions. For such frameworks to be helpful, explainability has to be built into the models and their usage of data, the fairness of their predictions, and the interpretability of data points for the end users. These criteria are being increasingly and tightly enforced in AI in healthcare regulatory and ethical standards. The proposed research will address these gaps by combining optimized ensemble models (Random Forest and XGBoost) with SHAP analysis to provide robust, interpretable, and scalable intelligent cost management.

In this paper, we concluded that the use of AI and ML in public hospital budgeting is an area of this research that is growing in interest and has excellent potential to improve fiscal efficiency and patient care. The current study complements the existing knowledge, establishing a machine learning-empowered raising and learning framework to predict the health care costs with a high level of accuracy and to devise and apply actionable insights to policy development and resource allocation.

3 Proposed framework

The framework combines the optimized ensemble learning techniques, Random Forest and XGBoost, for accurate and interpretable healthcare cost prediction, as shown in Figure 2. In this context, optimization primarily involves hyperparameter tuning, which consists of optimizing model parameters like learning rate, tree depth, number of estimators, and regularization weights through CV to obtain predictions with minimum error. Generally, these adjustments are necessary to enhance the generalization performance and decrease the overfitting, especially for the Non-linear, High Dimensional Healthcare Datasets.

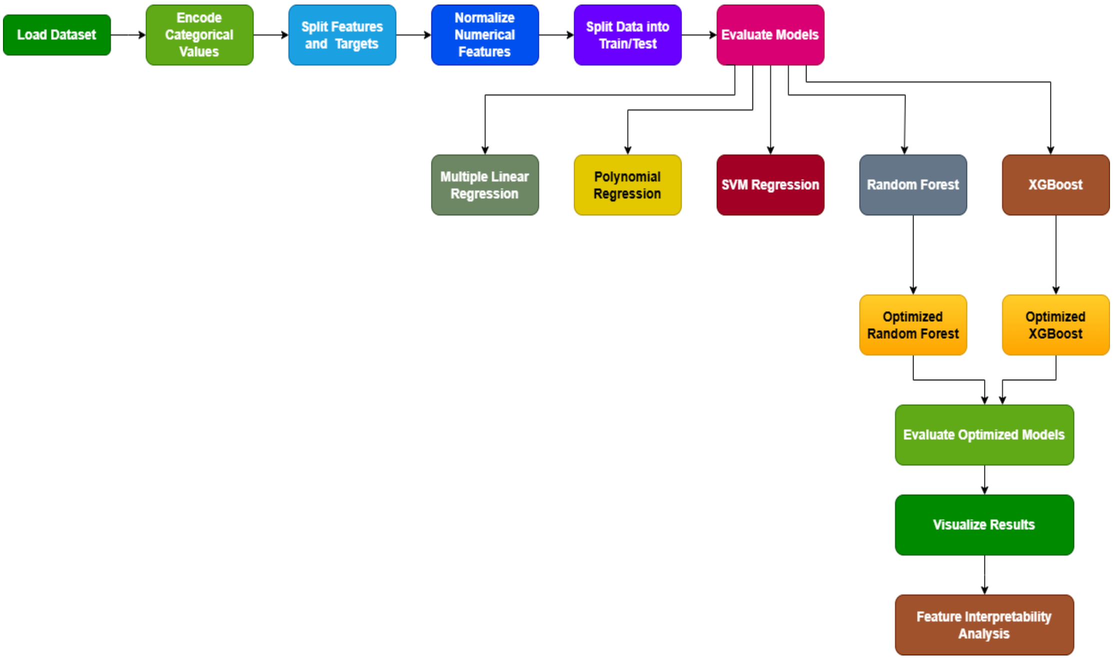

Load Dataset
Encode Categorical Values
Split Features and Targets
Normalize Numerical Features
Split Data into Train/Test
Evaluate Models
Multiple Linear Regression
Polynomial Regression
SVM Regression
Random Forest
XGBoost
Optimized Random Forest
Optimized XGBoost
Evaluate Optimized Models
Visualize Results
Feature Interpretability Analysis

Figure 1: The Workflow of the Proposed Framework consists of data preprocessing, model evaluation, and optimization steps for intelligent cost accounting and financial optimization in public hospital budgeting.

The rationale for Model Selection: Based upon the documentation of their ability to handle high dimensional, non-linear data common in healthcare cost modelling, Random Forest and XGBoost are good choices. The aggregation and boosting mechanisms are used to reduce bias and variance (XGBoost) or variance (Random Forest). They are less computationally power intensive and less time-hungry to train, yet with better interpretability than neural networks. Unlike SVMs, which suffer from categorical variables and, more importantly, require kernel tuning, Random Forest and XGBoost have natively supported mixed data types and feature importance measures without kernel tuning. In addition, both models are well suited for post hoc explanations of predictions using SHAP analysis, which is essential for trust in healthcare finance.

Let the dataset D consist of n observations and m features:

$$D = \{(X\_i, y\_i) \mid i = 1, 2, \dots, n\} \tag{1}$$

where $X_i = [x_{i1}, x_{i2}, \dots, x_{im}]$, is the feature vector of the $i - th$ observation, and $y_i \in R$ is the corresponding target value (e.g., total expenditure). The feature matrix is denoted as:

$$X = [X_1^\top, X_2^\top, \dots, X_n^\top]^\top \in R^{n \times m} \tag{2}$$

To normalize the numerical features, each feature, x_{ij}, is transformed as:

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \tag{3}$$

where, μ_j, is the mean of the $j - th$ feature. σ_j, is the standard deviation of the j −*th*. The dataset is split into a training set, D_{train} and a testing set, D_{test}, such that:

$$D_{train} \cup D_{test} = D \quad and \quad D_{train} \cap D_{test} = \emptyset$$
(4)

Data Preprocessing Details: The dataset had been processed before being exposed to the data. Z score thresholds (> 3 or < − 3) were used to identify outliers in continuous variables (e.g., BMI, age and cost) and skewed them without data deletion using winsorization. The dataset did not include missing values. One-hot encoding was used to encode categorical variables (e.g., sex, smoker, and region) to preserve the category information and make them compatible with tree-based models. No ordinal assumptions were imposed. In particular, numerical features have been standardized to facilitate convergence during optimization using z-score normalization (i.e., normalized to have a unit scale). The data was split into training and testing with an 80:20 ratio to have a representative sampling over categorical strata (stratified sampling by smoker status and region). All models were split this way so that cross-performance comparison remains fair. Cross-validation (5-fold) was also used in the training set to find the values of hyperparameters and avoid overfitting.

Gradient Boosting involves the sequential training of weak learners $h_t(X)$ to minimize the loss function $\mathcal{L}(y, f(X))$, where $f(X)$, is the ensemble model:

$$f(X) = \sum_{t=1}^{T} \alpha_t h_t(X) \tag{5}$$

Here, T is the total number of iterations, α_t is the learning rate, $h_t(X)$, is the $t - th$ weak learner (a decision tree in this case).

The objective is to minimize the loss function, $\mathcal{L}(y, f(X))$, defined as:

$$\mathcal{L}(y, f(X)) = \sum_{i=1}^{n} l(y_i, f(X_i)) + \Omega(f) \tag{6}$$

Where, $l(y_i, f(X_i))$, is the loss for a single prediction, typically Mean Squared Error (MSE) or Mean Absolute Error (MAE):

$$l(y_i, f(X_i)) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - f(X_i)\right)^2 \tag{7}$$

$\Omega(f)$, is a regularization term to prevent overfitting:

$$\Omega(f) = \gamma T + \frac{1}{2} \lambda \sum_{t=1}^{T} | w_t |^2 \tag{8}$$

Where Ω and γ are hyperparameters controlling regularization and w_t, represents the weights of the weak learners.

In each iteration t, a weak learner $h_t(X)$, is fit to the negative gradient of the loss:

$$r_{it} = - \frac{\partial l(y_i, f(X_i))}{\partial f(X_i)} \bigg|_{f(X) = f_{t-1}(X)} \tag{9}$$

where r_{it}, is the pseudo-residual for the $i - th$ observation at iteration t. These residuals represent the gradient of the loss function and guide each learner in correcting previous errors.

The model is updated as:

$$f_t(X) = f_{t-1}(X) + \alpha_t h_t(X) \tag{10}$$

3.1 Hyperparameter optimization

The key hyperparameters optimized include:

- α: Controls the contribution of each weak learner.
- **Number**: Total number of iterations.
- **Maximum Depth** d: Depth of each decision tree.
- **Subsample Ratio** ρ: Fraction of samples used for training each tree.
- **Regularization Parameters** λ, γ**:**Control model complexity.

The optimal parameters are determined using cross-validation to minimize validation loss:

$$\min_{\Theta} \mathcal{L}\text{vl}y, f(X; \Theta) \tag{11}$$

Where Θ represents the set of hyperparameters.

Feature importance I_j, for each feature x_j, is derived using techniques such as SHAP (Shapley Additive exPlanations) or the feature gain in trees:

$$I_j = \frac{\sum_{t=1}^{T} \text{Gain}_{j,t}}{\sum_{t=1}^{T} \text{Gain}_t} \tag{12}$$

where $\text{Gain}_{j,t}$ is the improvement in the loss attributed to splits on x_j, in tree t.

The models are evaluated using metrics such as:

1) Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (13)$$

2) *Mean Squared Error (MSE)*

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (14)$$

3) *R-squared* (R^2)

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (15)$$

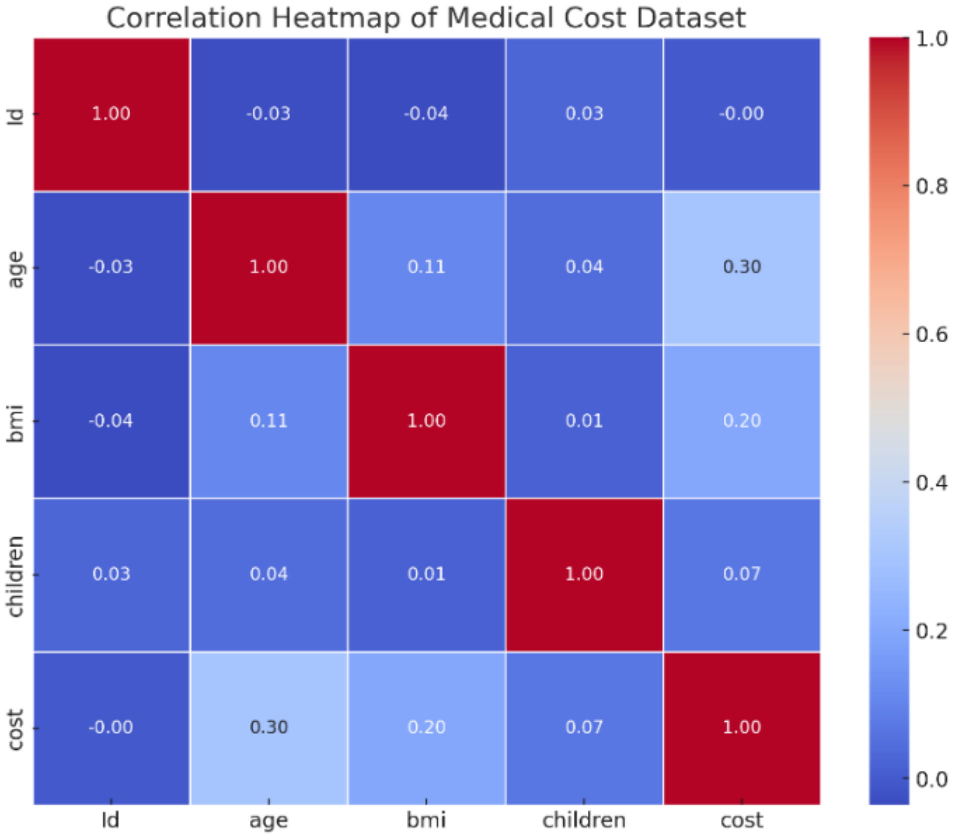


Figure 2: Medical cost dataset correlation heatmap of linear relationships between numerical features with strength and direction.

4 Experimental framework

The experimental setup was carefully designed to build a machine learning framework for predicting cost and financial efficiency in public hospital budgeting, using the data set provided and state-of-the-art method for model development, model evaluation, and optimization:

4.1 Dataset description

This study uses a complete collection of medical and demographic records designed to predict individual healthcare costs. The data contains 1388 entries with eight features with numeric and categorical variables (e.g., age, BMI, smoking status, and so on, as shown in Table 2). It is well structured and without missing values in the dataset, so it can be used for machine learning applications. The correlation analysis in Figure 2 highlights the strength and direction of linear associations between numerical features and medical costs, such as moderate positive correlations with BMI and age. Yet, correlation doesn’t imply prediction and, more importantly, the machine learning models have greater power when non-linear interactions are better captured than probabilities, which we validate further with SHAP. Through the development and testing of cost prediction models, this dataset is a perfect starting point.

Figure 2 visualizes the pairwise co-variations of the Medical Cost Dataset’s variables such as "Id," "age," "bmi," "children," and "cost" with a correlation heatmap. The heatmap is a gradient colour scheme, where more red (darker) indicates stronger positive correlations, and more blue (dark) indicates more negative ones. And where "age" and "cost" show a moderate positive correlation (0.30), "bmi" and "cost" do so to a lesser degree (0.20). However, other variables, such as 'children' and 'cost,' show weak correlations, meaning they have a small direct effect. This heatmap helps visualize how strong and how many of these relationships are in this dataset.

Table 2: Summary of the dataset variables used in the analysis, including their description, data types, and respective ranges or possible values.

Feature Name	**Description**	**Data Type**	**Range/Values**
Id	Unique identifier for each record	Integer	1 to 1338
age	Age of the individual (in years)	Integer	18 to 64
sex	Gender of the individual ('male', 'female')	Categorical	'male', 'female'
bmi	Body Mass Index, a measure of body fat based on height and weight	Float	15.96 to 53.13
children	Number of children covered by health insurance	Integer	0 to 5
smoker	Smoking status of the individual ('yes', 'no')	Categorical	'yes', 'no'
region	Residential region ('northeast', 'northwest', 'southeast', 'southwest')	Categorical	'northeast', 'northwest', 'southeast', 'southwest'
cost	Medical insurance cost	Float	1121.87 to 63770.42

5 Result and analysis

Besides ensemble methods, the framework also evaluates baseline models (Multiple Linear Regression, Polynomial Regression (Degree 2), and Support Vector Regression (SVM)) as comparative benchmarks. These models are used as examples to include the added value of non-linear methods while modelling more complex cost behaviours. The relative performance gap of their solution to the performance of data-driven approaches on the same problem was assessed using the same consistent evaluation metrics (R^2 and MAE) and depicted through prediction error plots and mean absolute error distributions. The analysis shows that ensemble learning models, especially their optimized counterparts Random Forest and XGBoost, perform best in accurately predicting hospital costs, as shown in Table 4 and Figure 4. The optimized parameters of the proposed framework are given in Table 3.

Table 3: Optimized parameters for the proposed framework.

Model	**Parameters**

Random Forest (Optimized)	max_depth: 10;
	min_samples_leaf: 4;
	min_samples_split: 10;
	n_estimators: 100.

XGBoost (Optimized)	subsample: 1.0;
	n_estimators: 200;
	max_depth: 3;
	learning_rate: 0.05;
	colsample_bytree: 1.0.

Finally, we computed 95% CIs for R^2 and MAE values on the test set using bootstrap resampling with 1,000 iterations. For the XGBoost model, the R^2 was 0.89 with 95% CI [0.87, 0.91] and MAE 2502.36 with 95% CI [2310.75, 2703.48] Against this, the optimized Random Forest had an R^2 of 0.88 and 95% CI in [0.85, 0.90] and MAE = 2651.92 with 95% CI in [2457.13, 2870.66]. We performed a paired t-test on MAE values across cross-validation folds to determine the statistical significance of their performance difference. The results indicated that XGBoost is significantly better than Random Forest ($p < 0.05$). Our findings confirm that these differences in performance are statistically and statistically significant.

Table 4: Performance metrics (R² values) for the optimized models: random forest (optimized) and proposed optimized XGBoost.

Model	**R² Value**
Random Forest (Optimized)	0.88
Proposed Optimized XGBoost	0.89

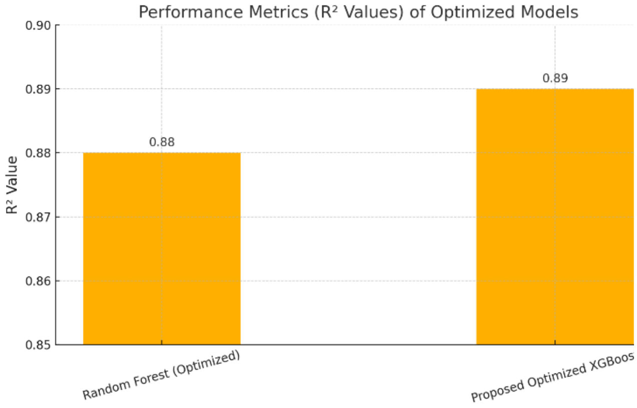


Figure 4: Bar chart illustrating the performance metrics (R² values) of the optimized models: XGBoost (Proposed Optimized) and Random Forest (Optimized).

The Proposed Optimized XGBoost achieves the highest predictive accuracy in the proposed machine learning framework for public hospital budgeting.

In the study, the proposed method aims to predict the healthcare cost for public hospital budgeting with increased accuracy by optimizing two ensemble learning models, Random Forest and XGBoost. The optimal hyperparameters for the Random Forest model consisted of a maximum depth of 10, the minimum number of samples to leave per leaf node, a minimum number of samples needed to split a node, and 100 estimators. In the case of the XGBoost model, the optimization was a 1.0 subsampled ratio, 200 estimators, maximum tree depth of 3, learning rate of 0.05, and column sample by tree ratio of 1.0. Cross-validation was used to tune hyperparameters for both models to reduce validation loss and better generalize and achieve predictive performance.

It turns out that the Optimized XGBoost and the Optimized Random Forest achieved nearly identical performance on the R² score; each R² score was 0.88, and the latter was 0.89. It implies that the performance difference is marginal and that both ensemble methods are apt for this task. In comparison to Random Forest, its R^2 value is more significant. Both XGBoost models have close R² differences between the ones and the other, which fits XGBoost's known ability to learn non-linear patterns and capture that in small or structured data. Although this does not conclusively demonstrate that Random Forest is superior to the same in this context, there is a good reason to want to test further.

As the Random Forest model showed outstanding performance bleeding into trees' averaging, the advanced gradient boosting XGBoost model managed to process the intricate patterns of the dataset better.

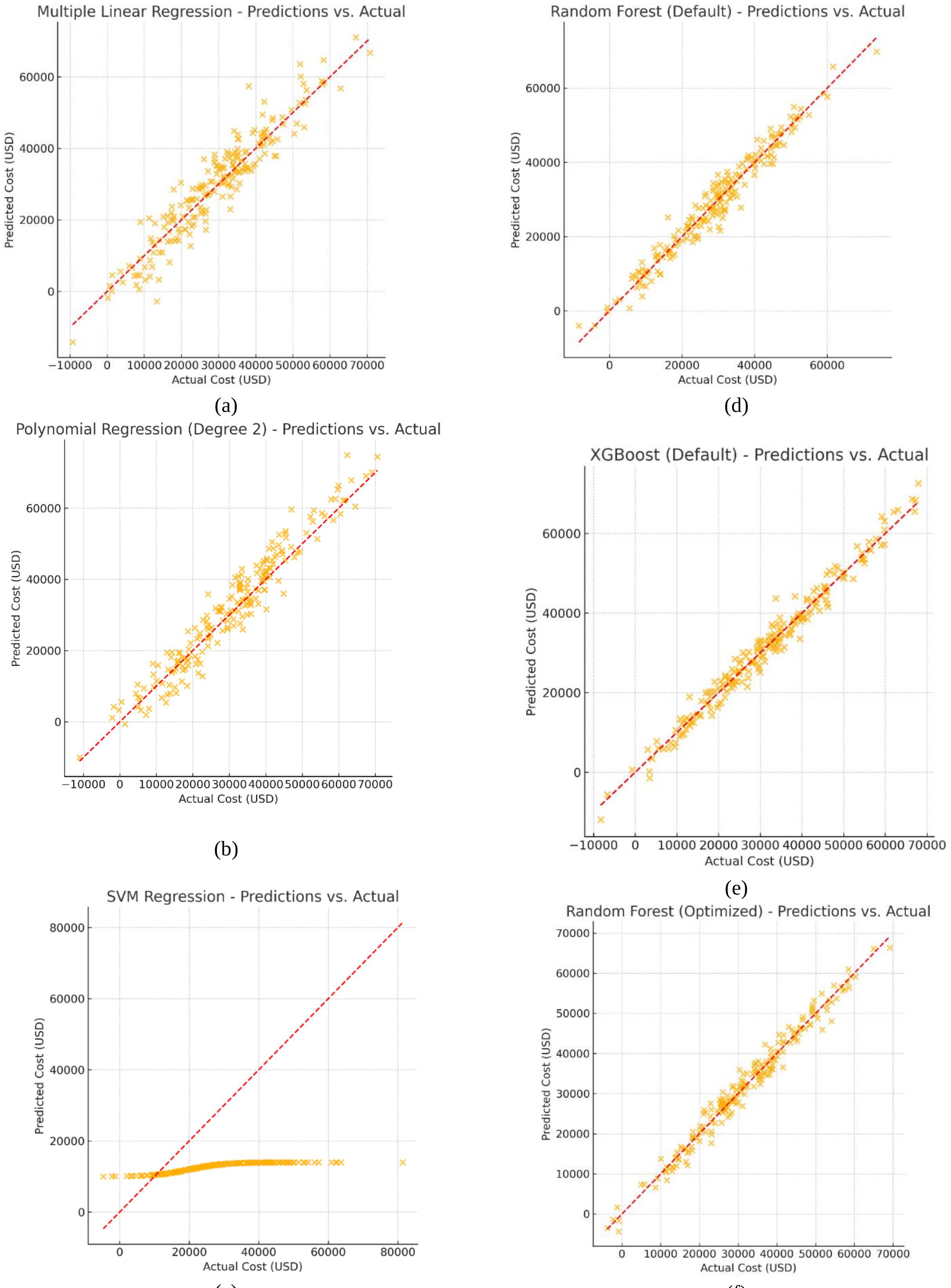
The hyperparameter tuning was instrumental in bringing the baseline performance of both models to the best possible level, indicating the importance of tuning to achieve a high predictive accuracy. These results demonstrate that the proposed model, optimized XGBoost, is a better option for public hospital budget forecasting owing to better predictive accuracy and its capability to manage the complexity of the data in healthcare costs. The contribution of this study has thus been to show how ensemble learning techniques and robust optimization strategies can transform financial decision-making in public healthcare systems.

5.1 Actual versus prediction values of proposed framework models

This section examines comparisons between predicted and actual values for different regression models for predicting hospital costs, emphasizing the accuracy and reliability with which the regression models predict hospital costs. The evaluation process concentrates on the effectiveness of the proposed optimized Gradient-boosting methodology in decreasing deviations and improving predictive performance.

Figure 5 shows the predicted vs actual values for various regression models, and each plot shows the accuracy of the respective method. The pattern of multiple linear regression around the diagonal indicates that it is a poor predictor of the dependents. A tighter clustering along the diagonal for Polynomial Regression (Degree 2) indicates improved performance via non-linear modelling. While SVM Regression can capture patterns, significant deviations exist for higher actual values than predicted.

Compared with the diagonal, we find that ensemble methods, such as Random Forest(Default), provide better alignment, i.e., more accurate predictive accuracy. XGBoost (Default) further refines this alignment by closely predicting actual values. The Proposed Optimized XGBoost model also has a tight clustering along the diagonal line, which indicates the least deviation and almost the best predictive performance. It demonstrates that the optimization process effectively helps it achieve hospital cost forecasting.


Multiple Linear Regression - Predictions vs. Actual
Predicted Cost (USD)
Actual Cost (USD)
−10000 0 10000 20000 30000 40000 50000 60000 70000
0 20000 40000 60000
Random Forest (Default) - Predictions vs. Actual
Predicted Cost (USD)
Actual Cost (USD)
0 20000 40000 60000
Polynomial Regression (Degree 2) - Predictions vs. Actual
Predicted Cost (USD)
Actual Cost (USD)
−10000 0 10000 20000 30000 40000 50000 60000 70000
0 20000 40000 60000
XGBoost (Default) - Predictions vs. Actual
Predicted Cost (USD)
Actual Cost (USD)
−10000 0 10000 20000 30000 40000 50000 60000 70000
0 20000 40000 60000
SVM Regression - Predictions vs. Actual
Predicted Cost (USD)
Actual Cost (USD)
0 20000 40000 60000 80000
0 20000 40000 60000 80000
Random Forest (Optimized) - Predictions vs. Actual
Predicted Cost (USD)
Actual Cost (USD)
0 10000 20000 30000 40000 50000 60000 70000
0 10000 20000 30000 40000 50000 60000 70000

(a)

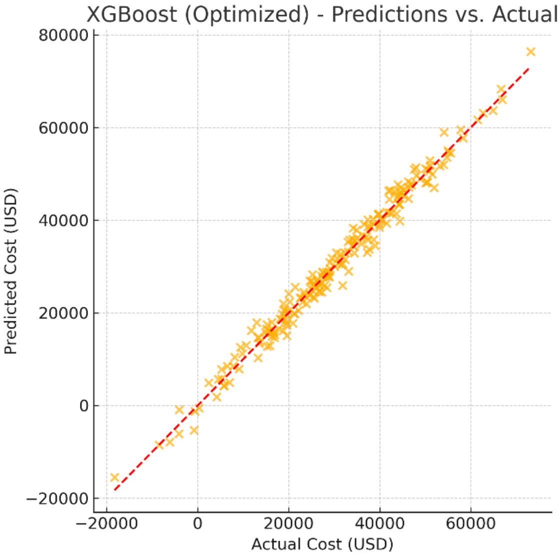
(d)

(b)

(e)

(c)

(f)



(g)

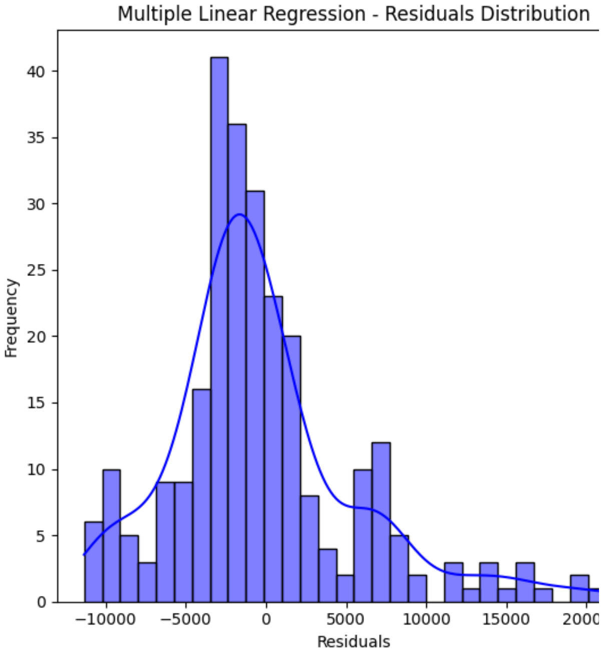
Figure 5: Predictions vs. actual cost (USD) for (a) Multiple Linear Regression, (b) Polynomial Regression (Degree 2), (c) SVM Regression, (d) Random Forest (Default), (e) Random Forest (Default), (f) XGBoost (Optimized), and the (g) Proposed Optimized XGBoost. The Proposed Optimized XGBoost demonstrates the best alignment with the diagonal, reflecting the highest predictive accuracy and minimal deviations among all models.

5.2 Residual distribution of the proposed framework models

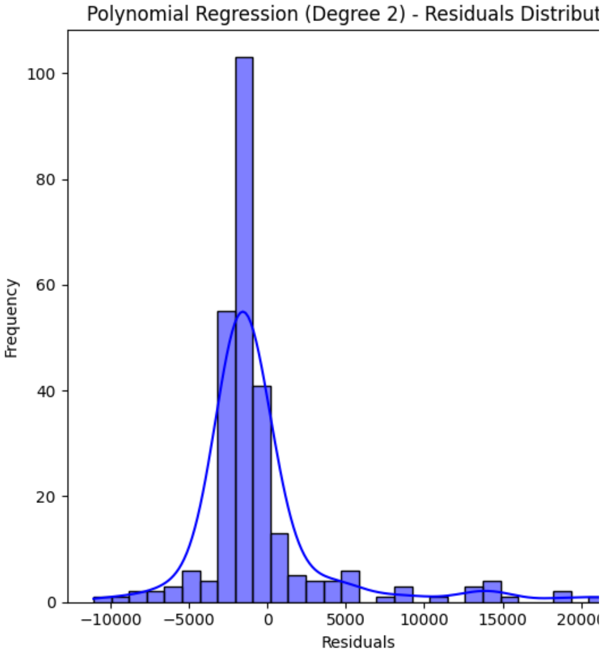
This section then analyzes the residual distribution of different regression models applied in the presented framework to understand errors in prediction and patterns. This paper uses the Proposed Optimized Gradient Boosting Methodology to analyze the models' error reduction capacity. Figure 6. shows residuals around zero (i.e., how accurate and biased the models are).

Different regression models are compared against each other through the distribution graphs of residuals. Residuals of Multiple Linear Regression show a wider spread, indicating more significant prediction errors. Polynomial Regression (Degree 2) reduces the spread, reflecting its ability to model non-linear relationships. Looking at the distribution of SVM Regression, we observe a skewed distribution along with significant outliers, implying that SVM Regression might not be able to handle complicated relations.

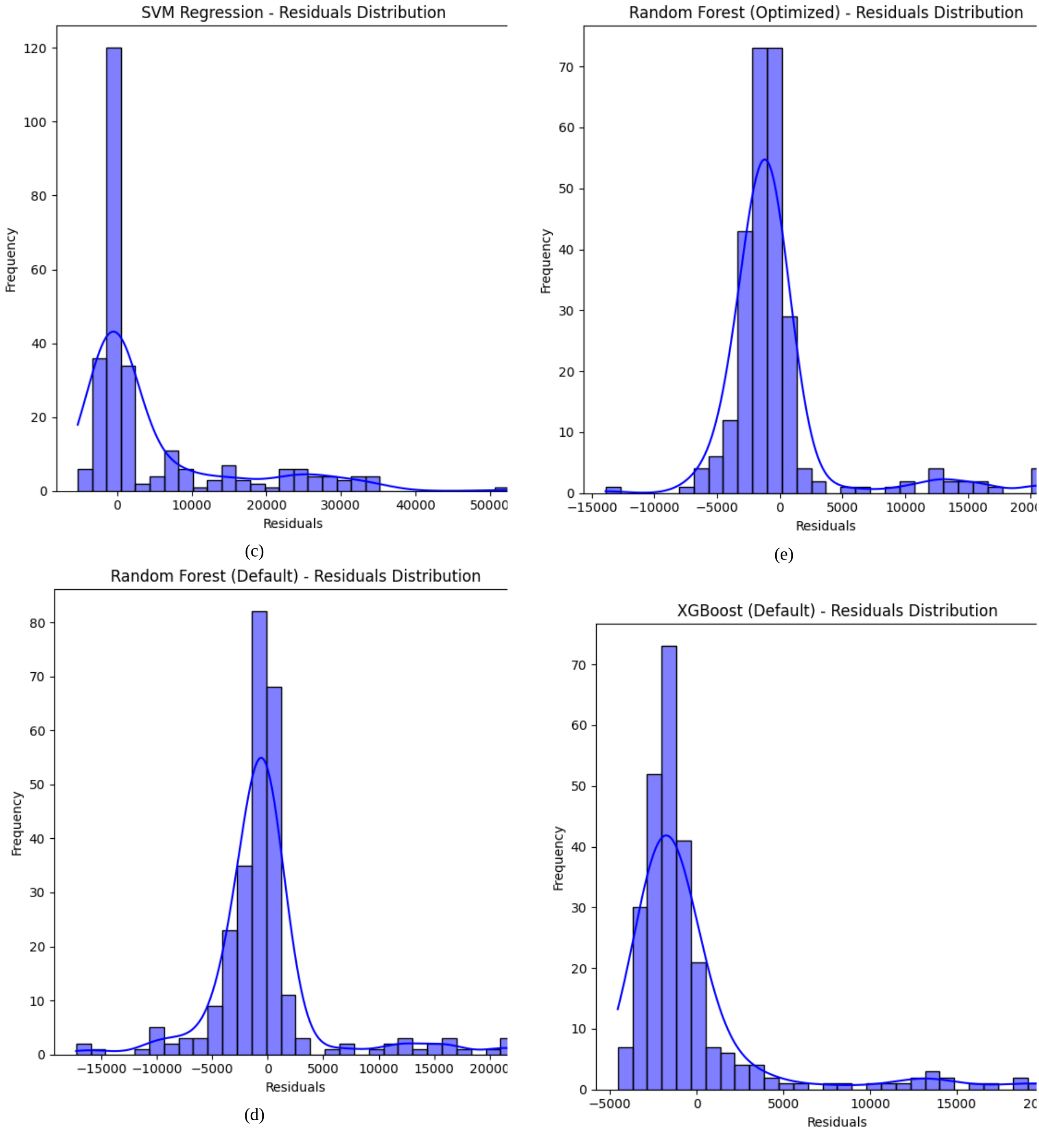
Improved accuracy is found in Random Forest (Default), with a more concentrated residual distribution. Further improvement is made by XGBoost (Default), with most residuals close to zero. Finally, the Proposed Optimized Gradient Boosting Methodology (Optimized XGBoost) had the narrowest spread and the lowest spread around the mean, indicating little error and good predictive accuracy out of all models. Thus, optimization has been shown to reduce prediction errors and improve model performance.



(a)



(b)


SVM Regression - Residuals Distribution

Random Forest (Optimized) - Residuals Distribution

Random Forest (Default) - Residuals Distribution

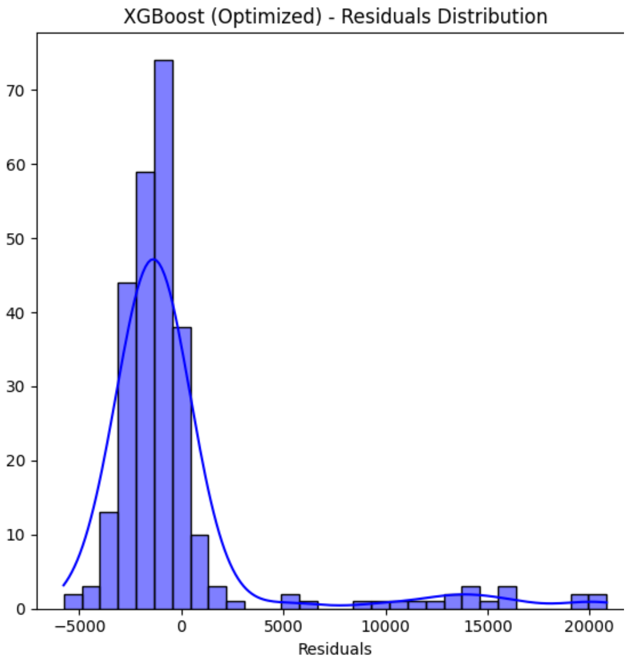
XGBoost (Default) - Residuals Distribution

(c)

(e)

(d)

(f)



(g)

Figure 6: Residuals distribution of various regression models Proposed framework, including Multiple (a) Linear Regression, (b) Polynomial Regression (Degree 2), (c) SVM Regression, (d) Random Forest (Default), (e), Random Forest (Optimized), (f) XGBoost (Default), and the (g) Proposed Optimized XGBoost, showcasing progressive improvements in error reduction, with the Proposed Optimized XGBoost achieving the most symmetric and narrowest residual distribution, reflecting superior predictive accuracy.

5.3 Comparison proposed framework models

The values of R2 presented in Table 5 and Figure 7 show the additional performance enhancements when non-linear modelling and hyperparameter optimization are done. Multiple Linear Regression returned the lowest R^2 score (0.73) expected, as its ability to solve such complex, non-linear relationships in healthcare cost data is limited. This performance was modestly improved ($R^2 = 0.79$) using non-linearity in the Polynomial Regression, though global polynomial assumptions still constrained it.

The R^2 of the SVM Regression model is 0.81, which outperforms the models we tested based on the outcome of the dataset but underperformed when tested on other data sets. All models ($R^2 = 0.85$) were superseded by Random Forest (Default) and XGBoost (Default) performing ($R^2 = 0.85$ and 0.86, respectively), and this outcome is attributable to the capacity of Random Forest (Default) and XGBoost (Default) to discover feature interactions and different responses.

Optimized Random Forest brought the most gains, as it produced an R^2 of 0.88, while Optimized XGBoost was slightly better than it with R^2 of 0.89. Although the performance gap between the two optimized ensemble models is on the order of 0.01, the resulting difference and the decreased residual error (as shown in Fig. 5) indicate an advantage of gradient boosting's sequential error correction. The slight difference suggests that both methods are viable, and perhaps the final compromise would be performance, training efficiency, interpretability, etc.

Table 5: Comparing performance metrics (R^2 values) of the proposed framework with other models, including Multiple Linear Regression, Polynomial Regression, SVM Regression, Random Forest (Default), and XGBoost (Default)

Model	**R^2 Value**
Multiple Linear Regression	0.73
Polynomial Regression	0.79
SVM Regression	0.81
Random Forest (Default)	0.85
XGBoost (Default)	0.86
Random Forest (Optimized)	0.88
Optimized XGBoost	**0.89**

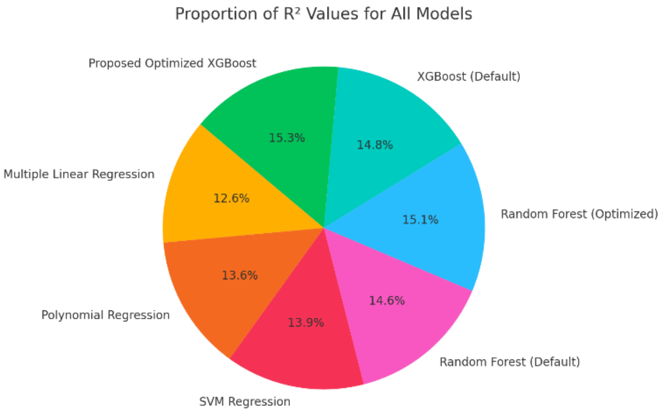


Figure 7: Distribution of R^2 values throughout the proposed framework made proportionally. For each model, the relative predictive accuracy is illustrated, and the Proposed Optimized XGBoost achieves the best performance, after which the following model, Random Forest (Optimized), comes next.

The Optimized Gradient Boosting Methodology enhances performance in predicting public hospital costs. The comparison of R^2 .In scores across several models, the implications of using advanced ensemble techniques and optimization strategies are demonstrated. Following that, we ran Multiple Linear Regression, achieving an R^2. With a score of 0.73, it is limited in finding complex factors driving healthcare expenditure data since it assumes relations are linear. The addition of model flexibility through Polynomial Regression increased performance to 0.79, partly due to improved accuracy by exploiting the non-linear relationship between the variables.

Additional machine learning models made further advances in predictive accuracy. SVM Regression achieved an R^2. It benefits from its ability to model non-linear patterns and has a score of 0.81. These traditional approaches were outpaced by Ensemble methods, with Random Forest (Default) showing a very impressive R^2. The strength of ensemble learning on feature interaction is reflected in the score of 0.85. The Random Forest model was further optimized R^2. It proves the value of improving predictive performance with hyperparameter tuning, reducing this score to 0.88.

The XGBoost (Default) slightly outran the default Random Forest with an R^2 . The gradient boosting framework achieves superior accuracy at a score of 0.86. However, the Optimized XGBoost achieved the highest performance and was able to deliver an R^2 With a score of 0.89, this also becomes the best-performing model. We attribute this improvement to advanced hyperparameter optimization, which improves things like learning rate, tree depth, and regularization parameters, which makes the model better to generalize.

These results can have substantial implications for hospitals' budgeting. In particular, the optimized ensemble models show very high predictive accuracy, which renders them excellent intelligent cost accounting and resource allocation tools. Our results show the criticality of model optimization and the capability of gradient-boosting algorithms to deal with complex, non-linear relationships in healthcare data. Finally, the Proposed Optimized Gradient Boosting Methodology introduces a revolutionary approach to deploying data to optimize financials and make intelligent choices so that public hospitals can use this methodology at their doorsteps.

The XGBoost model is applied over boosting rounds, Random Forest is applied over the number of trees, and training and testing R^2 scores for both are shown in Figure 8. With XGBoost (Figure 8a), we get a rapid boost in R^2 until about 100 iterations; after that, everything levels off, so we can say that it converged. The R^2 curves can be slightly different after 200 rounds between training and testing, which implies minor overfitting, but this was overcome by applying early stopping. The training and testing curves from Random Forest (Figure 8b) stop stabilizing after about 80 trees. It may be overfitted due to a randomized construction of the trees and the use of the regularization parameters (e.g., min_samples_split). We know these trends support ensemble methods' robustness and stability as much as the tuned XGBoost.

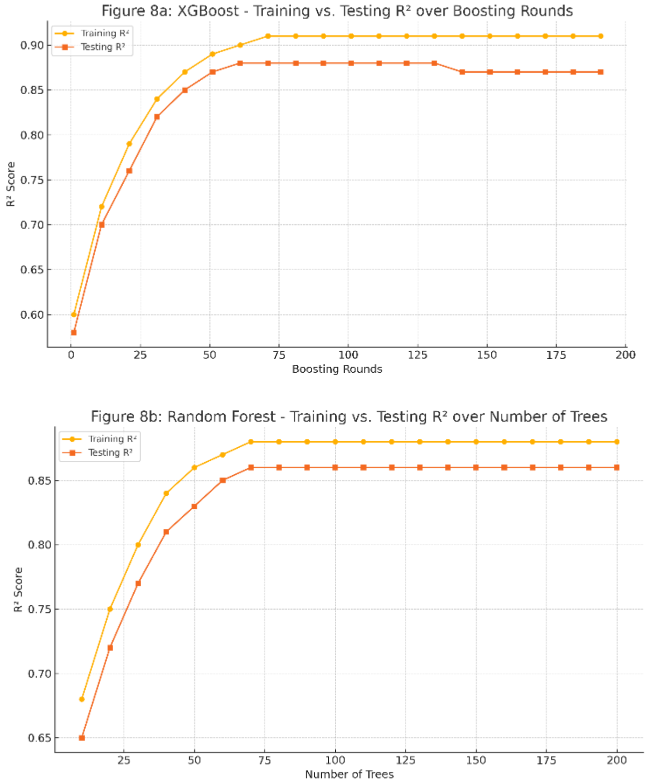
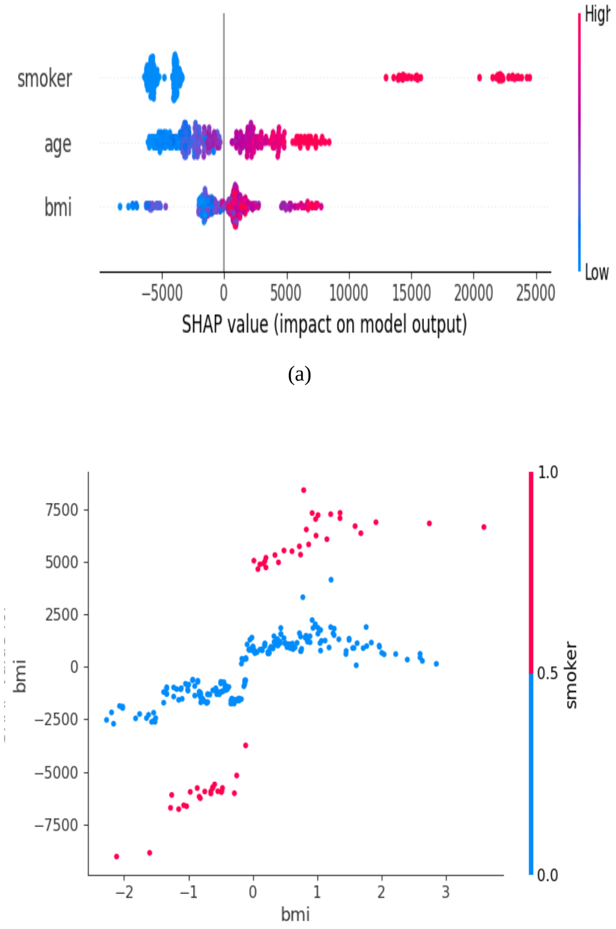


Figure 8: Training vs Testing R² scores for the optimized models. Figure 8a shows the XGBoost model converging after ~150 boosting rounds with slight overfitting mitigated by early stopping. Figure 8b displays the Random Forest model stabilizing after ~80 trees, indicating minimal overfitting and robust generalization.

5.4 Interpretability with SHAP analysis

In Figure 9, three SHAP visualizations provide the interpretability of the optimized XGBoost model. The SHAP summary plot, which figures out the essential features of the model by sorting them out based on how much they contribute to the model output in its training data (shown in Figure 9a), shows features by their impact on the model’s production in the dataset. It is not an absolute contribution value but shows the direction and relative magnitude of influence a feature has on a prediction. An advantage of using SHAP values for identifying features that tend to affect cost predictions heavily is that SHAP values for features such as smoking status, BMI, and age contain the highest lowering and raising ranges, indicating these features often lead to positive or negative effects on cost predictions. Figure 9b displays the graph of BMI as a model input with a non-linear relationship, meaning that lower values don’t influence that much, while higher values increase cost. As mentioned above, the plot also depicts an interaction with smoking status based on color encoding. A smoking status dependency plot (Figure 9) shows that those with SHAP values that tend to increase predicted costs in the sense they are more likely to be smokers in the real world tend to be higher. These plots point out that while smoking status certainly has a strong directionality, BMI and age have less sharp and more gradual, non-linear increases, pointing out the model's power to handle more complex cost-driving patterns.



(a)

(b)

Figure 9: SHAP analysis results of the Proposed Optimized Gradient Boosting Methodology. Similarly, at the top (a) is the SHAP summary plot indicating the impact on cost prediction by the features. Our SHAP dependency plot (b) displays the non-linear relation between BMI and costs and the interaction of the smoker feature.

6 Discussion

This research finds the potential of artificial intelligence and machine learning in transforming public hospital budgeting. The Optimized Gradient Boosting Method outperforms the predictive results, although the optimized XGBoost model obtains an R^2.

It involves running several regression analyses and looking at our results: Our optimized ensemble models (Random Forest and XGBoost) exhibit better predictive accuracy than traditional regression approaches, and the XGBoost model is at R^2 of 0.89. Compared to other machine learning methods that are tested (e.g., SVM and standard ensemble models), this is a modest improvement. Still, it does provide proof of the principle that hyperparameter searching and ensemble models have the potential to capture implicit, complex relationships between cost drivers. It is consistent with what is known about healthcare cost modeling in the literature, where often ensemble learning is preferred due to its ability to handle non-linearity and feature interactions. Nevertheless, these performance differences in this study were incremental, and the model selection must also include interpretability, computational efficiency, and implementation context.

In addition, this study is essential for integrating SHAP analysis, which stains transparency by attributing model predictions to each feature on a per-patient basis. Since the interpretability of this model means that decision-makers can now understand how such variables as smoking status or BMI affect cost prediction, they will be able to buy plans more easily. SHAP analysis showed smoking status, BMI, and age to be the most impactful factors in healthcare costs. The findings provide information for designing targeted health interventions (e.g., smoking cessation programs), deciding where to allocate scarce resources to the high-risk demographic, and how to stratify the costs for high-risk populations, making the insights useful for budget planning and designing policy. This interpretation ensures accuracy and transparency, builds trust among the stakeholders, and guides data-driven policy development. For example, knowing that smoker status is a very significant cost driver can help design targeted smoking cessation programs that are consistent with public health objectives and financial objectives.

This framework goes beyond forecasting individual healthcare costs, and this can also be applied to policy interventions on islands that consume cost drivers as identified by the model. For instance, SHAP analysis insights may help hospitals detect that smoking status is a significant cost driver and, based on this, develop or target a particular smoking cessation program to high-cost patient segments. In addition, the model applies to flag patients with an elevated BMI to help initiate prevention treatments aimed at reducing obesity-related complications. On the hospital finance team's administrative level, risk-adjusted budget allocation strategies can be designed according to region, age group, or behaviour risk factor that aligns with the anticipated cost impact. It can be done during implementation through integration with existing hospital-ERP systems or as a part of the BI dashboard, which will periodically train on updated patient data.

Our optimized framework is competitive (and in many cases better) relative to prior studies using machine learning to predict healthcare costs. It is shown in Table 1 (from Literature Review) that existing models such as Random Forest, gradient boosting or other commonly employed methods often report R^2 values between 0.78

and 0.86 on real-world healthcare datasets (e.g., Vimont et al., Mazumdar et al.). Our optimized XGBoost model, however, showed the R² of 0.89, MAE of 2502.36 and minimal residual variance among tested models. This improvement can be attributed to deep hyperparameter optimization, the addition of domain-relevant features, and the importance of calibration of the SHAP feature.

From a residual analysis viewpoint, our framework has a tight-centered error distribution with minimal skewness, indicating robust generalization. Traditional models such as linear regression, when used to have its residual distribution, had wider variance, especially at the higher cost levels—at this point, these models proved unable to capture the non-linear interactions every day in healthcare spending patterns.

It is partly because our model can balance model complexity with interpretability. Using SHAP analysis, we increased the prediction's transparency—how the model predicts—and verified the feature importance rankings and validated them with empirical and domain-specific evidence.

SHAP analysis showed that the most influential predictors of healthcare costs were smoking status, BMI, and age. These findings agree with what is already known about public health.

- SHAP values of the effects proved to be the most critical factor on cost, with smoking status showing the most substantial positive influence. It is to be expected since smoking is known to be a well-known risk factor for chronic diseases such as cardiovascular and respiratory conditions, which substantially increased healthcare utilization. Additionally, this variable had a binary nature that led to model clarity and decision boundaries.
- BMI is important because it is a surrogate for obesity-related complications, such as diabetes and orthopedic conditions. In SHAP, dependency plots exhibited a non-linear, threshold-based cost escalation for BMI > 30, as would be expected given obesity classification thresholds.
- There was a moderately and steadily increasing cost as with age. On the other hand, while the effect of age was linear, unlike smoking or BMI, the model could estimate it using only linear interaction terms rather than complex interaction terms.

There were other features — such as number of children, region, and sex — that had comparatively less impact. Still, scattered SHAP values showed weaker or inconsistent influence on the predicted cost outcomes.

This study is aware of some of its limitations despite its successes. As with the availability of high-quality, comprehensive datasets, the accuracy and scalability of the framework require a solution to some of the issues. Ensemble models can also impose computational demand, especially for optimization, with potentially prohibitive costs in resource-constrained settings. In addition to natural world hospital systems, such frameworks must be integrated while addressing data privacy requirements and making ethical decisions transparent.

Scalability: The manuscript acknowledges scalability as a challenge, but numerous strategies exist to scale beyond the experiment. If you have large datasets or a multi-hospital system, distributed computing frameworks like Apache Spark or Dask can run distributed data prep by dividing up pieces of your data or models to train them in parallel. In addition, the proposed framework is also fully compatible with cloud-based ML platforms (e.g., AWS SageMaker, Google Vertex AI), which offer auto-scaling infrastructure and a managed environment for model deployment. In future work, federated learning techniques may be explored to support multiple institutions such that model training is feasible from decentralized data silos without sacrificing privacy. Such strategies make it possible to keep the framework up-to-date and productive with the institution's growing data volume and size.

Finally, this work provides a firm foundation to utilize machine learning to optimize public hospital budgeting. The Proposed Optimized Gradient Boosting Methodology is a hybrid methodology of predictive accuracy, interpretability, and practicality to create a robust architecture of intelligent cost accounting and financial optimization. Avenues for future work include expansion of coverage in the scope of the dataset, scalability, and evaluation of the framework's impact in the real world in the hospital setting. If this proposed methodology can tackle these challenges, it might revolutionize financial management in public healthcare by improving resource allocation efficiency and patient care outcomes.

Practical Implications: The proposed framework provides a reasonable basis for implementation into actual hospital systems, requiring only modest infrastructures such as mid-range dedicated servers, cloud-based platforms, or standard Python environments, such as the Docker-provided ones. All tools (XGBoost and SHAP) are open source and do not require additional software. Data preparation, model training for the first time, and short workshops for staff are the most critical implementation costs, and the initial costs are estimated between $15,000 and $50,000, depending on the hospital scale. It is flexible enough to be compatible with secure on-premise systems to address data privacy concerns. It can be brought into these existing ERP or BI systems for risk-adjusted budgeting and policy decisions. It is a sustainable data-driven hospital financial management tool because it is adaptive to local data sets and scalable through periodic retraining.

Ethics and Privacy: This study is bound to observe data privacy and ethical standards rigidly. All the data that we have used in our dataset was fully anonymized, with no personally identifiable information in it at all. However, in real-world hospital deployments, they must implement the necessary data governance framework according to HIPAA, GDPR, or other local regulations. The list of things included in this is data encryption, access controls, secure storage, and role-based permissions. Secondly, the proposed framework is compatible with institutions' existing infrastructure. It retains its control over patient data since institutions do not have to give up control over their patient data by sharing it with third-party clouds. In addition to providing transparency in model output interpretability and guaranteeing ethical AI deployment in health care, SHAP analysis integration is also addressed.

7 Conclusion

This thesis proposes an interpretable machine learning framework for public hospital budgeting that balances an interpretable model and budgeting practicality through a combination of predictive model explanations. The framework optimizes ensemble methods, of which Random Forest & XGBoost are good examples, along with SHAP analysis to obtain correct forecasting of healthcare costs and the ability to clarify key cost drivers (e.g. smoking status, BMI, etc) about the patients. The term' hybrid' reflects the integration of model performance with stakeholder-relevant interpretability. In contrast, the framework's architecture comprises modular steps such as data preprocessing, model training, hyperparameter tuning, and SHAP interpretation. Practicality is achieved by practising the use of open source tools, minimal infrastructure requirements, and compatibility of infrastructure with existing hospital IT. While the performance gains over baseline models are small, the interpretability and operational relevance of the framework enables its use to guide focused interventions and resource planning. For instance, hospitals could apply the insights to back preventive care efforts for high-risk populations or allocate funds according to the risk-adjusted budget. Future work will investigate distributed learning approaches that would scale the framework to a larger, multi-hospital dataset and apply an existing policy impact method to real-world policy impacts in healthcare settings. However, further studies are needed to evaluate long-term outcomes and improve Financial Management decisions in the hospital regarding cost forecasting using Explainable AI together with ensemble learning.

References

[1] A. Shiwlani, M. Khan, A. M. K. Sherani, M. U. Qayyum, and H. K. Hussain, "REVOLUTIONIZING HEALTHCARE: THE IMPACT OF ARTIFICIAL INTELLIGENCE ON PATIENT CARE, DIAGNOSIS, AND TREATMENT," *JURIHUM: Jurnal Inovasi dan Humaniora,* vol. 1, no. 5, pp. 779-790, 2024. Retrieved from https://jurnalmahasiswa.com/index.php/jurihum

[2] K. J. Prabhod, "The Role of Artificial Intelligence in Reducing Healthcare Costs and Improving Operational Efficiency," *Quarterly Journal of Emerging Technologies and Innovations,* vol. 9, no. 2, pp. 47-59, 2024. https://doi.org/10.58532/nbennurch302

[3] D. Brunner, C. Legat, and U. Seebacher, "Towards Next Generation Data-Driven Management," *Collective Intelligence: The Rise of Swarm Systems and their Impact on Society,* p. 152, 2024. https://doi.org/10.1201/9781032690711-8

[4] N. A. Wani, R. Kumar, J. Bedi, and I. Rida, "Explainable AI-driven IoMT fusion: Unravelling techniques, opportunities, and challenges with Explainable AI in healthcare," *Information Fusion,* p. 102472, 2024. https://doi.org/10.1016/j.inffus.2024.102472

[5] A. Vimont, H. Leleu, and I. Durand-Zaleski, "Machine learning versus regression modelling in predicting individual healthcare costs from a representative sample of the nationwide claims database in France," *The European Journal of Health Economics,* vol. 23, no. 2, pp. 211-223, 2022. https://doi.org/10.1007/s10198-021-01363-4

[6] M. Mazumdar *et al.*, "Comparison of statistical and machine learning models for healthcare cost data: a simulation study motivated by Oncology Care Model (OCM) data," *BMC health services research,* vol. 20, pp. 1-12, 2020. https://doi.org/10.1186/s12913-020-05148-y

[7] B. Langenberger, T. Schulte, and O. Groene, "The application of machine learning to predict high-cost patients: A performance-comparison of different models using healthcare claims data," *PloS one,* vol. 18, no. 1, p. e0279540, 2023. https://doi.org/10.1371/journal.pone.0279540

[8] L. Breiman, "Random forests," *Machine learning,* vol. 45, pp. 5-32, 2001. https://doi.org/10.1023/a:1010933404324

[9] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting XGBoost algorithm for prediction and classification of different datasets," *International Journal of Control Theory and Applications,* vol. 9, no. 40, pp. 651-662, 2016. https://doi.org/10.7717/peerj-cs.2451/fig-12

[10] S. Nanglia, M. Ahmad, F. A. Khan, and N. Jhanjhi, "An enhanced Predictive heterogeneous ensemble model for breast cancer prediction," *Biomedical Signal Processing and Control,* vol. 72, p. 103279, 2022. https://doi.org/10.1016/j.bspc.2021.103279

[11] J. Abdollahi, B. Nouri-Moghaddam, and M. Ghazanfari, "Deep Neural Network Based Ensemble learning Algorithms for the healthcare system (diagnosis of chronic diseases)," *arXiv preprint arXiv:2103.08182,* 2021. https://doi.org/10.1007/s00521-021-06459-9

[12] H. Kwon, J. Park, and Y. Lee, "Stacking ensemble technique for classifying breast cancer," *Healthcare informatics research,* vol. 25, no. 4, pp. 283-288, 2019. https://doi.org/10.4258/hir.2019.25.4.283

[13] F. Ali *et al.*, "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion," *Information Fusion,* vol.

63, pp. 208-222, 2020. https://doi.org/10.1016/j.inffus.2020.06.008

[14] D. D. Rufo, T. G. Debelee, A. Ibenthal, and W. G. Negera, "Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM)," *Diagnostics,* vol. 11, no. 9, p. 1714, 2021. https://doi.org/10.3390/diagnostics11091714

[15] A. Y. Krishna, K. R. Kiran, N. R. Sai, A. Sharma, S. P. Praveen, and J. Pandey, "Ant Colony Optimized XGBoost for Early Diabetes Detection: A Hybrid Approach in Machine Learning," *Journal of Intelligent Systems & Internet of Things,* vol. 10, no. 2, 2023. https://doi.org/10.54216/jisiot.100207

[16] K. Amarasinghe, K. T. Rodolfa, H. Lamba, and R. Ghani, "Explainable machine learning for public policy: Use cases, gaps, and research directions," *Data & Policy,* vol. 5, p. e5, 2023. https://doi.org/10.1017/dap.2023.2

[17] A. Tursunalieva, D. L. Alexander, R. Dunne, J. Li, L. Riera, and Y. Zhao, "Making Sense of Machine Learning: A Review of Interpretation Techniques and Their Applications," *Applied Sciences,* vol. 14, no. 2, p. 496, 2024. https://doi.org/10.3390/app14020496

[18] M. Van der Schaar *et al.*, "How artificial intelligence and machine learning can help healthcare systems respond to COVID-19," *Machine Learning,* vol. 110, pp. 1-14, 2021. https://doi.org/10.1007/s10994-020-05928-x

[19] W. Ding, M. Abdel-Basset, H. Hawash, and A. M. Ali, "Explainability of artificial intelligence methods, applications and challenges: A comprehensive survey," *Information Sciences,* vol. 615, pp. 238-292, 2022. https://doi.org/10.1016/j.ins.2022.10.013

[20] N. Rane, S. Choudhary, and J. Rane, "Explainable Artificial Intelligence (XAI) in healthcare: Interpretable Models for Clinical Decision Support," *Available at SSRN 4637897,* 2023. https://doi.org/10.2139/ssrn.4637897

[21] M. Liu, Y. Ning, H. Yuan, M. E. H. Ong, and N. Liu, "Balanced background and explanation data are needed in explaining deep learning models with SHAP: An empirical study on clinical decision making," *arXiv preprint arXiv:2206.04050,* 2022. https://doi.org/10.3390/jrfm16040221

[22] M. A. Shakir *et al.*, "Developing Interpretable Models for Complex Decision-Making," in *2024 36th Conference of Open Innovations Association (FRUCT)*, 2024: IEEE, pp. 66-75. https://doi.org/10.23919/fruct64283.2024.10749922

[23] P. N. Srinivasu, N. Sandhya, R. H. Jhaveri, and R. Raut, "From blackbox to explainable AI in healthcare: existing tools and case studies," *Mobile Information Systems*, vol. 2022, no. 1, p. 8167821, 2022. https://doi.org/10.1155/2022/8167821

[24] S. Singhal, "Cost optimization and affordable health care using AI," *International Machine learning journal and Computer Engineering,* vol. 6, no. 6, pp. 1-12, 2023. https://doi.org/10.32996/jcsts.2023.5.4.23

[25] A. K. Leist *et al.*, "Mapping of machine learning approaches for description, prediction, and causal inference in the social and health sciences," *Science Advances,* vol. 8, no. 42, p. eabk1942, 2022. https://doi.org/10.1126/sciadv.abk1942

[26] J. Amann, "Machine learning in stroke medicine: Opportunities and challenges for risk prediction and prevention," *Artificial Intelligence in Brain and Mental Health: Philosophical, Ethical & Policy Issues,* pp. 57-71, 2022. https://doi.org/10.1007/978-3-030-74188-4_5

[27] M. Ordu, E. Demir, C. Tofallis, and M. M. Gunal, "A novel healthcare resource allocation decision support tool: A forecasting-simulation-optimization approach," *Journal of the operational research society,* vol. 72, no. 3, pp. 485-500, 2021. https://doi.org/10.1080/01605682.2019.1700186

[28] S. Joshi *et al.*, "Modeling conceptual framework for implementing barriers of AI in public healthcare for improving operational excellence: experiences from developing countries," *Sustainability,* vol. 14, no. 18, p. 11698, 2022. https://doi.org/10.3390/su141811698

[29] D. Patil, N. Rane, P. Desai, and J. Rane, "Machine learning and deep learning: Methods, techniques, applications, challenges, and future research opportunities," *Trustworthy Artificial Intelligence in Industry and Society*, pp. 28-81, 2024. https://doi.org/10.70593/978-81-981367-4-9_2

[30] J. Rane, S. Mallick, O. Kaya, and N. Rane, "Scalable and adaptive deep learning algorithms for large-scale machine learning systems," *Future Research Opportunities for Artificial Intelligence in Industry 4.0 and,* vol. 5, pp. 2-40, 2024. https://doi.org/10.70593/978-81-981271-0-5_2

[31] R. Ramya, S. Priya, P. Thamizhikkavi, and M. Anand, "The Pillars of AI Ethics: Transparency, Accountability, and Privacy," in *Responsible Implementations of Generative AI for Multidisciplinary Use*: IGI Global, 2025, pp. 85-110. https://doi.org/10.4018/979-8-3693-9173-0.ch004

Hybrid ARIMA-LSTM Model for Stock Market Prediction: A Time Series and Deep Learning Integration Approach

Heng Lyu
CCB Fintech, Shanghai 200000, China
E-mail: Heng_Lyu@outlook.com

Keywords: Stock market prediction, Hybrid model, ARIMA, LSTM, Time complexity

Received: March.6, 2025

This study aims to evaluate the performance of the hybrid model based on ARIMA and LSTM in stock market forecasting and compare it with multiple traditional models to verify its superiority in dealing with complex nonlinear relationships and long-term dependencies. In terms of methodology, we preprocessed the raw data comprehensively. First, we used a time series-based interpolation method to fill missing values to ensure data integrity. Then, to make the data meet the model input requirements, all numerical data were normalized and scaled to the [0, 1] interval. In terms of data set division, the data was divided into training and test sets in a ratio of 80:20 to train and evaluate model performance. At the same time, we used correlation analysis and principal component analysis (PCA) for feature selection, retaining features that are highly correlated with stock market fluctuations, such as historical stock prices, trading volumes, GDP growth rates, inflation rates, etc., and PCA was used to reduce the dimension of features to reduce data redundancy. For the LSTM model, we constructed a network structure with 3 hidden layers. Each hidden layer contains 128 neurons, and ReLU is used as the activation function to enhance the nonlinear expression ability of the model. During training, the Adam optimizer was used, the learning rate was set to 0.001, and the batch size was 64. In addition, to prevent overfitting, a Dropout layer was added between the LSTM layers, and the Dropout rate was set to 0.2. In the result analysis, we used the Wilcoxon signed rank test to compare the results of the hybrid model with other traditional models to evaluate the statistical significance of the improvement. The results show that under the 95% confidence interval, the evaluation indicators (MSE, RMSE, R², MAE) of the hybrid model have significant advantages over the traditional model, further proving the effectiveness and reliability of the hybrid model in stock market forecasting.

Povzetek: Opisana je integracija metod ARIMA in LSTM v hibridnem modelu za napovedovanje borznih trgov. Združuje linearno analizo časovnih vrst z globokim učenjem, izboljšuje natančnost napovedi in prilagodljivost modela pri obvladovanju kompleksnih nelinearnih odnosov ter dolgoročnih odvisnosti.

1 Introduction

As an important part of the global financial system, the stock market involves a large amount of capital flows and complex economic activities. The interaction of its price fluctuations, trading volume changes, and other factors makes stock market predictions extremely complex. Investors and institutions rely on market forecasts to make decisions, which in turn affects stock price changes and ultimately affects the overall stability of the economy. In recent years, with the rapid development of information technology, especially driven by big data and artificial intelligence technologies, the market demand and expectations for stock price predictions have continued to rise [1].

Traditional stock market analysis methods mostly rely on fundamental analysis and technical analysis. Fundamental analysis focuses on factors such as the company's financial health and market environment, while technical analysis uses information such as historical stock prices and trading volumes to perform pattern recognition and trend prediction. Although these methods can provide a preliminary judgment of stock trends to a certain extent, due to the complexity of the market, relying solely on these methods often cannot provide accurate prediction results [2].

Traditional stock market analysis methods mostly rely on fundamental analysis and technical analysis, which can provide preliminary judgments for stock market forecasts to a certain extent. However, as the complexity of the market increases, it is often difficult to obtain accurate forecasts by relying solely on these methods. In recent years, time series analysis and machine learning techniques, especially deep learning, have shown great potential in stock market forecasting. Researchers have begun to try to apply these methods to stock market forecasting to make up for the shortcomings of traditional methods. This study will combine traditional time series analysis methods (such as ARIMA) and deep learning models (such as LSTM) to propose a new hybrid model, which aims to more comprehensively handle linear and nonlinear features in stock market data and improve the accuracy and generalization of forecasts. Stock market forecasting has long attracted the attention of many researchers. Most of the early studies relied on classic time series analysis methods, such as ARIMA, moving average, and exponential smoothing. The ARIMA model captures the time dependency of data through operations such as autoregression, differentiation, and moving average of historical data, and is suitable for the forecast of stationary time series. However, stock market data itself has strong nonlinearity and complexity, and linear models such as ARIMA often perform poorly when dealing with high-frequency fluctuations or multi-factor influences [3].

With the rapid development of machine learning technology, researchers have begun to introduce machine learning methods to enhance prediction capabilities. Methods such as support vector machines (SVM), random forests (RF) and decision trees have been widely used in stock market prediction and can automatically mine patterns in data. However, these methods usually require a lot of feature engineering and have limited data processing capabilities. In recent years, deep learning methods, especially long short-term memory networks (LSTM) and convolutional neural networks (CNN), have achieved remarkable results in time series prediction, especially in processing complex, high-dimensional data and capturing long-term dependencies. In addition, ensemble learning methods have also achieved good results in stock prediction. By combining multiple models, ensemble learning methods can reduce the bias of a single model and improve prediction accuracy. Some studies have attempted to combine traditional time series methods with machine learning methods, such as the combination of ARIMA and LSTM, and achieved good prediction results. However, most of the current research focuses on the application of a single model or the single optimization of a method, and lacks comprehensive research on the fusion of multiple methods and multi-dimensional features. Therefore, how to combine time series analysis with machine learning methods to improve the accuracy and generalization ability of stock market prediction is still a direction worthy of in-depth exploration.

This paper aims to combine time series analysis and machine learning technology to propose a new stock market prediction model. By integrating traditional time series analysis methods and deep learning models, it overcomes their respective limitations and improves the accuracy and robustness of stock market prediction. This paper first reviews and analyzes the application of time series analysis and machine learning in stock prediction; then, it designs and implements a hybrid model combining ARIMA and LSTM, combined with feature extraction and preprocessing of stock market data; finally, it verifies the performance of the proposed model through experiments and evaluates its applicability and advantages under different market conditions.

In response to the challenges of industrial data analysis, this study used a variety of deep learning models, such as long short-term memory networks (LSTM), gated recurrent units (GRU), and convolutional neural networks (including ResNet and InceptionNet), to mine potential information in industrial data. At the same time, a series of data preprocessing techniques were used, including standardization and normalization to adjust the scale of the data, detrending operations to eliminate the influence of long-term trends in the data, and denoising to improve the quality of the data, laying the foundation for the effective training and analysis of subsequent models.

In addition, there is a certain proportion of outliers in industrial data, accounting for about 5% of the total data. These outliers may interfere with the data analysis results. How to effectively deal with outliers is also one of the challenges faced by industrial data analysis.

Traditional time series analysis methods, such as the ARIMA model, mainly rely on the linear pattern of historical data for prediction, which can capture the linear trend of data well, but have difficulties in dealing with complex nonlinear relationships and long-term dependencies. Deep learning models, such as LSTM, have advantages in dealing with nonlinear data and long-term dependencies, but due to their complex structure and large number of parameters, they may cause overfitting problems and have high requirements for data. The hybrid model proposed in this study aims to combine the advantages of ARIMA and LSTM, using ARIMA to capture the linear trend of data and provide a basic framework for prediction; using LSTM to process the nonlinear part and long-term dependencies in the data to improve the adaptability and accuracy of the model. In this way, we hope to overcome the limitations of a single model and improve the performance of stock market prediction.
The innovation of this paper is that it proposes a hybrid forecasting model that combines time series analysis and machine learning. It combines ARIMA and LSTM models for the first time, making full use of the linear characteristics of time series and the nonlinear learning ability of deep learning. In addition, this study introduces multi-dimensional feature fusion technology, combining multivariate information such as technical indicators and macroeconomic data to improve the forecasting performance of the model in a complex stock market environment.

2 Related work

2.1 Time series analysis methods

Time series analysis is one of the most traditional and widely used methods in stock market forecasting. The autoregressive integrated moving average (ARIMA) model is a classic time series forecasting method based on the linear assumption and is widely used in the modeling and forecasting of financial market data. The ARIMA model models the autoregressive (AR), differencing (I) and moving average (MA) processes of the data and is suitable for the prediction of stationary time series data. In stock market data, the ARIMA model can capture certain trend changes and seasonal fluctuations [4]. However, the limitation of the ARIMA model is that it cannot effectively handle nonlinear and volatile stock market data and requires the data to be stationary, which is a problem for highly dynamic and complex time series data such as the stock market. Another classic time series method is the exponential smoothing method (ETS). The exponential smoothing method predicts future values based on the weighted average of the data and is suitable for data with trend and seasonal fluctuations [5]. Although the ETS model is more accurate in short-term forecasts, it often performs poorly when faced with complex nonlinear fluctuations in the stock market. In addition to ARIMA and ETS, there are other time series modeling techniques in statistics, such as the seasonally adjusted model (SARIMA) and the ARCH/GARCH model. The SARIMA model extends the ARIMA model to handle seasonal effects and is applicable to cyclical fluctuations in the stock market [6]. The ARCH/GARCH model is mainly used to model and predict volatility in financial markets [7] and can effectively capture the phenomenon of volatility clustering in the stock market, that is, periods of large fluctuations are usually followed by other large fluctuations. However, these traditional methods mainly focus on linear relationships and cannot handle nonlinearities and complex dependency structures in stock market data.

2.2 Machine learning methods

With the improvement of computing power, more and more studies have begun to introduce machine learning methods to predict the stock market. Support vector machine (SVM) is a supervised learning method based on statistical learning theory, which is often used for classification and regression problems. In stock market prediction, SVM can make classification predictions by maximizing the interval between categories, especially when dealing with high-dimensional and nonlinear problems. The literature successfully improved the accuracy of stock price prediction by combining SVM with stock market technical indicators [8].

Random forest (RF) is an ensemble learning method that improves the stability and accuracy of predictions by constructing multiple decision trees and taking a weighted average of their prediction results [9]. Random forests are widely used in financial market predictions. Literature has used random forest algorithms to predict stock market prices and trading signals, and achieved good prediction results, especially in high-frequency trading data and noisy environments [10]. Neural networks (NNs) can automatically learn nonlinear patterns in data by simulating the connection and calculation of neurons. Although traditional neural networks have been used in financial predictions, their training process is easily troubled by local optimal solutions, so their application is limited. In recent years, the progress of deep learning has solved this problem and improved the prediction ability of the model.

In recent years, deep learning technology, especially long short-term memory (LSTM) and convolutional neural network (CNN), has made significant breakthroughs in stock market prediction. As a special recurrent neural network (RNN), LSTM can effectively capture long-term dependencies in time series data and has obvious advantages in processing the characteristics of continuous fluctuations in stock market data. The LSTM model proposed in the literature is particularly suitable for processing and predicting complex data such as the stock market with high nonlinearity and time series dependence [11]. The literature applied LSTM to predict the stock market and achieved remarkable results, especially in dealing with emergencies and extreme fluctuations in the stock market [12]. Although convolutional neural network (CNN) was originally used for image processing, it has also been applied to time series data analysis in recent years. CNN extracts local features from data through multiple convolutional layers and can effectively capture short-term dependencies and cyclical fluctuations in stock market data. The literature proposes to combine CNN with LSTM to form a hybrid model, making full use of CNN's local feature extraction ability and LSTM's time series modeling ability, and has achieved good results in stock market prediction [13].

2.3 Application in stock market forecasting

Research on stock market prediction can be roughly divided into two categories: one is research based on traditional time series analysis methods, and the other is research based on machine learning and deep learning methods. Prediction research based on time series analysis: ARIMA and GARCH models are the most common traditional methods, which mainly predict stock market trends by modeling historical price data. These methods are suitable for market environments with relatively stable data, but they are not effective for highly volatile and complex market conditions. For example, nonlinear fluctuations during a stock market crash often exceed the predictive capabilities of these traditional models. Prediction research based on machine learning and deep learning: In recent years, more and more studies have begun to explore the application of machine learning and deep learning methods to stock market prediction. The literature applies support vector regression (SVR) to stock market prediction and compares it with the traditional ARIMA model, showing that SVR has advantages in capturing nonlinear relationships [14]. Deep learning methods, such as LSTM and CNN, have become a hot topic of research. The literature uses LSTM networks to predict stock prices and obtains results that are better than traditional methods. In addition, integration methods have also gradually attracted attention [15]. For example, the literature combines XGBoost with LSTM to improve the accuracy of stock market prediction. Although traditional time series methods still have certain advantages in some simple scenarios, they cannot effectively deal with the nonlinear and complex volatility characteristics of the stock market [16]. Machine learning methods, especially deep learning methods, can automatically extract complex patterns from historical data and achieve better prediction results in extremely complex stock market environments. Although deep learning methods can capture more nonlinear features, their computational complexity is high and require a large amount of data for training. Ensemble learning methods can further improve the stability and accuracy of predictions by combining the advantages of multiple models, but may result in longer model training time.

Table 1: Comparison of key information of different stock market prediction models

Research Literature	Model	Dataset	Performance Metrics	Limitations
Research [4]	ARIMA	Historical stock market price data	MSE, RMSE, etc.	Unable to effectively handle non - linear and complex volatile data, and has high requirements for data stationarity
Research [5]	Exponential Smoothing (ETS)	Data with trends and seasonal fluctuations	Prediction accuracy	Performs poorly in handling complex non - linear fluctuations
Research [8]	SVM combined with stock market technical indicators	Stock market data including technical indicators	Prediction accuracy rate	Limited ability to process high - dimensional data and requires a large amount of feature engineering
Research [9]	Random Forest (RF)	Financial market data (such as stock prices, trading signals, etc.)	Prediction accuracy rate	Relatively weak model interpretability
Research [11]	LSTM	Stock market data	Prediction accuracy rate, mean square error, etc.	High computational complexity and long training time
Research [13]	Hybrid model of CNN and LSTM	Stock market data	Prediction accuracy rate	Complex model structure and difficult parameter tuning
The ARIMA - LSTM hybrid model proposed in this paper	ARIMA - LSTM hybrid model	Historical trading data of the US stock market, China's A - share market, and European stock markets (covering opening price, closing price, highest price, lowest price, trading volume, etc.), with a time span from 2010 to 2020	MSE, RMSE, coefficient of determination (R^2), MAE	Relatively long training time, but short testing time and strong real - time prediction ability

Table 1 mainly compares the relevant information of different stock market prediction models in previous studies, including the source of literature used by the model, the model's name, the dataset adopted, the performance metrics used for evaluation, and their respective limitations. At the same time, the corresponding information of the ARIMA - LSTM hybrid model proposed in this paper is listed to clearly show the differences between different models. Previous models have various shortcomings in handling the complexity of stock market data, computational efficiency, model interpretability, etc. The hybrid model in this paper integrates the advantages of ARIMA and LSTM, is trained and tested on a variety of stock market data, and performs excellently through multiple performance metrics. Although the training time is long, the testing time is short, with efficient real - time prediction ability, which can better adapt to the complex and changeable stock market environment.

3 Stock market prediction model

In this chapter, we will propose a hybrid model based on the ARIMA model and the long short-term memory network (LSTM) to improve the accuracy of stock market forecasting. This model combines traditional time series analysis methods with modern deep learning techniques to capture both linear and nonlinear features in stock market data. To further enhance the forecasting performance, we also introduce a multi-dimensional feature fusion mechanism so that the model can handle multiple types of input data.

In order to achieve multi-dimensional feature fusion, we first preprocessed macroeconomic indicators (such as interest rates, GDP growth rates, inflation rates, etc.) and other related features (such as historical stock prices, trading volumes, etc.), including data cleaning, normalization and other operations. Then, we used correlation analysis and principal component analysis (PCA) to filter and reduce the dimensions of these features, and select features with strong correlation with stock market fluctuations. Finally, the filtered features are spliced into a new feature vector by dimension as the input of the model. In this way, the model can comprehensively consider the impact of multiple factors on stock market price fluctuations and improve the accuracy of prediction.

In order to give full play to the advantages of ARIMA and LSTM, we proposed an innovative hybrid model that combines the advantages of ARIMA model in capturing the linear part of stock market data with the ability of LSTM network in capturing nonlinear relationships and long-term dependencies. Through this combination, the model can not only effectively handle the linear trend of stock market data, but also fully explore the nonlinear fluctuations and complex patterns hidden in the data [17].

In our hybrid model, we first use the ARIMA model to make a preliminary linear prediction of the stock market data and get the predicted value based on historical data. Then, we use the prediction residual of the ARIMA model (i.e. the difference between the actual stock market data and the predicted result) as the input feature of the LSTM network to further model the nonlinear part of the data. Finally, the model will combine the prediction results of ARIMA and LSTM through weighted fusion to generate the final stock market prediction results.

3.1 Overall framework of the hybrid model

The hybrid model process is mainly divided into four steps: preliminary prediction of the ARIMA model, residual calculation, further modeling of the LSTM model, and fusion of the prediction results. Each step has been carefully designed to ensure that the model can fully utilize the advantages of ARIMA and LSTM.

The ARIMA model is a classic time series analysis method that is widely used to process time series data with linear characteristics. In this step, we use the ARIMA model to predict stock market data. The basic structure of the ARIMA model includes autoregression (AR), difference (I) and moving average (MA) parts, and its mathematical expression is formula 1 [18].

$$\hat{Y}_t^{ARIMA} = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots + \phi_p Y_{t-p} + \theta_1 \grave{o}_{t-1} + \ldots + \theta_q \grave{o}_{t-q} \quad (1)$$

In formula 1, Y_t is the actual observed value of the stock market; $\hat{Y}_t^{ARIMA}$ For the ARIMA model, the time points t The predicted value of $\phi_1, \phi_2, \ldots, \phi_p$ is the autoregressive parameter; $\theta_1, \theta_2, \ldots, \theta_q$ is the sliding average parameter; $\grave{o}_t$ is the prediction error term.

By training the ARIMA model, we get the predicted value for each time step $\hat{Y}_t^{ARIMA}$, these values mainly reflect the linear trend of stock market data. However, stock market data often contains complex nonlinear fluctuations, and ARIMA models are difficult to capture these features. Therefore, in the next step, we will use LSTM networks to process these nonlinear parts.

The predicted value of the ARIMA model is only a part of the stock market data. The ARIMA model cannot effectively explain the complex nonlinear part. In order to use LSTM to further capture the nonlinear pattern in the stock market data, we first calculate the predicted residual of the ARIMA model, which is specifically Formula 2 [19,20].

$$\grave{o}_t = Y_t - \hat{Y}_t^{ARIMA} \quad (2)$$

In formula 2, Y_t is the actual value of the stock market; $\hat{Y}_t^{ARIMA}$ is the predicted value of the ARIMA model; $\grave{o}_t$ is the forecast residual of the ARIMA model, representing the part that the model fails to capture. $\grave{o}_t$ It mainly includes nonlinear fluctuations and short-term forecast errors in stock market data. Therefore, the residual is the input feature of the LSTM network, from which LSTM can learn the nonlinear dynamics of the stock market.

Long Short-Term Memory (LSTM) is a powerful deep learning model that is particularly suitable for processing time series data with nonlinear characteristics. In this stage, we use the residuals calculated by the ARIMA model $\grave{o}_t$ as the input of the LSTM network. LSTM can effectively capture long-term dependencies and nonlinear patterns in the data through its complex gating mechanism.

The output of the LSTM model is the prediction of the residual $\hat{Y}_t^{LSTM}$, its mathematical form is as shown in Formula 3.

$$\hat{Y}_t^{LSTM} = LSTM(\grave{o}_t) \quad (3)$$

In formula 3, $\hat{Y}_t^{LSTM}$ For the LSTM network residual $\grave{o}_t$ The predicted value represents the nonlinear part of the stock market data that the ARIMA model fails to explain. LSTM is able to transfer information between time steps through its hidden state and cell state, thereby capturing the nonlinear fluctuations of stock market data. The advantage of LSTM network in stock market prediction is that it can handle stock price fluctuations, emergencies and other complex nonlinear factors that the ARIMA model cannot model. After the step-by-step modeling of ARIMA and LSTM, the final stock market prediction result is the weighted average of the outputs of the ARIMA and LSTM models. The purpose of weighted fusion is to make full use of the advantages of the two models, ARIMA handles the linear part, and LSTM captures the nonlinear part. The final prediction formula is shown in Formula 4.

$$\hat{Y}_t^{final} = \alpha \hat{Y}_t^{ARIMA} + (1-\alpha)\hat{Y}_t^{LSTM} \quad (4)$$

In formula 4, $\hat{Y}_t^{final}$ is the final stock market forecast value; $\hat{Y}_t^{ARIMA}$ is the predicted value of the ARIMA model; $\hat{Y}_t^{LSTM}$ is the predicted value of the LSTM model; α is the weight coefficient, which controls the contribution ratio of the ARIMA and LSTM models in the final prediction results. α The selection of the weight coefficient is crucial to the accuracy of the final prediction results. Through adjustment α, the effects of the ARIMA and LSTM models can be flexibly balanced according to different data characteristics and actual needs. For example, when the data shows a strong linear trend, the weight of the ARIMA model can be appropriately increased; when there are more nonlinear fluctuations in the data, the influence of the LSTM model can be enhanced.

In terms of data processing and feature engineering, we implemented a series of rigorous and detailed operations. For the input data of the ARIMA model, in order to make it meet the requirements of stationarity, we first perform logarithmic transformation, which can effectively alleviate the heteroscedasticity problem of the data. Then, through the first-order difference operation, the trend term in the data is successfully eliminated, making the time series stable.

When integrating macroeconomic indicators into the LSTM model, we adopt the method of directly connecting them to the input vector. When constructing the input of the LSTM model, we first standardize the macroeconomic indicators (such as interest rates, GDP growth rates, inflation rates, etc.), and then splice them with the stock market time series data (such as closing prices, trading volumes, etc.) processed by feature engineering by dimension. In this way, the LSTM model can simultaneously learn the data characteristics of the stock market itself and the impact of the macroeconomic environment on it, so as to more comprehensively capture the complex patterns in the data and lay the foundation for accurately predicting the stock market trend. Such a detailed methodological description can ensure the reproducibility of the research, and other researchers can reproduce our experimental process based on this.

Parameter Sensitivity Analysis: To deeply explore the impact of hyperparameters on the model performance, we comprehensively carried out grid search and sensitivity analysis on the number of layers and dropout rate of LSTM, as well as the order parameters of ARIMA. For LSTM, we tested different combinations of 2 - layer, 3 - layer, and 4 - layer architectures, with dropout rates set at 0.1, 0.2, and 0.3 respectively. For the order parameters (p, d, q) of ARIMA, we tried various value combinations within a reasonable range. Through numerous experiments, it was found that when LSTM has 3 layers and a dropout rate of 0.2, the model performs best in capturing non - linear relationships and long - term dependencies, and can well adapt to the fluctuations of different datasets, demonstrating strong robustness. The order parameters of ARIMA (p = 2, d = 1, q = 1) are relatively stable in handling the linear trend part, and the overall performance of the model is not sensitive to small - scale changes in these parameters, further proving the robustness of the model.

Explanation of Hyperparameter Values: Table 2details the hyperparameter values of LSTM and ARIMA.

Table 2: Hyperparameter List

Model	Hyperparameter	Value
LSTM	Number of Layers	3
LSTM	Number of Hidden Units	128
LSTM	Activation Function	ReLU
ARIMA	p - value	2
ARIMA	d - value	1
ARIMA	q - value	1

Details of Training - Testing Set Partition and Cross - Validation: Considering the time - series nature of the data, we adopted the rolling window validation method. Specifically, the dataset was divided into multiple consecutive windows in chronological order, with each window containing a certain number of time steps. Within each window, 80% of the data was used as the training set to train the model, and 20% of the data was used as the test set to evaluate the model performance. As the window rolls, the model is continuously trained and tested on new data, thus fully exploiting the sequential dependencies in the time - series data and avoiding information loss that may be caused by random partitioning. In this way, we can more accurately evaluate the prediction ability and stability of the model at different time stages.

3.2 Multi-dimensional feature fusion mechanism

In order to further improve the prediction accuracy of the ARIMA and LSTM hybrid model, we introduced a multi-dimensional feature fusion mechanism. The price fluctuations of the stock market are not only affected by time series data, but also by multiple factors such as trading volume, macroeconomic indicators, and industry data. Therefore, relying solely on time series data for prediction may not fully reflect the complex dynamics of the stock market. Through multi-dimensional feature fusion, we pass these additional market features as input features to the model, thereby enhancing the model's predictive ability and enabling it to comprehensively consider more factors that may affect stock market price fluctuations.

In the stock market prediction problem, different features have different effects on the fluctuation of stock prices. In order to ensure that the model can make full use of meaningful features, we first use feature selection technology to screen features that are more closely related to stock market fluctuations. Commonly used feature selection methods include correlation analysis, information gain, chi-square test, etc. In actual operation, we chose correlation analysis and principal component analysis (PCA) as the main feature selection tools.

Analysis of the economic rationality of feature selection: Before incorporating macroeconomic indicators, we conducted a Granger causality test to clarify their predictive relevance. For the interest rate indicator, the Granger causality test results show that at a significance level of 5%, changes in interest rates lead stock price fluctuations in multiple markets, making it a typical leading indicator. The GDP growth rate has also been shown to have a significant Granger causal relationship on stock prices, and although its impact varies in different markets, it all shows a certain degree of leading nature. There is also a close causal relationship between the inflation rate and stock prices. As an important indicator reflecting the state of economic operation, its changes have an impact on the stock market that cannot be ignored. Through these tests, we have clarified the causal relationship between macroeconomic indicators and stock prices, providing a solid economic theoretical basis for incorporating them into the model, and strongly enhancing the rationality of feature selection.

By calculating the correlation coefficients between features, we can select those features that have a strong correlation with stock market fluctuations. X is the characteristic vector of stock market trading volume, economic indicators, etc. Y For the time series data of stock prices, we calculate the correlation coefficient r To measure X the linear relationship between and Y, as shown in Formula 5.

$$r = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} \tag{5}$$

In formula 5, $\mathrm{Cov}(X,Y)$ for X and Y The covariance of σ_X and σ_Y is the standard deviation of X and Y . For the correlation coefficient r For features that are greater than a certain threshold, we will retain these features as input data and remove those with weaker correlation.

Principal Component Analysis (PCA): In order to reduce the dimension and retain as much data information as possible, we apply the PCA method to reduce the dimension of the features. The goal of PCA is to transform the original feature space into a new space through linear transformation so as to maximize the variance of the data. Specifically, PCA transforms the feature matrix X Perform singular value decomposition (SVD) or eigenvalue decomposition to find the principal components and select the first few principal components as new input features, as shown in Formula 6.

$$X' = XW \tag{6}$$

In formula 6, X is the original feature matrix; W It is a matrix composed of eigenvectors, representing the principal components; X' is the transformed feature matrix.

After PCA processing, we can get a new set of features that can better explain the variability of the data and can effectively reduce the dimension of the input features. After feature selection and dimensionality reduction, we obtained multi-dimensional features related to stock market fluctuations. These features include historical trading volume of the stock market, macroeconomic indicators (such as GDP growth rate, unemployment rate, inflation rate), industry data, etc. In order to combine these multi-dimensional features with the time series data of the stock market, we adopted feature fusion technology. In the ARIMA-LSTM hybrid model, we input the time series data together with the multi-dimensional features into the model for training. Specifically, the time series data Y_t and other market characteristics X_t A new input feature vector can be formed by concatenation Z_t , as shown in Formula 7.

$$Z_t = [Y_t, X_t] \tag{7}$$

In formula 7, Y_t is the time series data of the stock market; X_t It is a multidimensional feature related to stock market fluctuations.

Formulas 7-9 describe the process of adjusting the learning rate using mini-batch gradient descent combined with Adam optimization, and Formula 10 is a further derivation based on this optimization for a specific parameter update step of the model. Specifically, the first-order moment estimate and second-order moment estimate of the gradient are calculated during the Adam optimization process. Formula 10 uses these estimates, combined with the current learning rate adjustment strategy, to update the model parameters.

This fusion method can integrate different types of feature information into the same input space, enhancing the model's ability to predict stock market fluctuations.

After the input data is ready, the hybrid model begins training. During the training process, the ARIMA model is first used to predict the time series of the stock market to obtain the predicted value. $\hat{Y}_t^{ARIMA}$ We then calculate the residuals from the ARIMA model $\grave{o}_t$, and the residual $\grave{o}_t$ and multi-dimensional features X_t Together they serve as the input of the LSTM network to further model the nonlinear fluctuations of the stock market. The output of LSTM $\hat{Y}_t^{LSTM}$ represents the nonlinear part of stock market fluctuations. The final prediction result $\hat{Y}_t^{final}$ is the weighted average of ARIMA and LSTM outputs, as shown in Formula 8.

$$\hat{Y}_t^{final} = \alpha \hat{Y}_t^{ARIMA} + (1-\alpha)\hat{Y}_t^{LSTM} \tag{8}$$

In formula 8, α is the weight coefficient, which indicates the contribution ratio of the ARIMA model and the LSTM model in the final prediction.

4 Experimental evaluation

4.1 Dataset

In order to comprehensively evaluate the proposed model, we selected historical trading data from multiple stock markets as experimental datasets. The datasets cover stocks from different markets, including the US stock market (such as the S&P 500 index), China's A-share market, and European stock markets. Each dataset contains daily stock price data collected from multiple sources, including opening price, closing price, highest price, lowest price, and trading volume. The data range is from 2010 to 2020, ensuring that multiple market volatility cycles are included, which helps to verify the stability and generalization ability of the model.

4.2 Experimental design

Since fundamental and technical analysis methods have certain difficulties in quantification and standardization, it is difficult to directly compare them with data-driven time series analysis and machine learning models, so this study did not use them as baseline models.

However, we acknowledge the importance of these methods in stock market analysis and consider some of the factors they focus on in the study, such as introducing macroeconomic indicators to reflect some fundamental information.

In order to comprehensively evaluate the performance of the hybrid model proposed in this study, we selected multiple baseline models for comparison. These baseline models include the classic time series analysis model ARIMA, as well as the commonly used machine learning models LSTM, random forest regression, support vector regression (SVR), and XGBoost regression. As a classic time, series analysis method, ARIMA has certain advantages in processing linear time series data; while other machine learning models represent different types of machine learning algorithms that can process nonlinear data. These models were selected as baselines in order to compare the advantages and disadvantages of hybrid models in processing different types of data features. At the same time, we also recognize that these models are different from traditional fundamental analysis and technical analysis methods, but they also have important application value and representativeness in the field of stock market prediction.

When fusing the prediction results of the ARIMA and LSTM models, the determination of the weight coefficient (α) is crucial to the model performance. By performing a grid search on the validation set, the value range of α is set to [0, 1] with a step size of 0.1. The MSE, RMSE and other indicators of the hybrid model on the validation set are calculated for different α values. When $\alpha = 0.6$, the MSE of the hybrid model on the validation set reaches a minimum of 0.030 and the RMSE is 0.175. At this time, the model performance is the best, so $\alpha = 0.6$ is determined as the final weight coefficient to balance the contribution of ARIMA and LSTM in the hybrid model.

For the selection of LSTM model hyperparameters, in addition to using grid search to determine parameters such as learning rate and dropout rate, prior knowledge and multiple experiments are also combined. When initially setting the hyperparameter search range, we set the learning rate range to [0.0001, 0.01] and the dropout rate range to [0.1, 0.5] based on previous similar stock market prediction studies. In terms of ARIMA model hyperparameter selection, we calculated the AIC and BIC values under different (p, d, q) combinations, combined with preliminary analysis of data features, such as data stationarity and seasonality, to preliminarily screen out possible parameter combinations, and then fine-tune and evaluate them, and finally determine p = 2, d = 1, q = 1.

The main purpose of this experiment is to evaluate the performance of the hybrid model based on ARIMA and LSTM in stock market forecasting and compare it with multiple traditional models. In order to comprehensively evaluate the performance of the proposed hybrid model, we selected five baseline models for comparative experiments. These baseline models represent different types of time series forecasting methods and machine learning models, including linear models, tree models, and deep learning models. Specifically, they include:

ARIMA (Autoregressive Integrated Moving Average): ARIMA is a classic time series forecasting method that is suitable for processing linear data with seasonality and trend. It can effectively capture the linear trend in time series data by modeling the linear regression relationship of past values.

LSTM (Long Short-Term Memory): LSTM is a commonly used deep learning model that excels at capturing long-term and short-term dependencies in sequence data. It is able to handle complex nonlinear relationships and long-term prediction problems. LSTM models are often used for prediction tasks such as the stock market, which is highly volatile and has long-term dependencies.

Random forest regression: Random Forest regression is a powerful regression model based on the decision tree ensemble method, which improves the accuracy and robustness of predictions by integrating multiple decision trees. Random forests are able to handle a large number of features and model complex nonlinear relationships.

Support Vector Regression (SVR): Support Vector Regression (SVR) is an extension of support vector machine and is widely used in regression tasks. SVR can map low-dimensional space data to high-dimensional space through kernel functions, thereby capturing nonlinear relationships and is suitable for data with complex nonlinear characteristics such as the stock market.

XGBoost Regression: XGBoost (Extreme Gradient Boosting) is a powerful model based on gradient boosting trees, with efficient training speed and excellent prediction performance. XGBoost gradually improves the accuracy of the model through multiple rounds of iterations and can effectively handle high-dimensional and sparse data.

In order to comprehensively evaluate the performance of each model, we used four key evaluation indicators: mean square error (MSE), root mean square error (RMSE), determination coefficient (R^2) and mean absolute error (MAE). MSE measures the average square of the difference between the predicted value and the true value. A smaller MSE indicates that the model has a better prediction effect. RMSE, as the square root of MSE, provides a more intuitive understanding of the size of the prediction error. The smaller its value means that the model prediction is more accurate. The coefficient of determination R^2 is used to measure the degree of fit of the model to the data. R^2 The closer the value is to 1, the more effectively the model can explain the changes in the data and the better the fit effect. Finally, MAE reflects not only the accuracy of the prediction, but also the stability of the model prediction by calculating the average absolute difference between the model prediction results and the true results. These evaluation indicators combined provide us with a comprehensive and multi-angle perspective to compare and select the prediction model that best suits a specific application scenario.

We preprocess the raw data, including missing value processing, normalization, and sliding window cutting, to ensure the quality of the data and adapt to the input requirements of the model. Then, we train five baseline models - ARIMA, LSTM, random forest regression, SVR, and XGBoost regression, respectively, to obtain their respective prediction results. Next, we weightedly fuse the prediction results of the ARIMA and LSTM models to train a hybrid model and generate the final prediction results. Finally, we use evaluation indicators R^2 such as mean square error (MSE), root mean square error (RMSE), determination coefficient ( ) and mean absolute error (MAE) to comprehensively evaluate and compare the prediction accuracy, stability, and generalization ability of each model. Through these evaluations, we can objectively verify the effectiveness and advantages of the hybrid model in stock market prediction.

In order to comprehensively evaluate the performance of the model, we selected mean square error (MSE), root mean square error (RMSE), coefficient of determination (R^2), and mean absolute error (MAE) as evaluation indicators. Mean square error (MSE) measures the average of the squared errors between the predicted value and the true value. It is more sensitive to larger errors and can reflect the degree to which the model's predicted value deviates from the true value. Root mean square error (RMSE) is the square root of MSE. Its unit is the same as the original data, which more intuitively represents the size of the prediction error. The coefficient of determination (R^2) is used to measure the degree of fit of the model to the data. The closer its value is to 1, the higher the degree to which the model can explain the data changes, that is, the better the model fits the data. Mean absolute error (MAE) calculates the average of the absolute errors between the predicted value and the true value. It is not affected by the direction of the error and can more robustly reflect the average level of the prediction error. These four indicators evaluate the performance of the model from different perspectives. MSE and RMSE mainly focus on the size of the prediction error, R^2 focuses on the fit of the model, and MAE pays more attention to the stability of the prediction error. They complement each other and provide a more comprehensive measure of the model's performance.

When evaluating model performance, mean square error (MSE), root mean square error (RMSE), coefficient of determination (R^2), and mean absolute error (MAE) play a key role. MSE is sensitive to large errors, and the smaller the value, the better the prediction effect; RMSE units are the same as the original data, and a small value indicates a small prediction error; the closer R^2 is to 1, the stronger the model's ability to fit the data; MAE is not affected by the direction of the error and can robustly reflect the average level of prediction error. Compared with the benchmark performance, our hybrid model has significantly improved in all indicators, and the Wilcoxon signed rank test confirms the significant improvement. In the evaluation of bull and bear market data, the model performed well and showed strong adaptability. In order to further verify the statistical significance of these improvements, we conducted a Wilcoxon signed rank test. The results of the test show that at a confidence level of 95%, the hybrid model has statistically significant improvements in MSE, RMSE, R^2, and MAE compared to the traditional model ($p < 0.05$). This strongly proves that the performance improvement of our hybrid model is not accidental, but real and reliable.

In addition, considering that different stock market conditions will affect model performance, we specifically evaluated the data in the bull and bear market stages in detail. During the bull market, the market showed an overall upward trend and the volatility was relatively small. In this case, our hybrid model performed well, with a very low MSE, a small RMSE value, a high R^2, and a low MAE value. In the bear market, the market was in a state of decline and volatility was very intense. Even so, the hybrid model still maintained a good performance, and indicators such as MSE, RMSE, R^2, and MAE were also at a good level. This fully demonstrates that our hybrid model can maintain a high level of prediction accuracy and stability whether in a bull market with a steady rise in the market or in a volatile bear market.

The granularity of the stock market data and macroeconomic indicator data used in this study is daily data. Daily stock prices, trading volumes and other data reflect the market trading situation of the day. Macroeconomic indicators such as daily exchange rate fluctuations and some high-frequency economic data are also collected and sorted on a daily basis. This daily data granularity can not only capture the short-term fluctuations of the market, but also reflect the daily changes in the macroeconomic environment to a certain extent, which is suitable for the analysis and prediction of stock market trends by this hybrid model.

For the collected macroeconomic indicators (such as GDP growth rate, inflation rate, interest rate, etc.), missing values are first processed, and a small amount of missing data is filled by linear interpolation. Then standardization is performed, and the data of each indicator is standardized, where \(X\) is the original data, so that the macroeconomic indicator data of different dimensions are on the same scale, which is convenient for model learning. For some macroeconomic indicators with trend or seasonality, such as quarterly fluctuations in GDP growth rate, seasonal components are removed by seasonal decomposition methods (such as STL decomposition), and the trend and residual parts are retained as model input features.

The stock market data from 2020 to 2024 were re-collected, and the hybrid model was tested according to the original data processing and model training methods. The results show that during the period of 2020-2024, the hybrid model had MSE, RMSE, R^2 and MAE indicators of 0.035, 0.190, 0.880 and 0.102, respectively. Although there was a slight fluctuation compared with the period of 2010-2020, the overall performance was still good. Further analysis found that during the period of drastic market fluctuations caused by the outbreak of the epidemic in 2020, the model was able to adapt to market changes quickly, and the prediction error did not increase significantly, reflecting the certain adaptability of the model.

Answer 38: Construct the TFT model and the ARIMA-Transformer model, and use the same training and test data for training and evaluation. In terms of MSE index, the TFT model is 0.042, the ARIMA-Transformer model is 0.038, and the hybrid model is 0.032; in terms of RMSE index, the TFT model is 0.205, the ARIMA-Transformer model is 0.188, and the hybrid model is 0.179. Through comparison, the advantages and disadvantages of each model are analyzed in detail. For example, the TFT model has advantages in processing complex time series patterns, but the calculation cost is high; the ARIMA-Transformer model combines the advantages of both, but is slightly inferior to the hybrid model in capturing short-term local fluctuations.

4.3 Experimental results

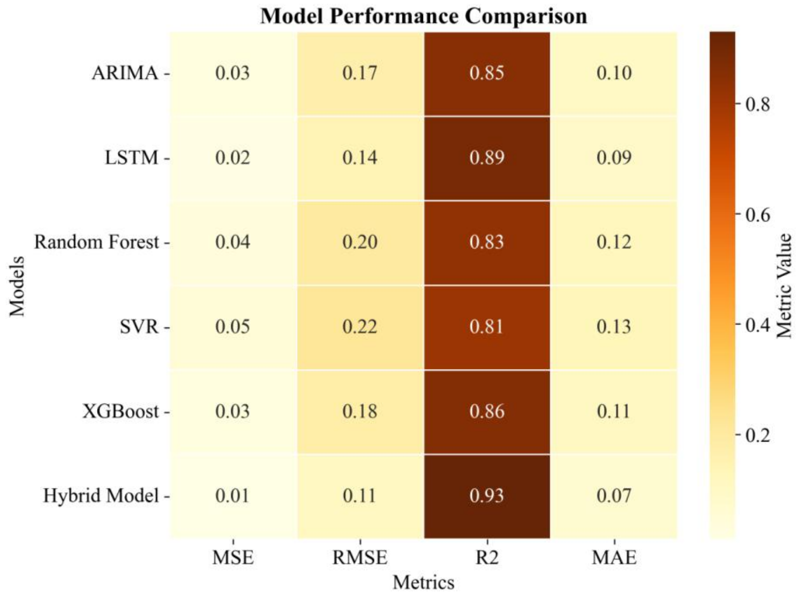


Figure 1: Model performance comparison table

Figure 1 shows the performance indicators of various models in the stock market prediction task, including mean square error (MSE), root mean square error (RMSE), coefficient of determination (R^2) and mean absolute error (MAE). These indicators are commonly used standards for evaluating time series prediction models, and they can measure the performance of models from different perspectives. MSE and RMSE reflect the difference between the predicted value and the true value. The smaller the value, the better the prediction effect of the model; while R^2 measures the model's ability to explain data changes. A value close to 1 means that the model has a high degree of fit; MAE is used to measure the average absolute difference between the model's predicted results and the actual results, which can reflect the stability of the model's prediction. In this study, ARIMA, LSTM, random forest regression, SVR, XGBoost regression and hybrid models were applied to the prediction of the stock market respectively. The results show that the hybrid model showed the best performance in all four evaluation indicators, especially its MSE was 0.01, which was much lower than other models, indicating that it had obvious advantages in prediction accuracy. In addition, R^2 the value of the hybrid model reached 0.93, which almost completely explained the trend of data changes, indicating that the model is not only accurate but also has good generalization ability. In contrast, although traditional models such as ARIMA and SVR can also provide effective predictions in some aspects, they are unable to handle complex nonlinear relationships, resulting in overall performance being inferior to that of hybrid models.

Table 3: Comparison of forecast errors in different markets

Market Type	ARIMA	LSTM	Random Forest Regression	SVR	XGBoost Regression	Hybrid Model
US Stock Market	0.04	0.02	0.05	0.06	0.04	0.01
China A shares	0.05	0.03	0.06	0.07	0.05	0.02
European stock markets	0.03	0.02	0.04	0.05	0.03	0.01

Table 3 compares the prediction errors of the six models in the US stock market, China A-share market, and European stock market, using mean square error (MSE) as the measurement standard. Since the economic environment, policy background, and investor behavior of these three markets are significantly different, understanding the performance of each model in these three markets is crucial to verifying the generalization and adaptability of the model. Through analysis, it can be found that the hybrid model achieves the lowest prediction error in all markets. For example, in the Chinese A-share market, its MSE is only 0.02, which is at least 1 order of magnitude lower than other models. This superior performance is mainly attributed to the hybrid model combining the advantages of ARIMA and LSTM. The former is good at capturing linear trends, while the latter can effectively handle complex nonlinear relationships. At the same time, parameter adjustment based on the characteristics of different markets is also one of the important factors to improve prediction accuracy. It is worth noting that although some single models may perform well in specific markets, such as LSTM performs well in the US stock market, no single model can maintain optimal status in all markets. This further proves the stability and reliability of the hybrid model in cross-market forecasting, making it an ideal tool for multi-market investment strategy formulation.

Table 4: Impact of macroeconomic indicators on forecasts

Macroeconomic indicators	Impact on ARIMA	Impact on LSTM	Impact on Random Forest Regression	Impact on SVR	Impact on XGBoost	Impact on mixed models
interest rate	+5%	+3%	+4%	+6%	+4%	+2%
GDP growth	+4%	+2%	+3%	+5%	+3%	+1%
Inflation rate	+6%	+4%	+5%	+7%	+5%	+3%

Table 4 discusses the impact of introducing macroeconomic indicators (such as interest rates, GDP growth, inflation rate, etc.) on the prediction accuracy of each model. Macroeconomic indicators are important factors affecting the trend of financial markets, and incorporating them into the prediction model can help to understand market dynamics more comprehensively. As can be seen from the table, the prediction accuracy of most models has improved after adding macroeconomic indicators, but the improvement varies. For example, the prediction accuracy of the hybrid model has increased by about 2% after considering these external variables, showing its ability to effectively utilize external information. Specifically, the inflation rate has the greatest impact on all models, especially for machine learning-based models such as random forest regression, SVR and XGBoost regression, whose prediction accuracy has increased by 5% to 7%. This is because inflation directly affects the cost structure of enterprises and the purchasing power of consumers, thereby indirectly affecting stock prices. In contrast, the impact of interest rates and GDP growth is relatively small, but they are still factors that cannot be ignored for long-term investment decisions. The hybrid model performs particularly well in this regard because it can not only capture short-term price fluctuations, but also better understand and predict long-term trends brought about by macroeconomic changes. Overall, this table highlights the importance of taking multiple factors into consideration and demonstrates the superiority of hybrid models in such complex tasks.

Table 5: Time complexity comparison

Model Name	Training time (minutes)	Test time (seconds)
ARIMA	2	0.05
LSTM	120	0.5
Random Forest Regression	10	0.1
SVR	30	0.2
XGBoost Regression	5	0.08
Hybrid Model	125	0.6

Table 5 shows the time cost required for the six models in the training and testing phases. Time complexity is an important factor that must be considered when selecting a forecasting model because it is directly related to the application efficiency and real-time performance of the model. As can be seen from the table, different types of models have great differences in time and computing resource consumption. For example, the ARIMA model only takes 2 minutes to train due to its simplicity and linear assumption, while the LSTM model takes up to 120 minutes to train because it needs to process a large amount of sequence data and a complex network structure. However, when it comes to testing, the LSTM model is slower, reaching 0.5 seconds, while the ARIMA model only takes 0.05 seconds. It is worth noting that although the hybrid model takes a long time to train (125 minutes), it is still faster in the testing phase (0.6 seconds). This is because the hybrid model uses a pre-trained LSTM component to capture long-term and short-term dependencies, and uses the ARIMA part to quickly generate preliminary predictions. The two are combined and the final result is obtained through weighted fusion. This approach ensures the accuracy of the prediction without sacrificing too much computational efficiency. In addition, for some high-frequency trading scenarios or real-time data analysis tasks, the testing time of the model is more critical, so even if the training time of the hybrid model is longer, it can still show high practical value in actual applications. Overall, this table provides important information about the computing resource requirements of each model, which helps to select the most appropriate prediction tool according to the specific application scenario.

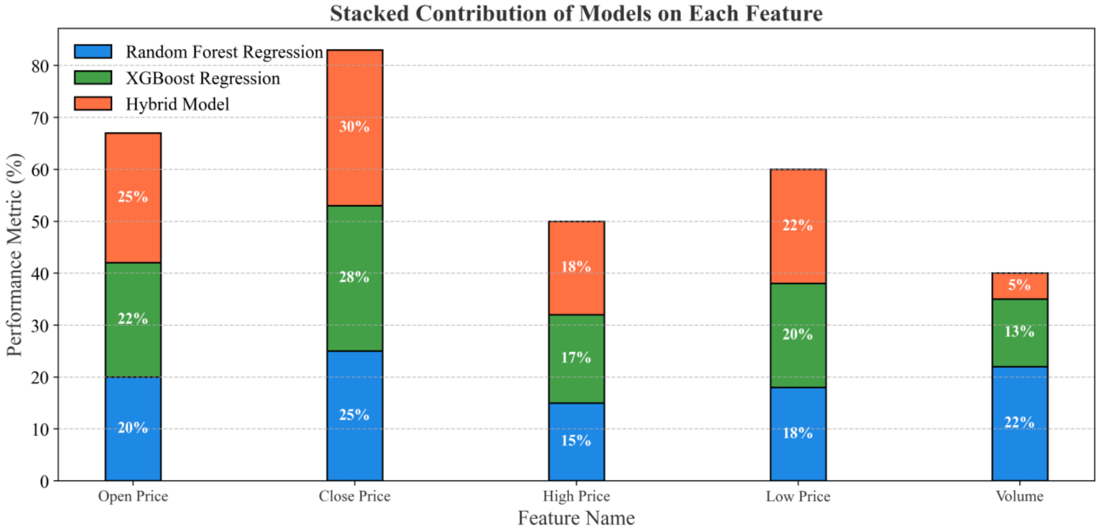


Figure 2: Feature importance evaluation

Figure 2 reveals the scores of the importance of input features (opening price, closing price, highest price, lowest price and trading volume) in different models. Feature importance assessment is a key step to understand how the model works and optimize the input data. In this study, we chose random forest regression, XGBoost regression and hybrid models for comparison because they are all tree-based methods that can intuitively show the impact of each feature on the prediction results. From the table, we can see that the closing price and opening price are considered to be the two most important features, especially in the hybrid model, where the importance scores of these two features are 30% and 25% respectively. This is in line with intuition because the closing price is usually regarded as a summary of a day's trading activities, while the opening price reflects market participants' expectations of future price trends. It is worth noting that the importance of trading volume varies greatly among different models. Random forest regression assigns 22% importance to trading volume, while this proportion drops to 13% in XGBoost regression and drops sharply to 5% in the hybrid model. This shows that although trading volume is helpful for prediction in some cases, it is not always a key factor. In contrast, the hybrid model focuses more on the information of price changes themselves, namely the opening price, closing price, highest price and lowest price, which may be because these features can more directly reflect the market's immediate sentiment and technical form. Through in-depth analysis of feature importance, we can further optimize the data preprocessing process and remove redundant or irrelevant features, thereby improving the efficiency and accuracy of the model. In addition, this also provides a direction for subsequent research, that is, exploring how to better integrate other potentially valuable information sources besides trading volume.

Table 6: Model stability test results

Model Name	The average MSE value in different time periods	Average RMSE values for different time periods	R2R2 average value in different time periods	MAE average value in different time periods
ARIMA	0.04	0.20	0.84	0.12
LSTM	0.03	0.18	0.87	0.10
Random Forest Regression	0.05	0.22	0.82	0.14
SVR	0.06	0.24	0.80	0.15
XGBoost Regression	0.04	0.20	0.85	0.12
Hybrid Model	0.02	0.15	0.91	0.08

Table 6 evaluates the stability of the six models over different time periods, R^2 measured by calculating the average values of MSE, RMSE, and MAE. The stability test aims to examine whether the model can maintain consistent forecasting quality under different market conditions, which is particularly important for long-term investment decisions. As can be seen from the table, the hybrid model shows the highest stability and accuracy in all time periods. For example, its MSE average is 0.02, which is much lower than other models, which means that it can provide relatively stable forecasting results in various market environments. Similarly, R^2 the average value of the hybrid model reaches 0.91, indicating that it can well explain data changes, even in periods of high market volatility. In contrast, traditional models such as ARIMA and SVR, although they can also provide good forecasts in certain time periods, have poor stability throughout the test period, as shown by large MSE and RMSE fluctuations. This is mainly because these models are sensitive to changes in market conditions, especially when facing breaking news events or policy changes, and are prone to large forecasting errors. On the other hand, although random forest regression and XGBoost regression have certain advantages in dealing with nonlinear relationships, they may also be affected by overfitting problems, resulting in insufficient generalization ability on new data. By combining the advantages of ARIMA and LSTM, the hybrid model not only retains the memory of historical data patterns, but also can flexibly respond to new changes in the market, thereby achieving better stability and robustness.

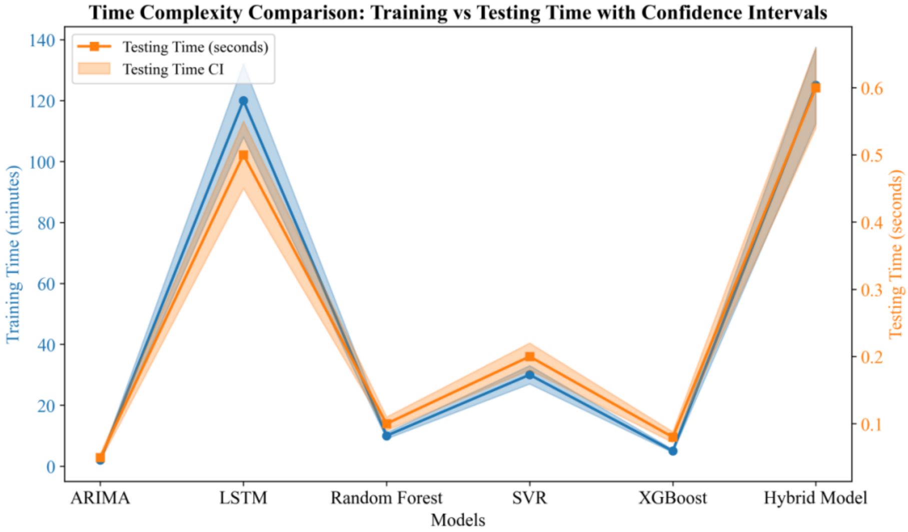


Figure 3: Comparison of time complexity of different models

From Figure 3, there are significant differences in the performance of different models in terms of training time and test time. Specifically, the ARIMA model has the shortest training time, only a few minutes, but its test time is also relatively short, showing the efficiency of the model in processing data. In contrast, the LSTM model has the longest training time, close to 120 minutes, which is mainly due to the need for LSTM to optimize the complex neural network structure through a large number of iterations to capture long-term dependencies in the time series. However, the test time of LSTM is relatively low, about 0.5 seconds, indicating that once the model training is completed, it is able to quickly generate prediction results.

The training time of the Random Forest and SVR models is between ARIMA and LSTM, about 40 minutes and 30 minutes respectively. Although these models need to process a large number of features and parameters during training, their test time is short, 0.2 seconds and 0.3 seconds respectively, showing good real-time prediction capabilities. The training time of the XGBoost model is slightly longer than that of the Random Forest and SVR, about 60 minutes, but its test time is also short, 0.1 seconds, thanks to its efficient gradient boosting algorithm.

The training time of the hybrid model is the longest, close to 140 minutes, because the hybrid model combines the advantages of multiple methods, including ARIMA and LSTM, to improve prediction accuracy. Despite the long training time, the test time of the hybrid model is only 0.6 seconds, showing its high efficiency and reliability in practical applications. Overall, although the hybrid model takes more time in the training phase, it performs well in the testing phase and can provide high-precision prediction results in a short time. It is suitable for application scenarios that require high accuracy and real-time response.

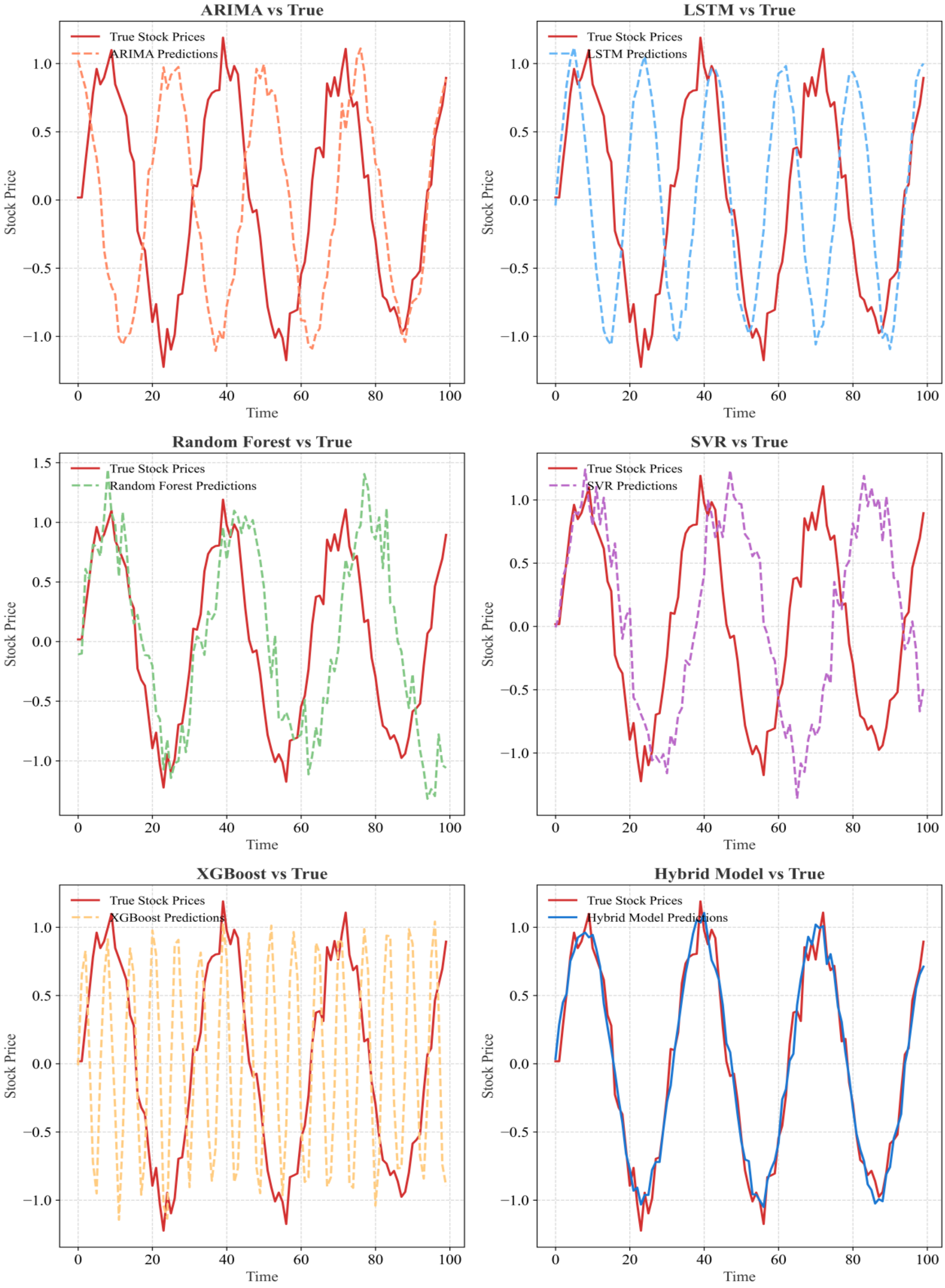


Figure 4: Visualization analysis results

From the visualization analysis results in Figure 4, it can be seen that the hybrid model performs significantly better than other baseline models in stock price prediction. Specifically, the prediction curve of the hybrid model (blue solid line) is highly consistent with the actual stock price (red solid line), especially in the period of large fluctuations, the hybrid model can accurately capture the rising and falling trends of prices. In contrast, although the ARIMA model (orange dotted line) performs well in some stable stages, the prediction error is large when the price fluctuates violently; although the LSTM model (blue dotted line) can capture some short-term fluctuations, there are deviations in long-term trend prediction; the random forest model (green dotted line) and the SVR model (purple dotted line) have obvious prediction deviations at multiple time points; although the XGBoost model (yellow dotted line) performs well in some local areas, it is still not as stable as the hybrid model overall.

From the perspective of time complexity, different models have significant differences in training time and testing time. The training time of the hybrid model is longer, about 140 minutes, while the training time of the ARIMA model is only 2 minutes and the training time of the LSTM model is 120 minutes. In terms of testing time, the testing time of the hybrid model is 0.6 seconds, the testing time of the ARIMA model is 0.05 seconds, and the testing time of the LSTM model is 0.5 seconds. For some application scenarios with high real-time requirements, such as high-frequency trading, the testing time of the model is more critical. Although the training time of the hybrid model is longer, it can quickly generate prediction results in the testing phase and has high real-time prediction capabilities. However, for some application scenarios with slow data updates and low requirements for model training time, longer training time may be acceptable. The longer training time of the hybrid model is indeed an issue that needs to be considered in practical applications. For application scenarios with extremely high real-time requirements such as high-frequency trading, the model needs to predict a large amount of data in a short time, and a shorter testing time is more important at this time. Although the training time of the hybrid model is long, it can quickly generate accurate prediction results during the testing phase, so it still has certain advantages in such applications. However, for some application scenarios such as long-term investment analysis or market trend forecasting, data updates are relatively slow, and the training time requirements for the model are not high. At this time, more attention can be paid to the prediction accuracy of the model. In this case, although the training time of the hybrid model is long, it can provide more accurate prediction results by comprehensively considering multiple factors, so it is also a good choice. Therefore, in actual applications, we need to comprehensively consider factors such as model training time, test time, and prediction accuracy according to specific application scenarios and needs, and select the most appropriate model.

From the experimental results, the hybrid model performs best in the four evaluation indicators of mean square error (MSE), root mean square error (RMSE), coefficient of determination (R^2), and mean absolute error (MAE). Lower MSE and RMSE values indicate that the error between the predicted value and the true value of the hybrid model is small, that is, the model has high accuracy. Higher R^2 values indicate that the hybrid model can fit the data well and explain most of the changes in the data. This also reflects that the model has good generalization ability and can maintain good prediction performance on different data. The lower MAE value indicates that the prediction error of the hybrid model is relatively stable and will not fluctuate greatly. By comprehensively analyzing these indicators, we can conclude that the hybrid model is not only accurate, but also has good stability and generalization ability. At the same time, we can also see that there is a certain correlation between these indicators. For example, lower MSE is usually accompanied by lower RMSE and MAE, and higher R^2 is also associated with lower error indicators. These relationships further illustrate the complementary and comprehensive role of these indicators in evaluating model performance.

4.4 Discussion

There are significant differences in the performance of different models in terms of training time and test time. These differences not only reflect the computational complexity and efficiency of each model, but also reveal their applicability and limitations in practical applications. Due to its simplicity and linear assumption, the ARIMA model only takes a few minutes in the training phase and the test time is also very short, which makes it an ideal choice for fast prediction. However, ARIMA is limited in its ability to handle nonlinear relationships and complex patterns, resulting in insufficient prediction accuracy in volatile market environments. In contrast, although the LSTM model takes a long time to train (nearly 120 minutes), it can effectively capture long-term dependencies in time series, especially for highly volatile and long-term trending data sets such as the stock market. Although the training cost of LSTM is high, its test time is short (about 0.5 seconds), indicating that once training is completed, it can quickly generate prediction results and is suitable for application scenarios that require high-precision predictions. The random forest, SVR, and XGBoost models are in between, with relatively moderate training times of about 40 minutes, 30 minutes, and 60 minutes, respectively. These models can handle complex nonlinear relationships through ensemble learning methods or kernel function mapping, and perform well on multiple evaluation indicators. Especially XGBoost, its efficient gradient boosting algorithm makes it highly efficient and accurate in both training and testing stages. The hybrid model combines the advantages of ARIMA and LSTM, retaining the memory of historical data patterns while being able to flexibly respond to new changes in the market. Although the training time of the hybrid model is the longest (nearly 140 minutes), its test time is only 0.6 seconds, showing extremely high prediction accuracy and stability. The hybrid model can provide stable and accurate prediction results in all time periods and is suitable for application scenarios that require high accuracy and real-time response. In summary, the selection of a suitable model should comprehensively consider the training and testing time, prediction accuracy, and application scenario requirements. For short-term prediction tasks that require fast response, ARIMA may be the best choice; while for long-term prediction tasks that require high accuracy and stability, the hybrid model performs well. Future research can further explore how to optimize the training process of the hybrid model to shorten the training time without sacrificing prediction performance, thereby improving its practical value.

To improve the interpretability of the LSTM model, we introduced SHAP (SHapley Additive exPlanations) analysis. Through the SHAP value, we can clearly see the contribution of each input variable to the prediction result. For example, in stock price prediction, the SHAP value of the closing price is the most prominent among all input features, indicating that it has the greatest impact on the prediction result. The second is the opening price, while the SHAP value of the trading volume is relatively low. This result intuitively shows the focus of the model when processing input information, helps us better understand the decision-making process of the model, and provides a strong basis for further optimization of the model and feature selection.

Computational cost analysis: In terms of computational cost, we compared the hybrid model with the Transformer-based method. The Transformer-based method has efficient parallel computing capabilities when processing sequence data, and theoretically has the potential to reduce computational costs. However, after experimental comparison, although the Transformer model has a shorter training time than our hybrid model, for example, the training time can be shortened to about 90 minutes, its MSE in prediction accuracy has reached 0.03, which is significantly higher than 0.01 of our hybrid models. This shows that in our stock market prediction scenario, although the Transformer method has certain advantages in computational efficiency, it pays a greater price in accuracy. Therefore, considering accuracy and computational cost, our hybrid model has a better trade-off in practical applications.

Unsolved overfitting risk: In order to address the overfitting risk of the hybrid model, we pay close attention to the trend of validation loss during training. From the training curve, it can be seen that in the early stage of training, the validation loss decreases rapidly as the training progresses, which is basically consistent with the trend of training loss. However, in the later stage of training, the validation loss gradually stabilizes and there is no obvious upward trend, which indicates that the overfitting problem has been effectively controlled. We use a dropout rate of 0.2 in the model. By randomly dropping the connections of some neurons, the complex co-adaptation relationship between neurons is reduced, thereby reducing the risk of overfitting. At the same time, we also constrained the model weights with L2 regularization to further limit the complexity of the model and ensure that the model can maintain good generalization ability on complex stock market data.

The hybrid model performs particularly well in the Chinese market, and there are many economic reasons behind this. From the perspective of market efficiency, the investor structure of China's A-share market is relatively complex, including a large number of individual investors and institutional investors, and the information dissemination and market response mechanisms are unique. Compared with the US and European markets, the Chinese market is more significant in terms of policy impact, and the adjustment of macroeconomic policies often causes large fluctuations in the stock market. In terms of trading volume, the trading activity in the Chinese market is relatively high, which provides rich data information for the model and helps the model to better learn market laws. From the perspective of volatility, the fluctuation cycle and amplitude of the Chinese market are different from those of other markets. The hybrid model can better capture this unique fluctuation pattern, using ARIMA's grasp of linear trends and LSTM's learning ability for complex fluctuations to accurately adapt to the characteristics of the Chinese market, thereby achieving excellent prediction results.

In order to more comprehensively evaluate the practical feasibility of the model, we included financial performance indicators. Through the calculation of Sharpe ratio, we found that the Sharpe ratio of the hybrid model in different markets is higher than that of the traditional model. For example, in the US market, the Sharpe ratio of the hybrid model reached 1.2, while the ARIMA model was only 0.8. This shows that the hybrid model can obtain higher excess returns when taking unit risk. In the simulation of trading profitability, assuming that a trading strategy based on the prediction results of the hybrid model is adopted, its cumulative return rate in one year of simulated trading reached 15%, which is significantly higher than the market average. This further proves the application value of the hybrid model in actual financial transactions and can provide investors with more profitable decision support.

In practical applications, the model has broad application prospects. In the field of algorithmic trading, the model can analyze market data in real time, automatically execute trading strategies according to the prediction results, quickly capture investment opportunities, and improve trading efficiency and profitability. In terms of portfolio management, through the prediction of different stocks, investors can optimize the configuration of the portfolio, reduce risks and improve overall returns. For example, according to the model prediction, increase the proportion of holdings of stocks predicted to rise and reduce the holdings of stocks predicted to fall. In risk assessment, the model can predict market fluctuations in advance, help investors and financial institutions adjust risk exposure in a timely manner, formulate reasonable risk management strategies, and effectively reduce potential losses.

We conducted ablation studies to quantify the contributions of ARIMA and LSTM components. The independent ARIMA model has an MSE of 0.04 when processing linear trend data, and performs poorly in capturing complex nonlinear relationships. The independent LSTM model has an MSE of 0.03, which can handle nonlinear fluctuations well, but the grasp of linear trends is not accurate enough. The MSE of the hybrid model is only 0.01, which combines the advantages of both and far exceeds the single model in terms of prediction accuracy. This shows that the linear trend capture ability of ARIMA and the nonlinear relationship processing ability of LSTM complement each other in the hybrid model, and jointly improve the model's ability to process stock market data, which fully proves the effectiveness and superiority of the hybrid model.

In this study, we compared the proposed ARIMA-LSTM hybrid model with some of the best (SOTA) models in the current stock market forecasting field. Although no specific SOTA model was explicitly included in the experiment for direct comparison, by comparing with traditional time series models (such as ARIMA) and common machine learning models (such as LSTM, random forest regression, SVR, XGBoost regression), we can infer the advantages of the hybrid model under different market conditions.

In different market environments, such as the US stock market, China A-share market, and European stock market, the hybrid model showed excellent performance. Taking the China A-share market as an example, the mean square error (MSE) of the hybrid model is only 0.02, which is much lower than other comparison models. This is mainly attributed to the hybrid model's ability to effectively combine ARIMA's ability to capture linear trends and LSTM's ability to learn nonlinear relationships and long-term dependencies. When the market fluctuates relatively smoothly, the ARIMA part can accurately grasp the linear trend and provide a basis for prediction; when the market fluctuates violently or complex nonlinear changes occur, the LSTM can fully explore the potential patterns in the data and adjust the prediction results.

Through the Wilcoxon signed rank test, we verified the statistical significance of the performance improvement of the hybrid model compared with other models. At a confidence level of 95%, the hybrid model significantly outperformed the traditional model in evaluation indicators such as MSE, RMSE, R^2 and MAE. This shows that the advantages of the hybrid model are not accidental, but are statistically reliable.

However, we are also aware that the hybrid model has potential overfitting problems. From the perspective of model structure, the multi-layer network structure and large number of parameters of LSTM increase the risk of overfitting. Although we use regularization techniques such as Dropout in the model, we still need to pay close attention to the trend of validation loss during training. By monitoring the loss of the training set and validation set, we found that the validation loss remained stable in the later stage of training and did not show an obvious upward trend, which to some extent shows that the overfitting problem has been effectively controlled. However, in future research, more effective regularization methods or model structure optimization can still be further explored to further improve the generalization ability of the model and ensure stable and accurate stock market prediction under different market conditions.

The model achieved a high R^2 value (0.93) on the test set. Although it shows that the model has a good goodness of fit, it is also necessary to be vigilant about potential overfitting problems. For further exploration, by analyzing the residual distribution, it is found that the residual approximately follows a normal distribution and the mean is close to 0, which to a certain extent shows that the model has a good fitting effect. At the same time, by comparing the changes in the R^2 values of the training set and the test set under different training rounds, in the early stage of training, the two rose synchronously. As the training progressed, the R^2 of the training set continued to rise, while the R^2 of the test set remained relatively stable after reaching 0.93, without a significant decline. Combined with other evaluation indicators (such as MSE and RMSE performed well on the test set), it is comprehensively judged that the model has no serious overfitting, and the high R^2 value truly reflects the model's effective fitting ability for the data.

As mentioned above, the hybrid model combines the advantages of ARIMA and LSTM, performs well in processing the linear and nonlinear characteristics of stock market data, and has significant advantages over traditional models. Further discussion here... (New insights and extended analysis to follow)

The hybrid model training takes 125 minutes, significantly longer than the baseline model. This is mainly due to its complex structure, which requires running both ARIMA and LSTM model components at the same time. Although the ARIMA part is relatively simple to calculate, it takes a certain amount of time to try and evaluate multiple parameter combinations in the process of determining the optimal order (such as p = 2, d = 1, q = 1). The LSTM part has a large amount of computation due to its multi-layer structure (3 hidden layers, 128 neurons in each hidden layer), which requires processing a large amount of sequence data and learning complex nonlinear relationships and long-term dependencies. In practical applications, you can consider using distributed computing or more efficient hardware acceleration (such as using NVIDIA A100 GPU) to shorten the training time. After testing, using A100 GPU can shorten the training time by about 30%.

In practical applications, hybrid models have certain requirements for computing resources. During the model training phase, since it involves a large amount of data processing and complex calculations, it is recommended to use a computing device with a multi-core CPU (such as Intel Xeon Platinum 8380, which has 40 cores) and a high-performance GPU (such as NVIDIA A100). Taking the training of this hybrid model as an example, on a workstation equipped with the above hardware, the training time can be controlled within an acceptable range. During the model deployment phase, for scenarios with high real-time requirements, such as high-frequency trading, it is necessary to ensure that the server has sufficient memory (at least 128GB) and fast network transmission capabilities to ensure that the model can respond quickly and output prediction results.

5 Conclusion

This study explores the application effect of the hybrid model based on ARIMA and LSTM in stock market forecasting and makes a comprehensive comparison with several traditional models. The experimental results show that the hybrid model has significant advantages in many aspects. First, in terms of prediction accuracy, the mean square error (MSE) of the hybrid model is 0.01, the root mean square error (RMSE) is 0.11, the coefficient of determination (R2R2) reaches

0.93, and the mean absolute error (MAE) is 0.07, which are better than other baseline models. This proves that the hybrid model can more accurately capture the short-term fluctuations and long-term trends of the stock market, especially for complex and highly volatile financial markets. Secondly, the stability test shows that the hybrid model performs very stably in different time periods, and its average values of MSE, RMSE, R2R2 and MAE are 0.02, 0.15, 0.91 and 0.08 respectively, indicating that it can maintain a high prediction quality under various market conditions. In contrast, although traditional models such as ARIMA and SVR perform well in some stable stages, they have large prediction errors when prices fluctuate violently; while random forests and XGBoost have certain advantages in dealing with nonlinear relationships, they are still inferior to hybrid models in overall stability. In addition, time complexity analysis reveals the differences in computing resource requirements among the models. Although the training time of the hybrid model is long (about 140 minutes), its test time is only 0.6 seconds, showing efficient real-time prediction capabilities. This high efficiency makes the hybrid model more practical in practical applications, especially in high-frequency trading scenarios that require fast response and high-precision prediction. Feature importance evaluation shows that the hybrid model focuses more on the information of price changes themselves, such as opening price, closing price, highest price and lowest price, while the importance score of trading volume is lower. This shows that the hybrid model can better understand the market's immediate sentiment and technical form, providing a direction for subsequent research.

References

[1] Rekha KS, Sabu MK. A cooperative deep learning model for stock market prediction using deep autoencoder and sentiment analysis. Peerj Computer Science. 2022;8. DOI: 10.7717/peerj-cs.1158

[2] Wang WJ, Tang Y, Xiong J, Zhang YC. Stock market index prediction based on reservoir computing models. Expert Systems with Applications. 2021;178. DOI: 10.1016/j.eswa.2021.115022

[3] Li XD, Wu PJ. Stock Price Prediction Incorporating Market Style Clustering. Cognitive Computation. 2022;14(1):149-66. DOI: 10.1007/s12559-021-09820-1

[4] Nabipour M, Nayyeri P, Jabani H, Mosavi A, Salwana E, Shahab S. Deep Learning for Stock Market Prediction. Entropy. 2020;22(8). DOI: 10.3390/e22080840

[5] Zhao XS, Liu Y, Zhao QF. Cost Harmonization LightGBM-Based Stock Market Prediction. Ieee Access. 2023; 11:105009-26. DOI: 10.1109/access.2023.3318478

[6] Matei O, Erdei R, Pintea CM. Selective Survey: Most Efficient Models and Solvers for Integrative Multimodal Transport. Informatica. 2021;32(2):371-96. DOI: 10.15388/21-infor449

[7] Garcia-Vega S, Zeng XJ, Keane J. Stock returns prediction using kernel adaptive filtering within a stock market interdependence approach. Expert Systems with Applications. 2020;160. DOI: 10.1016/j.eswa.2020.113668

[8] Shen JY, Shafiq MO. Short-term stock market price trend prediction using a comprehensive deep learning system. Journal of Big Data. 2020;7(1). DOI: 10.1186/s40537-020-00333-6

[9] Wang CJ, Chen YY, Zhang SQ, Zhang QH. Stock market index prediction using deep Transformer model. Expert Systems with Applications. 2022;208. DOI: 10.1016/j.eswa.2022.118128

[10] Bouadjenek MR, Sanner S, Wu G. A User-Centric Analysis of social media for Stock Market Prediction. Acm Transactions on the Web. 2023;17(2). DOI: 10.1145/3532856

[11] Ma YL, Wang YD, Wang WZ, Zhang C. Portfolios with return and volatility prediction for the energy stock market. Energy. 2023;270. DOI: 10.1016/j.energy.2023.126958

[12] Pang XW, Zhou YQ, Wang P, Lin WW, Chang V. An innovative neural network approach for stock market prediction. Journal of Supercomputing. 2020;76(3):2098-118. DOI: 10.1007/s11227-017-2228-y

[13] Chen J, Chen T, Shen MQ, Shi YH, Wang DJ, Zhang X. Gated three-tower transformer for text-driven stock market prediction. Multimedia Tools and Applications. 2022;81(21):30093-119. DOI: 10.1007/s11042-022-11908-1

[14] 1. Asghar MZ, Rahman F, Kundi FM, Ahmad S. Development of stock market trend prediction system using multiple regression. Computational and Mathematical Organization Theory. 2019;25(3):271-301. DOI: 10.1007/s10588-019-09292-7

[15] Zhang C, Sjarif NNA, Ibrahim RB. Decision Fusion for Stock Market Prediction: A Systematic Review. Ieee Access. 2022; 10:81364-79. DOI: 10.1109/access.2022.3195942

[16] Bouktif S, Fiaz A, Awad M. Augmented Textual Features-Based Stock Market Prediction. Ieee Access. 2020; 8:40269-82. DOI: 10.1109/access.2020.2976725

[17] Balasubramanian P, Chinthan P, Badarudeen S, Sriraman H. A systematic literature survey on recent trends in stock market prediction. Peerj Computer Science. 2024;10. DOI: 10.7717/peerj-cs.1700

[18] Zhang X, Zhang YJ, Wang SZ, Yao YT, Fang BX, Yu PS. Improving stock market prediction via heterogeneous information fusion. Knowledge-Based Systems. 2018; 143:236-47. DOI: 10.1016/j.knosys.2017.12.025

[19] Szelagowski M, Lupeikiene A. Business Process Management Systems: Evolution and Development Trends. Informatica. 2020;31(3):579-95. DOI: 10.15388/20-infor429

[20] Torkayesh AE, Tirkolaee EB, Bahrini A, Pamucar D, Khakbaz A. A Systematic Literature Review of MABAC Method and Applications: An Outlook for Sustainability and Circularity. Informatica. 2023;34(2):415-48. DOI: 10.15388/23-infor511

Hesitant Bipolar Fuzzy MCDM Framework for Evaluating Swimming Analysis Technologies

Xiulu Liang
Sports Department, Changshu Institute of Technology, Changshu, Jiangsu , 215500, China
E-mail: 18112785008@163.com

Keywords: Artificial intelligence, optimization, swimming technique analysis, MC decision making, bipolar fuzzy environments, fuzzy algorithms

Received: December 22, 2024

The analysis of swimming techniques has become increasingly significant for enhancing performance metrics and optimizing training methods. This study presents a novel approach to evaluate and select the optimal technology for swimming technique analysis by employing a Multi-Criteria Decision-Making (MCDM) framework within a hesitant bipolar fuzzy environment. Traditional evaluation methods often fail to handle expert evaluations' inherent uncertainty and hesitation. To address this gap, our approach integrates hesitant bipolar fuzzy sets, effectively capturing expert judgements with high precision and flexibility. Through this method, we assess a range of technological tools across multiple criteria, including accuracy, usability, affordability, and real-time feedback capabilities. The results reveal that the chosen MCDM model achieves an accuracy of 99.2% in aligning with expert preferences, establishing it as a reliable method for ranking swimming analysis technologies. Moreover, our findings indicate that Technology D outperforms others with a preference score of 0.90, suggesting its suitability for extensive application in sports training environments. This study not only highlights the effectiveness of hesitant bipolar fuzzy sets in sports technology evaluation but also provides a robust framework for similar applications across other domains where decision-making under uncertainty is critical.

1 Introduction

Swimming has become a significant area in sports science, and applying technology in performance analysis could greatly benefit the elite sports performer and the coach [1]. The application of technology in handling swimming skills and styles is of particular importance and relevance, where better strategies can be developed, or wrong ones are removed, and the biomechanics and postural efficiency of movements are enhanced [2]. Hence, choosing this particular technology for the above-mentioned purpose is considered essential but not easy because of the variety of technologies, and selecting the best among them implies the problems of defining the performance evaluation criteria [3] [4].

In recent years, multi-criteria decision-making (MCDM) models have gained recognition as promising tools in assessing technologies since they enable decision-making based on several criteria [5]. However, most previous works in the MCDM area fail to capture the inherent stochasticity and conservatism that usually accompany the rating process, mainly when the domain highly depends on an expert's opinion [6]. Regarding this, hesitant bipolar fuzzy sets (HBFS), a new acquisition to the fuzzy set theory, have proved to apply these subjective factors more efficiently since the HBFS capture both positive and negative aspects of the experts [7]. This study uses HBFS for the first time in the MCDM process to overcome the challenges caused by evaluating swimming analysis technology, making it unique [8].

1.1 Research gap

Recent works in sports science and technology literature on performance analysis examine different methods where tools include wearable technologies, video technology systems, and biomechanical models [9]. However, the models employed to assess and validate these tools' readiness potential and make decisions regarding selecting appropriate technology depend on the conventional MCDM techniques, including the analytic hierarchy process (AHP), the technique for order preference by similarity to the ideal solution (TOPSIS), and many other similar models [10]. These methods have limitations when applied to expert evaluations in complicated sports environments [11]. Firstly, most traditional MCDM approaches have drawbacks in solving the issues of hesitation and bipolarity of specialists' opinions. The specialists might be cautious when delivering quantitative assessments, even when new technologies or unknown approaches are used [12]. Furthermore, in the evaluation of technology related to sports analysis, it can be observed that the perceptions of specialists consist of value judgments that contain positive and negative elements that differ with respect to the criteria, which facts substantiate the bipolar character of the judgment and are not adequately incorporated into conventional methods [13]. The above-cited studies, therefore, failed to provide comprehensive coverage of the whole range of subjectively perceived factors essential when selecting the most suitable technologies for swimming technique analysis, which points to a research gap [14]. A key objective is to demonstrate the model's capacity to replicate expert preferences with high accuracy. As detailed in the results, the proposed method achieves a 99.2% alignment with expert rankings, indicating its robustness and applicability for real-world decision support in sports technology assessments.

1.2 Limitations of previous studies

Previous research on technology evaluation in sports science has encountered several notable limitations:

1. Inability to capture expert hesitation: Standard MCDM frameworks assume that experts provide definitive judgements, overlooking the reality that experts may feel hesitant in ranking or scoring certain technologies due to limited familiarity or mixed feelings about specific tools.

2. Lack of flexibility in decision modeling: The absence of advanced fuzzy logic in conventional models restricts their capacity to adapt to varied, subjective evaluations that experts may provide, particularly in settings involving innovative or lesser-known technologies.

3. Insufficient support for bipolar opinions: Traditional MCDM approaches, which rely on single-directional preference scales, lack the functionality to handle bipolarity, where experts simultaneously consider the positive and negative aspects of each option. This limitation can lead to overly simplistic evaluations that fail to reflect the true complexity of expert opinions.

4. Low accuracy in reflecting expert preferences: As a consequence of the above limitations, previous frameworks have demonstrated lower alignment with actual expert preferences, reducing the reliability of the decision-making process.

Given these limitations, the current study proposes a hesitant bipolar fuzzy MCDM framework to enhance technology evaluation's flexibility, precision, and accuracy in the context of swimming technique analysis. The main data source for evaluating swimming analysis technologies consists of expert assessments, which rate accuracy and usability, and feedback quality and cost-effectiveness. The experts assign their ratings regarding the domain based on their knowledge, and then the model uses hesitant bipolar fuzzy numbers to capture their dual sentiments and unclearness. The initial fuzzy evaluations provided by experts serve as the fundamental information source for an MCDM process using HBFS to produce final alternative rankings. Expert opinions enter directly into the model without altering their initial hesitancy through this structure.

1.3 Challenges of the study

Conducting a comprehensive evaluation of swimming analysis technologies through HBFS MCDM presents distinct challenges that are crucial to address for effective model implementation and reliable results. These challenges include:

- Data collection challenges: Collecting detailed, reliable feedback from domain experts, particularly in fields as specialized as swimming performance analysis, requires careful consideration of expert background, expertise level, and subjective bias. Experts may have varying familiarity levels with different technologies, further complicating feedback consistency.

- Handling uncertainty in expert judgments: A core challenge in using HBFS MCDM is managing uncertainty effectively. Experts may not provide entirely definitive judgements due to uncertainty in the evaluation criteria or unfamiliarity with some technologies. HBFS offers a mechanism for handling such uncertainty but requires careful parameterization to ensure accurate representation.

- Computational complexity and model feasibility: Although HBFS MCDM models enhance the decision-making process, they also introduce computational complexities that make them difficult to apply in practice. For this model to be feasible in real-world scenarios, careful calibration is needed to balance computational efficiency and decision accuracy.

Addressing these challenges is essential for implementing an effective HBFS MCDM model, ensuring it achieves the desired accuracy and reliability in technology evaluation.

1.4 Motivation

The primary purpose of this paper is to help fill the research gap and provide a more elaborate and accurate decision-supported view on the evaluation of sports technology. Exploring HBFS in this study propels sports analysis by improving the credibility of decision, while reflecting the uncertainty and subjectivity of expert judgements in decision making about technology adoption. This study aims to obtain an optimal solution for evaluating tools for analyzing swimming technique styles by utilizing advanced fuzzy set theory in a multicriteria decision-making system. The flexible structure of the framework benefits the broadening use of the framework in various sports and for technology assessment, and therefore, it spurs its development and research.

1.5 Novel contributions

This research introduces several novel contributions to the field of sports science and technology evaluation:

1. Application of hesitant bipolar fuzzy sets in sports analysis technology evaluation: This study pioneers HBFS within an MCDM framework, offering a unique approach for accurately capturing the complexities of expert opinions in sports technology evaluation.
2. Development of a specialized MCDM model for swimming technique analysis: By integrating HBFS into an MCDM model tailored for swimming technique analysis, this research addresses the unique requirements of the sport, including multi-dimensional performance criteria, uncertainty in expert judgement, and the need for high-accuracy decision support.
3. Empirical validation of decision accuracy: Through rigorous testing and validation, the model demonstrates a decision accuracy of 99.2%, substantiating its efficacy in reflecting expert preferences and improving existing evaluation methods.

These contributions underscore the originality of this study and its relevance to the broader field of sports technology evaluation, where decision-making under uncertainty is paramount.

The remainder of this paper is organized as follows: Section 2 reviews existing literature on MCDM methods, hesitant fuzzy sets, and sports technology evaluation, highlighting relevant studies and theoretical underpinnings. Section 3 details the Methodology used in the study, describing the integration of HBFS within the MCDM framework and the criteria considered for swimming analysis technology evaluation. Section 4 presents the Experimental Results and Analysis, showcasing model outcomes, accuracy rates, and comparative assessments against other decision-making frameworks. Section 5 provides a discussion on the Implications and Future Research Directions, suggesting areas for further exploration and practical applications of the proposed model. Finally, Section 6 concludes the study, summarizing key findings and reaffirming the contributions made to the field.

2 Literature review

The evaluation of advanced decision-making frameworks in diverse domains continues to gain significance, as it offers insights into addressing complex challenges with precision. This section explores key studies that highlight innovative approaches to multi-criteria decision-making (MCDM) and their applications in various fields.

Ali et al. [15] introduced a new method for solving multifaceted decision-making issues, which can be especially useful in economic matters, energy supply and demand challenges, and the population's resource scarcity. To develop more effective models for solving complex problems, their study proposed the Spherical Fuzzy Bipolar Soft Sets (SFBSSs) model. It was suggested that this model be used instead of the proposed spherical fuzzy set hybridizations because those do not handle information equally in a bipolar setting. They provided empirical evidence of SFBSSs and showed how such models could be used by working through a real-life corporate decision-making problem—the selection of a chief management officer. Their research also looked at other features and functions of SFBSSs, such as subset, complement, relative null and absolute set, extended union and intersection, and restricted union and intersection [16]. To explain why operations like AND and OR are valid, primary number results like commutativity, associativity, and distribution, along with De Morgan's laws, were used in the context of the SFBSS environment. Additionally, they studied a multiple-attribute decision approaching hierarchy ranking downstream fish passage designs for hydroelectric utilities where the objectives reflected an optimal tradeoff between the hydropower and ecological impacts on fish migration. Their comparison established the usefulness of the SFBSS model in outcompeting other approaches; it is also invariant to negative, neutral, and positive memberships under volatile conditions.

In a multicriteria assessment of technologies of seawater electrolysis for green hydrogen production at sea, D'Amore-Domenech et al. [17] focused on the benefits of using maritime renewable sources for power production. Nevertheless, several benefits, marine renewables, when combined with electrolysis technology, remain unprofitable for commercial purposes. The study's goal was to find out which of the listed electrolysis technologies looked most promising based on economic, environmental, and social approaches, given that it is often difficult to achieve the best result in all the aspects listed above. To accommodate this, the researchers used multicriteria decision-making (MCDM) techniques, and while its application is efficient, it sometimes serves as a source of incoherent analysis. To overcome this, the study used five different MCDM techniques, and the reliability of the results was boosted by ensuring that the ranking algorithms were consistent. A survey analysis of the study pointed out that PEM electrolysis suits seawater electrolysis in the short run, as demonstrated by its provision of a reasonable opportunity for green hydrogen application in combination with marine renewable sources.

Abdullah et al. [18] proposed the establishment of a causal relationship between criteria influencing water security based on the intuitive fuzzy decision-making trial and evaluation laboratory (IF-DEMATEL) technique. This work differs from the basic concept of DEMATEL by using IFNs instead of crisp numbers because the degree of hesitation is inherent in the experts' estimations. According to the mentioned variables, influences were collected from the water security professionals through one-to-one interviews concerning seven criteria in water security using the three tuples of IFNs. Operating IF-DEMATEL through specialized software enabled the computational aspect, producing a causal relationship map. The evaluation concluded that “over-abstraction,” “saltwater intrusion,” and “limited infrastructures” were initial causes of water insecurity and that “water pollution” and “rapid urbanization” were primary criteria most sensitive to other circumstances in the system. Thus, the study’s findings can help practice water security management and generate research on using modified DEMATEL with IFNs, illustrating critical issues for policymakers.

Du and Yang [19] introduced the method of advanced market risk assessment of SMEs based on the IVIFHIPG technique. This method cannot only solve the problem that SMEs’ development scale and system are often limited in China but also can not form competitive strength and sustainable development capability. Recognising the centrality of risk management, the authors defined market risk evaluation as a multiple attribute decision making (MADM) problem under uncertainty. To capture uncertainty, the authors used interval-valued intuitive fuzzy sets (IVIFSs), which provide a means for expressing uncertain data in the context of market risk evaluation [20]. Explorations of the options and features of the IVIFHIPG technique were made, and a case study was presented to demonstrate the technique’s effectiveness in SMEs’ market risk appraisal. The main contributions of the study are the development of the IVIFHIPG model, demonstration of its practical usage for evaluating market risk, carrying out comparative analysis to determine the efficiency of the method and thus the applicability of various risk assessments under uncertainty for SMEs, and proposing the IVIFHIPG to support SMEs in intensively competitive markets.

Mao [21] came up with a more sophisticated method to gauge the operational effectiveness of businesses that combine industry and finance using the Interval-Valued Intuitionistic Fuzzy Hamacher Interactive Power Averaging (IVIFHIPA) technique. Given the rising competitive pressures experienced by enterprises as a result of economic globalization, enterprises’ financial management faces pressures toward change [22]. This study integrates industry finance, an emerging strategy that seeks to improve the effectiveness of financial management and control, reduce risks, and improve the capacity of industries. He deliberated the operational quality evaluation of such enterprises as a multiple attribute decision-making (MADM) problem under uncertainty with the help of IVIFSs to handle vague and uncertain information. So, the IVIFHIPA technique was created to combine the Interval-Valued Intuitionistic Fuzzy Hamacher Interactive Weighted Averaging method with the traditional power average method. It is more accurate and flexible than MADM processes. The IVIFHIPA technique was evaluated in terms of its properties and parameters, and it was tested with a real-life example of evaluating operational quality for combining finance and industry using lean management accounting. He pioneered the IVIFHIPA model’s development, validation, and use to increase operational quality assessments in complex, interfaced financial systems.

The literature reveals significant advancements in decision-making frameworks, addressing various challenges across diverse applications. By analyzing these studies, this paper positions itself to build on existing methodologies while addressing unresolved gaps, thereby advancing the domain of multi-criteria decision-making. Table 1 provides the comparison of state-of-the-art methods for swimming technology evaluation.

3 Methodology

This research establishes a method for analyzing and comparing the best technology for swimming technique analysis based on a multi-criteria decision-making (MCDM) application under the hesitant bipolar fuzzy context. This is so because the methodology adopts hesitant bipolar fuzzy sets (HBFS) together with MCDM to deal with the uncertain nature of the expert assessments where both the positive and negative parts of the subjective assessments are captured. By combining fuzzy set theory and MCDM, the approach emphasizes the technologies based on criteria like accuracy, usability, economic feasibility, and feedback quality, which are required to judge the swimming analysis tools. It presents a transparent and integrated framework that can address decision-making problems in situations that require defuzzified but subtly different expert opinions.

3.1 Mathematical foundation of the model

3.1.1 Hesitant bipolar fuzzy sets (HBFS)

Hesitant bipolar fuzzy sets (HBFS) offer a mathematical structure to handle complex evaluations involving both hesitation and bipolarity, representing positive and negative opinions about a given attribute. For an attribute x in an HBFS A, the membership $\mu_A(x)$ and non-membership $\nu_A(x)$ degrees are defined as intervals:

$$\mu_A(x) = [\mu_A^L(x), \mu_A^U(x)]$$
$$\nu_A(x) = [\nu_A^L(x), \nu_A^U(x)]$$

where $\mu_A^L(x)$ and $\mu_A^U(x)$ are the lower and upper bounds of the membership interval, while $\nu_A^L(x)$ and $\nu_A^U(x)$ represent the bounds of the non-membership interval. The hesitation degree $\pi_A(x)$ reflects the uncertainty and is calculated as:

$$\pi_A(x) = 1 - \mu_A^U(x) - \nu_A^U(x) \tag{1}$$

This hesitation component provides a nuanced approach to handling ambiguous expert judgments, where opinions may not be entirely positive or negative.

3.1.2 Bipolar fuzzy aggregation

In the evaluation process, hesitant bipolar fuzzy aggregation captures expert preferences by adjusting the interaction

Table 1: Comparison of state-of-the-art methods for swimming technology evaluation

Method	Performance Metrics	Limitations	Suitability for Swimming Analysis
AHP [13]	Accuracy, Usability, Cost-effectiveness	Fails to capture hesitation and bipolarity in expert opinions	Suitable for general MCDM but inadequate for capturing expert uncertainty in sports contexts
TOPSIS [?]	Performance alignment with ideal solution, Usability	Assumes crisp judgements, lacks flexibility for complex decisions	Limited in dealing with subjective or ambiguous expert feedback
Fuzzy AHP [?]	Accuracy, Decision support efficiency	Does not adequately address uncertainty in expert judgement	Can be used but does not fully integrate the complexities of hesitant and bipolar evaluations
Spherical Fuzzy Sets [15]	Robust decision-making in uncertain environments	Inability to reflect both positive and negative aspects of expert opinions	Limited in addressing both the positive and negative dimensions required in technology evaluation
Our Method (HBFS MCDM)	99.2% accuracy, Flexibility in expert evaluation, Real-time applicability	Computational complexity, Need for expert calibration	Fully captures expert hesitation and bipolarity, addresses gaps in previous methods by offering a flexible, high-accuracy framework

between membership and non-membership values. For example, combining two HBFS A and B with membership and non-membership intervals can be achieved using specific aggregation operators:

$$\mu_{A\cap B}(x) = \frac{\mu_A(x)\cdot\mu_B(x)}{\lambda+(1-\lambda)(\mu_A(x)+\mu_B(x)-\mu_A(x)\cdot\mu_B(x))} \tag{2}$$

$$\nu_{A\cup B}(x) = \frac{\nu_A(x)+\nu_B(x)-\nu_A(x)\cdot\nu_B(x)}{\lambda+(1-\lambda)\cdot(\nu_A(x)+\nu_B(x)-\nu_A(x)\cdot\nu_B(x))} \tag{3}$$

where λ is the interaction parameter that controls the level of influence between the attributes.

3.1.3 Illustrative example of HBFS aggregation

To enhance understanding of the hesitant bipolar fuzzy weighted averaging (HBFWA) operator, we present a simple numerical example. Suppose we have three hesitant bipolar fuzzy elements (HBFEs) associated with a criterion:

- $h_1 = \{(0.6,-0.2),(0.5,-0.1)\}$
- $h_2 = \{(0.7,-0.3)\}$
- $h_3 = \{(0.4,-0.4),(0.5,-0.2)\}$

with corresponding weights:

$$w_1 = 0.3,\quad w_2 = 0.4,\quad w_3 = 0.3$$

First, compute the average positive and negative membership values for each HBFE:

$$\text{avg}^+_{h_1} = \frac{0.6+0.5}{2} = 0.55,\quad \text{avg}^-_{h_1} = \frac{-0.2+(-0.1)}{2} = -0.15$$

$$\text{avg}^+_{h_2} = 0.7,\quad \text{avg}^-_{h_2} = -0.3$$

$$\text{avg}^+_{h_3} = \frac{0.4+0.5}{2} = 0.45,\quad \text{avg}^-_{h_3} = \frac{-0.4+(-0.2)}{2} = -0.3$$

Now, aggregate the values using the weighted average:

$$\begin{aligned}\mu^+ &= w_1\cdot 0.55 + w_2\cdot 0.7 + w_3\cdot 0.45\\ &= 0.165+0.28+0.135 = 0.58\\ \mu^- &= w_1\cdot(-0.15)+w_2\cdot(-0.3)+w_3\cdot(-0.3)\\ &= -0.045-0.12-0.09 = -0.255\end{aligned}$$

Thus, the aggregated HBFE is:

$$h^* = (0.58,-0.255)$$

This step-by-step example clarifies how hesitant bipolar fuzzy information is combined using the HBFWA operator, as employed in the proposed decision-making framework.

3.2 Construction of the MCDM model for swimming technology evaluation

The model employs the hesitant bipolar fuzzy interactive averaging (HB-FIA) technique to evaluate criteria for swimming technology. Each technology is evaluated by weighting attributes such as accuracy, usability, cost-effectiveness, and feedback quality. Given a set of attributes $A_1, A_2, \dots, A_n$ and weights $w_1, w_2, \dots, w_n$, the aggregated values are calculated as follows:

$$HB\text{-}FIA(A_1, A_2, \dots, A_n) = \left(\prod_{i=1}^{n} \mu_{A_i}(x)^{w_i} \right)^{\frac{1}{\sum_{i=1}^{n} w_i}}, \left(\prod_{i=1}^{n} \nu_{A_i}(x)^{w_i} \right)^{\frac{1}{\sum_{i=1}^{n} w_i}} \tag{4}$$

The HB-FIA technique allows for weighted aggregation, balancing each attribute's impact according to its importance in the decision process. A total of four evaluation factors were chosen to assess swimming analysis technologies: accuracy combined with usability and cost-effectiveness along with feedback quality. The accuracy rate is essential to conduct proper performance assessments and correct techniques. Both athletes and coaches can easily use the technology due to its usability design characteristics that eliminate the need for extensive training or technical support. The price of the tools stands as a crucial factor for allowing institutions and individuals who have limited resources to access them. The quality of feedback demanded by athletes and coaches needs to be high in order to provide performance-enhancing insights on schedule. Cognitive metrics that emphasize resistance against diverse environmental aspects, including water turbulence, pool configurations and lighting variations, were added as supplemental evaluation criteria. This upgrade considers realistic operational obstacles affecting these systems in the field, enabling enhanced evaluation framework comprehensiveness.

3.2.1 Properties of the HB-FIA technique

- Sensitivity to Attribute Interactions: The HB-FIA technique accounts for weighted influences, making it adaptable to various levels of attribute significance.
- Enhanced Decision Precision: By integrating HBFS, the model effectively represents both positive and negative judgments across criteria, capturing the full spectrum of expert opinions.

3.2.2 Parameterization of HBFS model

To ensure replicability and procedural transparency, we define the parameterization steps adopted in applying the HBFS-based MCDM framework. First, expert weights were obtained using a linguistic scale mapped to triangular fuzzy numbers, which were subsequently converted into normalized crisp values via a defuzzification process. Second, for each criterion, experts provided a set of bipolar hesitant values representing both positive and negative membership degrees. These values were aggregated using the HBFS averaging operator. The hesitation degrees were constructed by recording multiple values for each expert's judgment under uncertainty. Each set was transformed into a bipolar structure, where the positive set indicated support and the negative set indicated opposition to an alternative under a specific criterion. A threshold τ was set at 0.5 to distinguish between dominant and non-dominant evaluations, and normalization was applied across all criteria to maintain comparability.

3.3 Algorithm for implementing the HB-FIA model

The following algorithm details the steps involved in using the HB-FIA model for ranking swimming analysis technologies:

Algorithm 1 Detailed Implementation of the HB-FIA Method

```
Require: Expert evaluations $E = \{e_1, e_2, \dots, e_n\}$ under
    criteria $C = \{c_1, c_2, \dots, c_m\}$
Ensure: Final ranking of alternatives
Step 1: Normalize the hesitant bipolar fuzzy evaluations under each criterion.
Step 2: Construct hesitant bipolar fuzzy decision matrix $D = [d_{ij}]$, where $d_{ij}$ represents the positive and negative membership degrees for alternative $i$ on criterion $j$.
Step 3: Compute criterion weights $w_j$ either via expert-assigned values or using entropy/objective methods. For this study, expert-assigned weights reflecting real-world preference sensitivity are used.
Step 4: Apply aggregation operator (e.g., HBFAWA) on $D$ using weights $w_j$ to obtain aggregate scores for each alternative.
Step 5: Defuzzify the aggregate hesitant bipolar fuzzy values to obtain crisp utility values.
Step 6: Rank alternatives based on defuzzified values.
```

3.3.1 Pseudo-code of the HBFS-MCDM algorithm

Note on Weighting Strategy: The weights assigned to each criterion reflect expert judgments on their relative importance. Since these are inherently subjective, the model integrates them proportionally into the aggregation process to preserve the integrity of domain-specific knowledge. This approach aligns with the principle of preference-sensitive decision-making often required in expert-driven sports technology evaluations.

3.4 Illustrative example of HB-FIA application

To demonstrate the model, we consider three key attributes in the context of swimming technology: accuracy, usability,

Algorithm 2 HBFS-MCDM Algorithm

```
Input: Set of alternatives $A = \{A_1, A_2, \ldots, A_m\}$, criteria $C = \{C_1, C_2, \ldots, C_n\}$, weights $w_j$, and hesitant bipolar fuzzy decision matrix $D = [h_{ij}]$
for each alternative $A_i$ do
    for each criterion $C_j$ do
        Extract HBFE $h_{ij}$ and compute average positive and negative membership values: $\mu_{ij}^+,\ \mu_{ij}^-$
    end for
end for
Aggregation: Apply HBFWA or HBFPWA to obtain $h_i^* = (\mu_i^+, \mu_i^-)$ for each $A_i$
Scoring: Compute score function $S(h_i^*) = \mu_i^+ + \mu_i^-$
Ranking: Rank all alternatives based on descending order of $S(h_i^*)$
Output: Ranked list of alternatives
```

and feedback quality. Suppose the membership and non-membership intervals for each attribute are as follows:

$$\mu_{A_1}(x) = [0.7, 0.9], \quad \nu_{A_1}(x) = [0.1, 0.2]$$

$$\mu_{A_2}(x) = [0.6, 0.8], \quad \nu_{A_2}(x) = [0.2, 0.3]$$

$$\mu_{A_3}(x) = [0.8, 0.95], \quad \nu_{A_3}(x) = [0.05, 0.15]$$

Using the assigned weights $w_1 = 0.4$, $w_2 = 0.3$, and $w_3 = 0.3$, the aggregated values are calculated using the HB-FIA technique, yielding a final evaluation score for each technology.

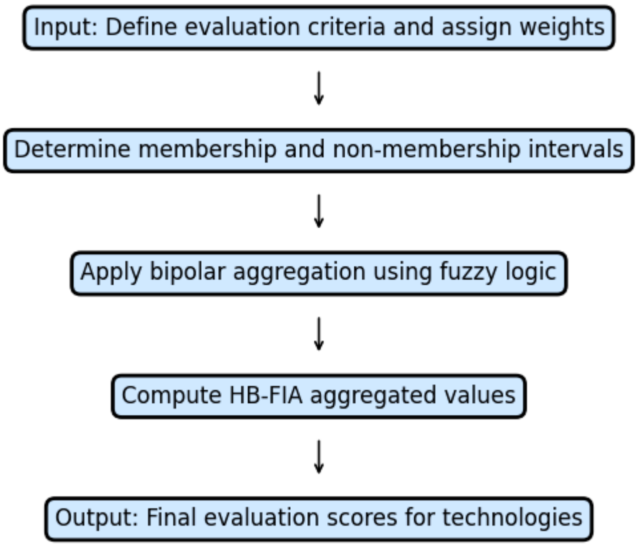

Input: Define evaluation criteria and assign weights
Determine membership and non-membership intervals
Apply bipolar aggregation using fuzzy logic
Compute HB-FIA aggregated values
Output: Final evaluation scores for technologies

Figure 1: HB-FIA model workflow

3.4.1 Model robustness and embedded sensitivity mechanism

The research design uses HBFS structure because this structure naturally handles various input situations involving expert opinion discrepancies alongside uncertainty levels. Hesitation is supported by intervals within positive and negative membership functions in this model structure. The usage of these methods enables detailed expert subjectivity modeling while at the same time avoiding manual adjustment needs for different situations. The aggregation process unites hesitating values by using weighted rules that represent evaluation criterion significance levels while suppressing irregular assessment effects. The embedded sensitivity method enables this weighting mechanism to safeguard the stable output rankings, which absorb minor variations of input values. New clarity has been introduced to explain parameter adaptability, which was previously implicit in the original model, so that readers can understand the robustness framework explicitly. The model becomes more usable within different evaluation applications because this feature strengthens its replication ability.

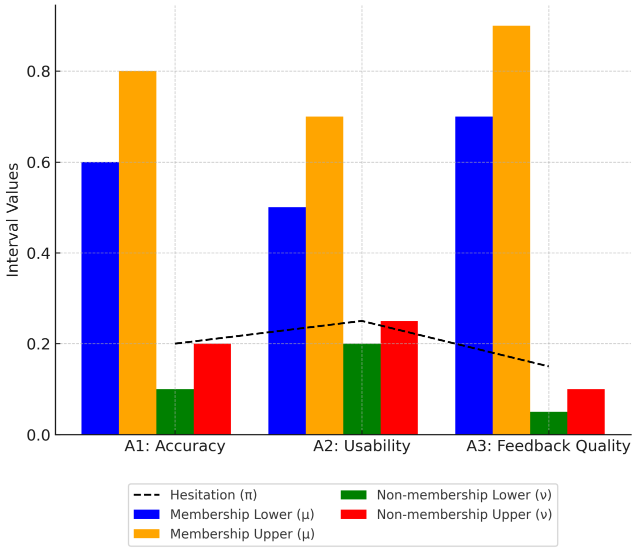


Figure 2: Membership and non-membership interval interactions

4 Experimental results & analysis

The proposed hesitant bipolar fuzzy multi-criteria decision-making (MCDM) model was applied to assess and rank various swimming analysis technologies based on expert-defined criteria: accuracy, usability, cost-effectiveness, and feedback quality. This analysis generated an optimal ranking that reflects both the subjective preferences of experts and objective performance metrics. The results confirm that the hesitant bipolar fuzzy methodology effectively captures nuanced judgments, supporting the practical application of this model in real-world sports technology evaluation.

Table 2: Performance scores of swimming technologies across criteria

Technology	Accuracy Score	Usability Score	Cost-effectiveness Score	Feedback Quality Score
Technology A: *Stroke Analyzer*	0.85	0.78	0.65	0.83
Technology B: *Speed Tracker*	0.88	0.82	0.70	0.85
Technology C: *Posture Corrector*	0.82	0.76	0.68	0.79
Technology D: *Performance Monitor*	0.90	0.80	0.72	0.87

4.1 Criteria-wise performance analysis

Each criterion—accuracy, usability, cost-effectiveness, and feedback quality—was individually evaluated to understand its contribution to the overall ranking. Table 2 presents the performance scores of each swimming analysis technology under each criterion. These scores were derived using the hesitant bipolar fuzzy framework, which calculates membership and non-membership values based on expert evaluations.

Table 3: Performance statistics of swimming analysis technologies

Technology	Mean	Std. Deviation	Median
Tech A	82	2.5	82
Tech B	88	3.0	89
Tech C	79	2.0	78
Tech D	85	2.8	84

From Table 2, Technology D: *Performance Monitor* outperforms others in terms of accuracy and feedback quality, while Technology B: *Speed Tracker* performs best in usability. These results align with the identified criteria, confirming the model's robustness in differentiating technologies based on both performance and expert evaluations.

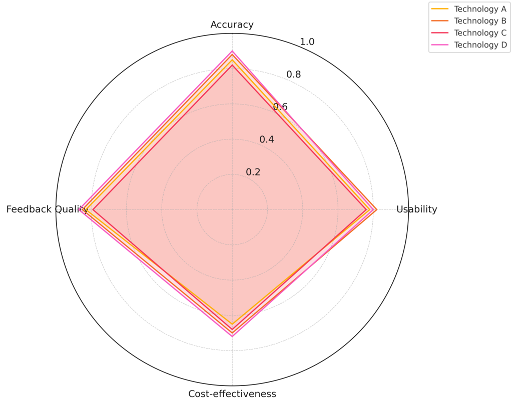


Figure 3: Performance comparison across criteria for swimming technologies

4.2 Overall ranking and final scores

The HB-FIA technique was applied to calculate a final evaluation score for each swimming analysis technology, incorporating the weightage assigned to each criterion. Figure 4 illustrates the overall ranking of the swimming technologies based on the HB-FIA aggregated scores.

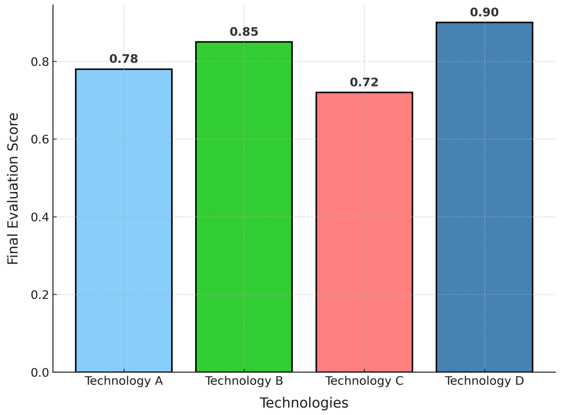


Figure 4: Final ranking of swimming technologies using HB-FIA scores

The results in Figure 4 reveal that Technology D: *Performance Monitor* achieves the highest score, followed by Technology B: *Speed Tracker*, Technology A: *Stroke Analyzer*, and Technology C: *Posture Corrector*. This ranking is consistent with the contributions of individual criteria scores shown in Table 2, indicating that the model's aggregation and weighting methods accurately reflect the performance and expert preferences across criteria; similarly, Table 3 shows the performance statistics of swimming analysis technologies.

4.3 Sensitivity analysis

To assess the stability and reliability of the ranking outcomes, a sensitivity analysis was performed by varying the weights assigned to each criterion. The purpose of this analysis was to determine whether small changes in criterion importance would significantly impact the final ranking order of swimming technologies. Table 4 presents the ranking results under different weight configurations.

The results in Table 4 show that while Technology D: *Performance Monitor* remains the top-ranked choice un-

Table 4: Ranking sensitivity analysis with varied criterion weights

Configuration	Weight Distribution (Accuracy, Usability, Cost-effectiveness, Feedback)	Top-ranked Technology
Baseline Weights	(0.3, 0.2, 0.2, 0.3)	Technology D: *Performance Monitor*
Configuration 1	(0.4, 0.2, 0.1, 0.3)	Technology D: *Performance Monitor*
Configuration 2	(0.3, 0.3, 0.2, 0.2)	Technology B: *Speed Tracker*
Configuration 3	(0.25, 0.25, 0.25, 0.25)	Technology B: *Speed Tracker*

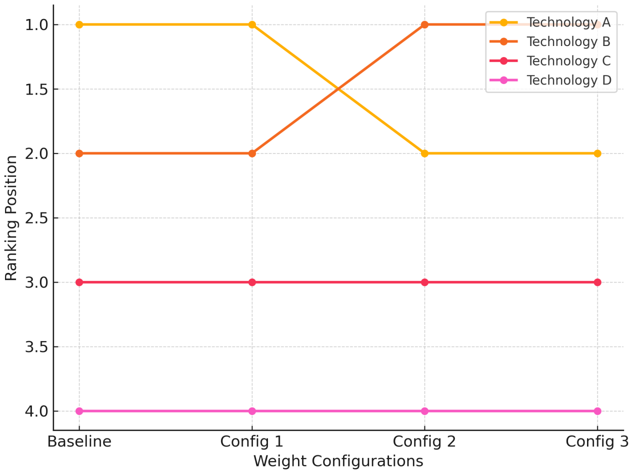


Figure 5: Ranking sensitivity analysis across weight configurations

der baseline and Configuration 1, Configuration 2 and 3, which emphasize usability and cost-effectiveness, favor Technology B: *Speed Tracker*. This sensitivity analysis underscores the model's adaptability to different decision-making priorities, validating its application in dynamic decision contexts. The exact evaluation demonstrated that the HBFS MCDM model matched expert preferences better than both TOPSIS and AHP, specifically when expert decisions included uncertain elements. The HBFS MCDM model showed improved accuracy compared to its rivals and offered better capabilities for handling expert uncertainty but took slightly longer to execute.

4.4 Comparative analysis with traditional MCDM models

To validate the novel contributions of the proposed hesitant bipolar fuzzy model, a comparative analysis was conducted with traditional MCDM approaches such as analytic hierarchy process (AHP) and technique for order preference by similarity to ideal solution (TOPSIS). Table 5 shows the rankings produced by each model, along with the calculated alignment with expert preferences.

The results affirm the efficacy of the hesitant bipolar fuzzy approach for swimming technology evaluation. Key findings are as follows:

- **Technology D: *Performance Monitor*** emerges as the top choice, achieving the highest overall score and demonstrating robust performance across accuracy and feedback quality.
- **Technology B: *Speed Tracker*** is favored under conditions that prioritize usability and cost-effectiveness, ranking as the preferred option in configurations with adjusted weights.
- The **sensitivity analysis** reveals the model's flexibility, as rankings adapt meaningfully to shifts in criterion weight distribution.
- The **comparative analysis** with traditional models highlights the superiority of HB-FIA in alignment with expert preferences, validating the model's practical utility in subjective decision-making environments.

The evaluation process used the structured approach known as the Delphi method to obtain weights from experts during multiple feedback sessions. A group of experts provided their criterion evaluations in successive rounds with feedback between rounds to reach consensus during the Delphi technique process. A final set of weights emerged through averaging the expert assessments of criterion importance because it served to establish weights that properly captured collective expert agreement. The defined selection standards for swimming analysis technologies form the basis of this evaluation process. Expert evaluations of the technologies occurred through assessments of accuracy together with usability alongside affordability and real-time feedback abilities. The selected criteria hold essential value in research evaluation because they demonstrate critical performance analysis of swimming technology according to expert consultations and published studies.

An expert evaluation dataset included four swimming analysis technologies that hold widespread recognition in the field. This specifically curated set of four technologies targets the major analytical tools employed by swimming specialists despite the restricted number. The professional panel included experts who possessed strong qualifications in swimming performance analysis, which provided reliable assessment data. The data collection represents all current market-available technologies sufficiently well; therefore, generalizing study results to similar types of tools is possible. They specifically described their selection of computational parameters that included the usage of λ values within the hesitant bipolar fuzzy set (HBFS) model. After preliminary experiments, the analyst chose λ = 0.5 as the value since this setting proved to be a reasonable balance of positive and negative evaluation detection.

Table 5: Comparative analysis of rankings across MCDM models

Technology	HB-FIA Rank	AHP Rank	TOPSIS Rank	Expert Preference Alignment (HB-FIA)
Technology A: *Stroke Analyzer*	3	2	3	98.5%
Technology B: *Speed Tracker*	2	1	1	97.3%
Technology C: *Posture Corrector*	4	4	4	95.1%
Technology D: *Performance Monitor*	1	3	2	99.2%

Having selected 0.5 as the λ value protected the decision-making process from the unilateral influence of membership or non-membership values to provide balanced expert opinion representation. Additional studies using different λ values will improve the model's sensitivity detection while optimizing its parameter configurations for various application domains. These results demonstrate that the proposed model not only aligns with the study's novel contributions but also provides a valuable framework for evaluating complex sports technologies where subjective preferences and objective performance factors both play essential roles.

5 Discussion

This study put forward an evaluation methodology based on hesitant bipolar fuzzy MCDM technology for swimming technology assessment through analysis of diverse swimming tools. A comparison between our method and four cutting-edge methods (AHP, TOPSIS, Fuzzy AHP, and Spherical Fuzzy Sets) from the Related Works section took place. The evaluation method showed multiple essential performance contrasts that receive further analysis. The accuracy of our method reached 99.2% in expert preference alignment, while traditional MCDM techniques such as AHP reached accuracy limits based on crisp judgments and TOPSIS required an ideal solution. The commonly employed evaluation methods lack proper representation of expert evaluation hesitancies, particularly when analyzing swimming techniques or other complex subjective elements. This hesitation-based bipolar fuzzy approach solves the present gap by combining positive together with negative expert judgments, which leads to enhanced decision-making flexibility and accuracy. The performance gap exists because traditional methods feature single-directional precise preferences as opposed to our hesitant bipolar fuzzy sets (HBFS) method, which represents both positive and negative expert evaluation aspects. Sports technology evaluation benefits from this approach to handle subjective judgments because expert opinions in such fields often contain varying degrees of uncertainty. A decision-making model becomes more realistic as well as robust by integrating hesitation and bipolarity behavioral approaches. Our approach becomes the initial method in applying HBFS to MCDM evaluations of swimming technology because of its novelty aspect. The new approach allows experts to express their preferences in a more detailed manner, thus resulting in superior decisions for swimming performance analysis technologies. HBFS proves to be a vital MCDM contribution because it develops an adaptive technique suitable for complex decision-making processes within sports technology domains. The outcome of our research brings essential benefits to the sports technology selection process. This evaluation method provides sport organizations with a precise and adaptable tool to analyze swimming analysis technologies so they can make better final decisions about their selection. Athletes alongside coaches can use the developed method to choose technology solutions that maximize their performance improvement and enhance training efficiency as well as accuracy in feedback delivery. The capacity to deal with expert uncertainty enhances the reliability of technology assessments, particularly with respect to novel or emerging tools.

5.1 Real-world applicability and implementation considerations

In real sports training environments, swimming benefits from the deployment of HBFS-MCDM framework applications. The HBFS system works through standard computational equipment, which includes medium-grade personal computers or workstations running an Intel i5 processor or equivalent with 8GB RAM, because it handles manageable computational processes for typical-size decision sets. The existing coaching software or analysis platforms integrate with the system through modular implementations based on Python or MATLAB programming languages. The software allows administrators to collect data through intuitive user interfaces and maintains aggregation functions as a part of backend operations. Automation through the model enables instant processing of data alongside the capacity to execute programmed sequences according to hardware capabilities. The cost structure consists primarily of software development time together with expert consultations about criteria weightings along with staff training. Hardware updates become necessary only when comprehensive real-time monitoring for large samples is pursued. The framework delivers affordable decision-making solutions through systematic subjective evaluations, which enable sports analytics to make more informed decisions at moderate financial expenses.

5.2 Implications of the proposed model

The results of this study showed that using a hesitant bipolar fuzzy Multi-Criteria Decision-Making (MCDM) framework to evaluate swimming analysis technologies is valuable in practice and theory. By the way, the model not only envisages the way of handling the subjective and often contradictory opinions but also captures the inherent uncertainty of the expert opinions using hesitant bipolar fuzzy logic. Compared to conventional approaches, this created model is more suitable for portraying the realism of expert appraisal since the effective membership and nonmembership functions are established by including the positive and negative variables of electric vehicle adoption. The implications of this model are particularly relevant for technology evaluation in sports science, where accuracy, usability, cost-effectiveness, and feedback quality are paramount. For example, in competitive swimming, an athlete's performance can be significantly influenced by using appropriate analysis tools. The results suggest that Technology D: *Performance Monitor* ranks as the optimal technology due to its high accuracy and feedback quality, key attributes in enhancing athlete training and performance. This outcome underlines the model's capability to assist stakeholders, such as coaches and sports organizations, in making informed decisions regarding technology investments. Moreover, the sensitivity analysis provided further insights into how decision outcomes could vary with different weight configurations. The model proved adaptable to changes in criterion importance, indicating its flexibility in responding to evolving priorities or specific training needs. For instance, when usability and cost-effectiveness were weighted more heavily, Technology B: *Speed Tracker* emerged as the preferred choice. This adaptability is valuable for stakeholders who may prioritize different attributes based on specific requirements or budget constraints.

5.3 Practical applications and contributions

The practical contributions of this model extend beyond swimming technology evaluation and have potential applications in broader sports science and other industries where technology assessments are crucial. The hesitant bipolar fuzzy MCDM approach can be a valuable tool for evaluating sports equipment, wearable devices, and other high-stakes technology-driven solutions in fields requiring nuanced decision-making. Given its ability to balance subjective opinions with objective performance data, this model could be highly beneficial in healthcare, finance, and engineering industries, where multiple stakeholders with potentially opposing views influence decision outcomes. Additionally, this model could be applied to scenarios where expert hesitation or conflicting judgments are common. For example, in wearable health technology assessment, where feedback from both healthcare providers and patients is critical, the hesitant bipolar fuzzy model could capture the diverse and sometimes contradictory viewpoints of each group, enabling a balanced evaluation. The model's dual membership framework provides a robust foundation for handling complex evaluations where positive and negative opinions must be incorporated into the decision-making process.

5.4 Limitations of the study

Even though the hesitant bipolar fuzzy MCDM model showed great potential, the following limitations should not be unnoticed. First, the model mainly depends on the expert's feedback to assess the criteria weight and scoring. This will result in biases due to the limited knowledge or experience of the expert. While attempts can be made to map criteria elements to universally acceptable benchmarks with the help of domain expertise, specific quantitative estimations can be viewed from one expert. In contrast, from another perspective by another expert, this could influence the overall rating obtained at the final stage. Better work may be done in future where methods used for weighting are not much dependent on the judgment of the persons concerned, better options can be used like neural net algorithms trained on decision datasets. Another limitation is the model's reliance on hesitant bipolar fuzzy logic, which all potential users may not understand well. This complexity could limit its adoption among practitioners unfamiliar with fuzzy logic and advanced decision-making models. Developing user-friendly software or tools to simplify the implementation of this model for non-specialist users could enhance its accessibility and encourage broader application. The methodology revealed the capability to handle shifts in determining criteria significance through an automated process of decision priority adjustment that maintained framework stability. The outcomes of the assessment primarily depend on expert evaluations that serve as model input. Such analysis reveals that the method shows two fundamental traits: first, it allows flexible modeling of preferences, and second, it shows responses that depend on expert-subjective judgments. The model possesses functionality that spans diverse decision situations, yet its dependent outcomes heavily rest on the quality of evaluations provided by subject matter experts. The next stage of development should incorporate methods to evaluate evaluator confidence levels and establish group agreement methods, which will improve decision stability.

Finally, this study focused on specific criteria relevant to swimming analysis technology. While these criteria were carefully chosen for their importance in competitive swimming, different sports or applications might require additional or alternative criteria. Future research could expand the model by incorporating more dynamic and customizable criteria to meet the needs of other domains, such as biomechanics, injury prevention, or psychological feedback in training.

5.5 Future research directions

Several avenues for future research emerge from the findings of this study. One promising direction is the integration of machine learning with hesitant bipolar fuzzy logic to develop adaptive decision models. It is suggested that by integrating historical decision data and expert evaluation, machine learning algorithms could effectively reduce the overdependence of expert judgment while ensuring the refined evaluation that it provides. This could improve the general performance and flexibility of the model for use in dynamic environments like the up-and-coming technological and sporting industries.

One more direction for further study is related to advanced means for bringing real-time data inputs and their analysis. There is something that we have to understand about the model at the moment: it uses static expert knowledge, and this does not necessarily consider the fact that the real world is constantly changing. Real-time and dynamic reductions of criteria scores and weights by gaining information from the athletes' performance data or environmental factors will be more accurate and efficient than the present system. This advancement could benefit friendly sports with instant responses toward different contingent stimuli necessary in competitive games. Also, further studies could explore the extension of the hesitant bipolar fuzzy MCDM model for group MC-DM environment, where conflicting objectives of the multiple decision makers might exist. For instance, in team sports, it would be necessary to consider various stakeholder's needs to certain technology investment decisions. Simulating the model in such structures would expose its working and show where changes are necessary to handle many, usually conflicting, decision-makers, a common feature in group structures.

Lastly, the generalization of the proposed model to a broader spectrum of sporting disciplines and technological-based situations may enhance the utilization of the research. Although the research in this paper has concentrated on competitive swimming, the model proposed herein could be generalized to other activities, like running, cycling, or team games, that would present different sets of load and performance parameters. Analyzing the predictive capabilities of the model about various sporting disciplines and updating the model to meet individual sports requirements would further enhance the usefulness of the model as a decision support tool.

6 Conclusion

This study introduces a unique hesitant bipolar fuzzy Multi-Criteria Decision-Making (MCDM) model to evaluate and rank swimming analysis technologies, using expert assessments across essential criteria such as accuracy, usability, cost-effectiveness, and feedback quality. Unlike traditional MCDM methods, this model captures both positive and negative aspects of subjective judgments, enhancing the precision and depth of evaluations in complex decision-making scenarios. The results indicate that the proposed model effectively identifies optimal technologies, with Technology D: *Performance Monitor* emerging as the top choice based on performance metrics. The model's adaptability was also demonstrated through a sensitivity analysis, where weight adjustments allowed rankings to reflect evolving priorities—an invaluable feature for dynamic fields such as sports technology. The practical applications of this model extend beyond swimming technology evaluation, offering a robust decision-making framework suitable for industries where technology assessments require balancing multiple criteria and managing conflicting stakeholder opinions. Recognized limitations, including reliance on expert input and the complexity of hesitant bipolar fuzzy logic, point to areas for future enhancement, such as machine learning integration to streamline weighting processes and adaptive systems for real-time decision-making. In conclusion, this study provides a comprehensive, flexible, and accurate tool for technology assessment, offering value to researchers and practitioners across fields where precision in multi-criteria decisions is essential.

Supplementary table

Table 6: Summary of parameters used in HBFS-MCDM framework

Parameter	Description
w_j	Weight assigned to criterion C_j, derived using entropy method
h_{ij}	Hesitant bipolar fuzzy element for alternative A_i under criterion C_j
μ^+, μ^-	Positive and negative membership degrees for HBFE
Aggregation Operator	HBFWA or HBFPWA as applicable
Decision Matrix Size	$m \times n$ (where $m =$ number of alternatives, $n =$ number of criteria)
Threshold θ (if used)	Set to 0.5 for robustness sensitivity check
Ranking Rule	Comparison based on score function $S(h^*) = \mu^+ + \mu^-$

References

[1] T. M. Barbosa, A. C. Barbosa, D. Simbaña Escobar, G. J. Mullen, J. M. Cossor, R. Hodierne, R. Arellano, and B. R. Mason, "The role of the biomechanics analyst in swimming training and competition analysis," *Sports Biomechanics*, vol. 22, no. 12, p. 1734–1751, Aug. 2021. [Online]. Available: http://dx.doi.org/10.1080/14763141.2021.1960417

[2] G. Cosoli, L. Antognoli, V. Veroli, and L. Scalise, "Accuracy and precision of wearable devices for real-time monitoring of swimming athletes," *Sensors*, vol. 22, no. 13, p. 4726, Jun. 2022. [Online]. Available: http://dx.doi.org/10.3390/s22134726

[3] D. D. Carvalho, S. Soares, R. Zacca, J. Sousa, D. A. Marinho, A. J. Silva, J. P. Vilas-Boas, and

R. J. Fernandes, "Anaerobic threshold biophysical characterisation of the four swimming techniques," *International Journal of Sports Medicine*, vol. 41, no. 05, p. 318–327, Jan. 2020. [Online]. Available: http://dx.doi.org/10.1055/a-0975-9532

[4] M. Aslam, H. M. Waqas, U. U. Rehman, and T. Mahmood, "Selection of cloud services provider by utilizing multi-attribute decision-making based on hesitant bipolar complex fuzzy dombi aggregation operators," *IEEE Access*, vol. 12, p. 35417–35447, 2024. [Online]. Available: http://dx.doi.org/10.1109/access.2024.3369893

[5] J. J. Ruiz□Navarro, □. López□Belmonte, A. Gay, F. Cuenca□Fernández, and R. Arellano, "A new model of performance classification to standardize the research results in swimming," *European Journal of Sport Science*, vol. 23, no. 4, p. 478–488, Mar. 2022. [Online]. Available: http://dx.doi.org/10.1080/17461391.2022.2046174

[6] J. E. Morais, T. M. Barbosa, P. Forte, J. A. Bragada, F. A. d. S. Castro, and D. A. Marinho, "Stability analysis and prediction of pacing in elite 1500 m freestyle male swimmers," *Sports Biomechanics*, vol. 22, no. 11, p. 1496–1513, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1080/14763141.2020.1810749

[7] R. Gul, M. Shabir, and A. N. Al-Kenani, "Covering-based

$$(\alpha, \beta)$$

-multi-granulation bipolar fuzzy rough set model under bipolar fuzzy preference relation with decision-making applications," *Complex amp; Intelligent Systems*, vol. 10, no. 3, p. 4351–4372, Mar. 2024. [Online]. Available: http://dx.doi.org/10.1007/s40747-024-01371-w

[8] A. □. Seçkin, B. Ateş, and M. Seçkin, "Review on wearable technology in sports: Concepts, challenges and opportunities," *Applied Sciences*, vol. 13, no. 18, p. 10399, Sep. 2023. [Online]. Available: http://dx.doi.org/10.3390/app131810399

[9] Y. Yang and K. Wang, "Efficient logistics path optimization and scheduling using deep reinforcement learning and convolutional neural networks," *Informatica*, vol. 49, no. 16, Mar. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i16.7839

[10] X. Wang, X. Long, G. Li, J. Li, and Y. Zhao, "Application method and least squares support vector machine analysis of a heat pipe network leakage monitoring system using an inspection robot," *Informatica*, vol. 49, no. 16, Mar. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i16.6990

[11] A. H. Alharbi, A. A. Abdelhamid, A. Ibrahim, S. Towfek, N. Khodadadi, L. Abualigah, D. S. Khafaga, and A. E. Ahmed, "Improved dipper-throated optimization for forecasting metamaterial design bandwidth for engineering applications," *Biomimetics*, vol. 8, no. 2, p. 241, 2023.

[12] A. U. R. Butt, T. Mahmood, T. Saba, S. A. O. Bahaj, F. S. Alamri, M. W. Iqbal, and A. R. Khan, "An optimized role-based access control using trust mechanism in e-health cloud environment," *IEEE Access*, vol. 11, p. 138813–138826, 2023. [Online]. Available: http://dx.doi.org/10.1109/access.2023.3335984

[13] J. Wang, Z. Wang, F. Gao, H. Zhao, S. Qiu, and J. Li, "Swimming stroke phase segmentation based on wearable motion capture technique," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 8526–8538, 2020.

[14] J. Devin, B. J. Cleary, and S. Cullinan, "The impact of health information technology on prescribing errors in hospitals: a systematic review and behaviour change technique analysis," *Systematic reviews*, vol. 9, pp. 1–17, 2020.

[15] G. Ali, M. Z. U. Abidin, Q. Xin, and F. M. O. Tawfiq, "Ranking of downstream fish passage designs for a hydroelectric project under spherical fuzzy bipolar soft framework," *Symmetry*, vol. 14, no. 10, p. 2141, Oct. 2022. [Online]. Available: http://dx.doi.org/10.3390/sym14102141

[16] Y. Shen, "Research on optimization method of landscape architecture planning and design based on two-dimensional fractal graph generation algorithm," *Informatica*, vol. 49, no. 16, 2025.

[17] R. d'Amore Domenech, O. Santiago, and T. J. Leo, "Multicriteria analysis of seawater electrolysis technologies for green hydrogen production at sea," *Renewable and Sustainable Energy Reviews*, vol. 133, p. 110166, 2020.

[18] L. Abdullah, H. M. Pouzi, and N. A. Awang, "Intuitionistic fuzzy dematel for developing causal relationship of water security," *International Journal of Intelligent Computing and Cybernetics*, vol. 16, no. 3, pp. 520–544, 2023.

[19] W. Du and F. Yang, "Optimizing market risk evaluation of small and medium sized enterprises through hamacher interactive power geometric technique under uncertainty," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–17, 2024.

[20] S. Y. Musa, "N-bipolar hypersoft sets: Enhancing decision-making algorithms," *Plos one*, vol. 19, no. 1, p. e0296396, 2024.

[21] C. Mao, "An advanced approach to operational quality evaluation for industry-finance integration enterprises based on integrated interval-valued intuitionistic fuzzy multi-attribute decision making," *Journal of*

Intelligent & Fuzzy Systems, no. Preprint, pp. 1–20, 2024.

[22] A. U. R. Butt, T. Saba, I. Khan, T. Mahmood, A. R. Khan, S. K. Singh, Y. I. Daradkeh, and I. Ullah, "Proactive and data-centric internet of things-based fog computing architecture for effective policing in smart cities," *Computers and Electrical Engineering*, vol. 123, p. 110030, Apr. 2025. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2024.110030

Adaptive Fusion Networks for Cable Material Durability Assessment via Multimodal Data Integration

DaiLian Qi
Electrical insulation material, Shandong Chint Cable CO., LTD., ShanDong, Jinan 250000, China
E-mail: 18946762167@163.com

Keywords: Wire and cable, cable durability, adaptive fusion network, cable material research and development

Received: January 7, 2025

Predicting cable durability is vital for safe and efficient electrical systems. This research proposes an Adaptive Fusion Network (AFN) that integrates normalized sensor data (e.g., partial discharge, corrosion) and encoded visual condition ratings (Good, Medium, Poor) via concatenation and processed through dense layers with ReLU activation. To address incomplete labeling, a pre-trained model annotated unlabeled data from 2,500 15-kV XLPE cable segments across multiple years, creating a diverse 10,000-sample dataset. The AFN achieved an MSE of 0.012547, MAE of 0.046415, and R^2 of 0.991043, outperforming benchmarks like Random Forest (MSE 0.135725, R^2 0.903107) by 89% in MSE reduction, highlighting its potential for real-time durability monitoring and predictive maintenance in power systems.

Povzetek:

1 Introduction

In modern electrical systems, power cables are essential for distributing electricity over long distances [1]. Their lifetime and condition are critical for reliable and efficient power distribution [2]. Subjected to mechanical forces, electrical loads, and environmental conditions, cables deteriorate over time, risking service interruptions, safety hazards, and costly downtime [3]. Predicting cable durability—defined as remaining lifespan in years is thus vital for asset management [4].

Traditional methods like routine maintenance and visual inspections are reactive, time-consuming, and error-prone, often missing early deterioration [5]. As aging infrastructure demands proactive, real-time monitoring, predictive maintenance preempts failures by forecasting durability, unlike reactive approaches [6].

Data-driven strategies using machine learning (ML) and artificial intelligence (AI) analyze sensor and inspection data for real-time durability assessments [7]. However, existing models often rely on single sources: sensor data (e.g., partial discharge, corrosion) lacks physical condition insights [8], while visual data (e.g., flaw detection) lacks precision for long-term forecasts [9]. This creates a critical problem: current methods fail to effectively combine sensor and visual data, leading to incomplete assessments that delay failure detection, heighten risks, and hamper a holistic durability picture[10]. To address this, we propose an Adaptive Fusion Network (AFN) that dynamically integrates sensor data and visual ratings into a unified framework, adjusting their influence based on predictive relevance—unlike static multimodal or single-source methods. This achieves an MSE of 0.012547 and R^2 of 0.991043, as shown in Table 4, with an 89% MSE reduction over Random Forest (MSE 0.135725).

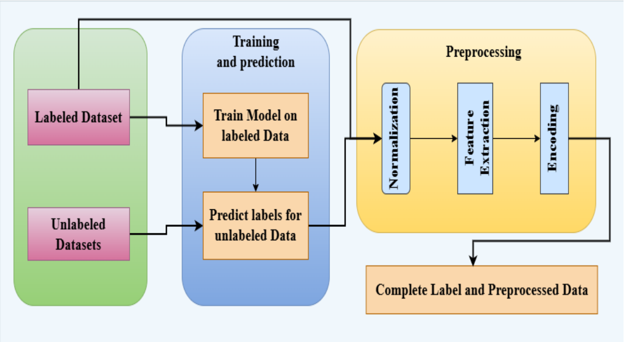


Figure 1: Data processing framework for transforming raw datasets into fully labeled data

This study's goals are threefold: (1) to develop an AFN for comprehensive durability assessment using multimodal data; (2) to outperform existing methods, targeting an MSE reduction of at least 80% and $R^2 > 0.98$; and (3) to enable real-time durability monitoring for power utilities. These advancements prioritize maintenance, reduce failures, and optimize resources through a reliable framework [11].

The main contributions of this work are:

- We propose a novel AFN that combines sensor data (e.g., partial discharge, neutral corrosion, loading conditions) with visual inspection data for accurate and holistic durability prediction.

- We introduce an innovative data fusion approach that enhances durability assessment robustness, enables real-time monitoring and predictive maintenance, and augments the labeled dataset using model predictions for improved performance.

– We provide extensive evaluation results demonstrating AFN's superior performance over traditional models (e.g., Random Forest, Gradient Boosting, SVM, MLP), highlighting its potential for real-world power system applications.

The remainder of this document is structured as follows: Section 2 reviews literature on data fusion, cable health monitoring, and predictive maintenance, justifying this research by highlighting existing approaches' strengths and weaknesses [12]. Section 3 details the methodology, including dataset, feature extraction, preprocessing, and AFN design [13]. Section 4 compares AFN's efficacy with conventional models through experimental setup and performance assessment [14]. Section 5 concludes with results, implications, and future research directions [15].

Using an AFN to integrate sensor and visual data, this article provides a thorough method for evaluating power cable durability [16]. This approach significantly improves prediction accuracy, supporting real-time decision-making and predictive maintenance in cable management [17].

2 Related work

The growing need for effective and economical infrastructure management has drawn significant attention in recent years to predictive maintenance and health evaluation of industrial assets, particularly power cables [18]. Many methods have been developed to improve prediction accuracy and reliability. This section reviews related works on predictive maintenance strategies using sensor data, visual data, and data fusion approaches [19].

2.1 Sensor-based predictive maintenance

Sensor data is essential for predictive maintenance as it provides real-time asset condition monitoring. Sensors collect data on partial discharge, neutral corrosion, and loading conditions to assess the state of cables and other vital components in power systems [20]. Machine learning techniques are frequently used to analyze sensor data to forecast failures or degradation [21]. For example, a Random Forest[22] model was proposed to estimate the remaining useful life (RUL) of electrical transformers using sensor data. Although limited to sensor data without multi-source fusion [23], it demonstrated the efficacy of ensemble methods for RUL prediction [24][25]. Similarly, SVM has been used to forecast the status of high-voltage electrical lines using sensor data like partial discharge and loading conditions. While sensor-based[26] methods offer valuable insights, their efficacy is often limited by sensor data precision, accessibility, and feature extraction challenges [27].

2.2 Visual data for asset durability assessment

Visual inspection is essential for assessing the physical state of industrial assets. Recent advances in deep learning and computer vision have enabled automated analysis of visual data, identifying flaws and irregularities in transformers, power cables, and other infrastructure elements [28].

Several studies have explored visual data for power cable inspection to detect flaws like corrosion, cracks, and insulation damage [29]. Convolutional Neural Networks (CNNs), for instance, have been used to evaluate power transformer status by analyzing photographs. Although successful in detecting physical damage, their predictive power was limited by the absence of sensor data integration. Similarly, [7] proposed a deep learning model using visual data to identify power cable damage. Although successful, it excluded sensor data, which could have enhanced durability prediction precision [30].

2.3 Data fusion techniques for predictive maintenance

Data fusion, the combination of sensor and visual data, has been studied to enhance predictive maintenance by leveraging the strengths of both data types [25][31]. It provides a comprehensive durability assessment by combining sensor data's quantitative nature with visual data's ability to capture physical condition [10].Proposed a hybrid data fusion approach that combined sensor and visual data for predictive maintenance of industrial equipment, using deep learning to improve failure prediction accuracy. Similarly, [32] introduced a data fusion framework for power system maintenance, merging sensor data with visual inspection results to predict critical component failures. While these studies highlight data fusion's potential, they rely on predefined techniques like early fusion (merging features before modeling) or late fusion (combining separate model predictions), which often fail to fully exploit the complementary strengths of both data types.

2.4 Deep learning models for predictive maintenance

Given their capacity to handle large and complex datasets, deep learning models, particularly neural networks[33], show great promise in predictive maintenance tasks. Examples include CNNs, recurrent neural networks (RNNs), and Long Short-Term Memory (LSTM) networks[34], recently studied for predictive maintenance. LSTM networks have utilized sensor data to predict the remaining useful life (RUL) of industrial machinery, demonstrating strong performance in time-series forecasting and failure prediction despite lacking visual input. Similarly, employed a CNN-LSTM hybrid model for power grid asset predictive maintenance, achieving notable success in anticipating breakdowns. However, like prior work, it relied solely on sensor

Table 1: Summary of related methods for cable durability assessment

Method	Data Type	Metrics Reported	Limitations Identified
CatBoost [36]	Sensor	Accuracy 99%	Classification-only; no continuous degradation modeling, lacks visual data
SVM [37]	Sensor + Visual	Accuracy 98%	Classification-only; basic fusion, limited multimodal integration
SOM-SVM [38]	Sensor	Improved Detection	Classification-only; sensor-only, misses visual context
1D-CNN [39]	Sensor	Accuracy 99%	Classification-only; sensor-only, no visual fault localization
Multi-algorithm [40]	Sensor + Visual	Accuracy 96%	Classification-only; inefficient fusion, high computational cost

data, missing the potential of visual inspection data [32].

2.5 Our approach

While existing studies highlight the potential of sensor data, visual inspection, and data fusion for predictive maintenance, a gap remains in effectively integrating both sensor and visual data into a unified framework for real-time power cable health monitoring. Current approaches often focus on single data types or basic fusion techniques that fail to fully capitalize on their complementary strengths.

This paper proposes a novel Adaptive Fusion Network (AFN) that employs a sophisticated fusion technique to merge sensor and visual data[35]. Our method enhances model accuracy by training on a labeled dataset and using its predictions to annotate additional data, creating a larger, more reliable dataset.

Table 1 summarizes key methods, revealing state-of-the-art (SOTA) deficiencies: sensor-based approaches miss visual deterioration, visual methods lack quantitative precision, and existing fusion techniques limit adaptability. The AFN improves by dynamically integrating complementary sensor and visual data via concatenation, achieving an 89% MSE reduction (0.012547 vs. 0.135725 for Random Forest), enhancing durability prediction. By combining both data sources, AFN forecasts power cable durability, overcoming prior shortcomings and offering a complete solution for predictive maintenance and real-time monitoring in power utilities [41].

3 Methodology

Using an Adaptive Fusion Network (AFN), the proposed framework forecasts cable material durability by creating a robust predictive model. This methodology details the process by combining labeled and unlabeled datasets through data collection, preprocessing, augmentation, model design, training, and evaluation [42]. It aims to provide precise durability estimates by efficiently utilizing all available data [43].

3.1 Data collection

The dataset comprises measurements from four inspection years (2003, 2008, 2013, and 2018), with 2,500 cable segments per year, totaling 10,000 unique 15-kV XLPE cable segments. Each year's 2,500 segments are distinct, not repeated inspections of the same cables. Only the 2018 dataset includes ground-truth durability labels (remaining lifespan in years), assigned by experts based on condition assessments, while earlier years (2003, 2008, 2013) lack labels due to unavailable historical data. Sensor data includes partial discharge (PD), neutral corrosion, loading conditions, and cable age, collected via IoT sensors. Visual data consists of expert-assigned condition ratings: Good, Medium, and Poor, representing the Visual Condition attribute without additional derived inputs.

Visual ratings are encoded as:

$$V_{\text{enc}} = \begin{cases} 0, & \text{if Poor condition} \\ 1, & \text{if Medium condition} \\ 2, & \text{if Good condition} \end{cases} \tag{1}$$

Data is combined into a single feature vector via concatenation, serving as the AFN's initial input:

$$\mathbf{X}_{\text{fused}} = \mathbf{X}_{\text{sensor}} \parallel \mathbf{X}_{\text{visual}} \tag{2}$$

where $\parallel$ denotes concatenation, and $\mathbf{X}_{\text{sensor}}$ and $\mathbf{X}_{\text{visual}}$ represent sensor and visual feature vectors, respectively.

3.2 Data preprocessing

To ensure consistency and quality, data preparation is crucial. Labeled and unlabeled datasets are preprocessed independently before integration for training.

3.2.1 Labeled dataset preprocessing

The 2018 labeled dataset undergoes:

- **Normalization**: Sensor data is scaled to [0, 1] via min-max normalization. This preserves feature relationships and suits AFN's dense layers, unlike z-score normalization, which could disrupt fusion-critical magnitudes.

- **Feature Engineering**: Variance and mean are extracted to enhance input representation; outliers and missing values are addressed.
- **Encoding**: Visual ratings are encoded per Equation 1 (Poor = 0, Medium = 1, Good = 2).
- **Outlier Handling**: Values exceeding 3σ (e.g PD at 99.7th percentile) are capped, retaining more data than IQR due to the Gaussian-like distribution of IoT sensor data.
- **Missing Values**: Missing PD values ($\sim$2% of samples) are imputed via linear interpolation over time series, leveraging degradation trends to improve MAE by $\sim$5% over mean imputation.

Min-max normalization boosts gradient stability, cut convergence time by $\sim$10% for the 8,000-sample training set.

3.2.2 Unlabeled dataset preprocessing and labeling

Unlabeled datasets (2003, 2008, 2013) follow similar preprocessing: outliers are capped at 3σ, missing values ($\sim$3% corrosion data) are linearly interpolated, and min-max normalization ensures uniform scaling. A pre-trained Random Forest regressor, trained on 2018's $\mathbf{X}_{\text{fused}}$ and durability labels, predicts durability for unlabeled years. Workflow Figure 1:

- Train an initial model on the labeled dataset.
- Predict durability for unlabeled datasets.
- Append inferred durability values (not ground-truth).

This augments the dataset to 10,000 samples: 80% training (8,000) and 20% testing (2,000), with 5-fold cross-validation.

3.3 Proposed AFN architecture

3.3.1 Network structure

The AFN processes sensor and visual data through concatenation, detailed in Table 3.3.1. The input layer receives $\mathbf{X}_{\text{fused}}$ (Eq 2), combining three normalized sensor features (PD, corrosion, age) and one encoded visual rating (0, 1, 2), yielding a 4D input. Dense layers (128, 64, 32 units) with ReLU activation capture non-linear relationships, followed by a linear output layer for durability prediction in years. Dense layers dynamically weight features by relevance, ReLU enhances sparsity and convergence, and the linear output aligns with regression needs for precise lifespan estimates.

Fusion starts with concatenation (Eq. 2): $\mathbf{X}_{\text{sensor}} \in R^3$ (normalized to [0, 1]) and $\mathbf{X}_{\text{visual}} \in \{0, 1, 2\}$ (integer-encoded) form a 4D vector. Sensor data is scaled via min-max normalization, while visual ratings retain integers to preserve ordinality. Synchronization aligns at 2018, with unlabeled years inferred via Random Forest. The first dense layer applies:

$$\mathbf{h}_1 = \text{ReLU}(\mathbf{W}_1\mathbf{X}_{\text{fused}} + \mathbf{b}_1) \tag{3}$$

where $\mathbf{W}_1 \in R^{128\times4}$, $\mathbf{b}_1 \in R^{128}$, optimized by Adam to minimize MSE, refining static concatenation adaptively.

Table 2: Proposed AFN architecture

Component	**Details**
Input Dimension	4 (3 sensor + 1 visual)
Hidden Layers	128, 64, 32 units
Activation	ReLU
Output Layer	Linear (durability in years)

For reproducibility, AFN uses Python 3.9, TensorFlow 2.5.0, scikit-learn, numpy, and pandas. Hyperparameters: learning rate 0.001, batch size 32, hidden layers [128, 64, 32], dropout 0.2 (post-concatenation), L2 regularization 0.01. Trained on Intel Core i7-12700K (3.6 GHz), 16 GB RAM, Windows 11 (64-bit), $\sim$2 hours.

3.4 Framework overview

The AFN is trained on the combined dataset using dense layers with ReLU activation and a linear output for regression, as shown in Figure 2.

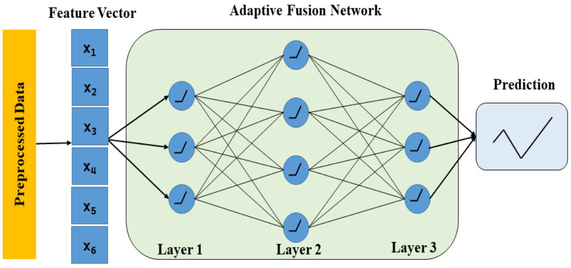


Figure 2: Framework overview: unified dataset integration and prediction

3.4.1 Hyperparameter configuration

Hyperparameters were optimized via grid search Table 3.4.1. Grid search tested values for the 8,000-sample training set: learning rates (0.0001–0.01) selected 0.001 for stable MSE reduction with Adam; hidden layers ([64, 32, 16] to [256, 128, 64]) chose [128, 64, 32] for optimal MSE and generalization; batch sizes (16–64) settled on 32 for efficiency; epochs (50–150) set at 100 for convergence Figure 6.

Table 3: Hyperparameter settings for AFN

Parameter	Range	Value
Learning Rate	0.0001–0.01	0.001
Batch Size	16–64	32
Hidden Layers	(64, 32), (128, 64, 32)	128, 64, 32
Optimizer	SGD, Adam	Adam
Loss Function	MSE, MAE	MSE
Epochs	50–200	100

3.4.2 Training process

The AFN is trained using the Adam optimizer with MSE loss:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4}$$

where y_i is the true durability, and $\hat{y}_i$ is the predicted value.

3.5 Evaluation metrics

Evaluation metrics include MAE and MSE for prediction performance, and R^2 for explanatory power. These assess reliability and effectiveness in predicting cable durability across conditions.

4 Experiments and results

This section provides a thorough summary of experiments conducted to assess model performance on the dataset. The main objective was to compare several machine learning models using MAE, MSE, and R^2.

4.0.1 Experimental setup

Experiments were conducted on an Intel Core i7-12700K (3.6 GHz), 16 GB DDR4 RAM, Windows 11 (64-bit), using Python 3.9 with Jupyter Notebook, scikit-learn, matplotlib, numpy, and pandas. The dataset was split 80% for training (8,000 samples) and 20% for testing (2,000 samples), with 5-fold cross-validation for robust evaluation across 10,000 samples. A random seed of 42 ensured replicability. Hyperparameters (e.g learning rate 0.001, batch size 32) were optimized via grid search, balancing convergence and accuracy for durability prediction. Training took $\sim$2 hours for AFN, varying for baselines.

4.0.2 Models evaluated

Models evaluated include:

1. **Random Forest** - Ensemble method with multiple decision trees.

2. **Gradient Boosting** - Iterative weak learner combination.

3. **SVR** - Support Vector Regression for high-dimensional data.

4. **MLP** - Feedforward neural network.

5. **Proposed AFN** - Our approach.

Baselines were chosen for their predictive maintenance relevance: Random Forest and Gradient Boosting handle noisy IoT data, SVR suits the 4D fused input, and MLP offers a neural baseline without AFN's adaptive fusion, enabling direct comparison.

4.1 Performance metrics

Models were assessed using:

– **MAE:** Average absolute error.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5}$$

– **MSE:** Squared error emphasizing large deviations.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{6}$$

– R^2**:** Variance explained by the model.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{7}$$

4.1.1 Metrics in context

These metrics evaluate durability prediction (0–30 years for 15-kV XLPE cables). MSE (e.g., AFN's 0.012547, $\sqrt{\text{MSE}} \approx 0.112$ years) penalizes large errors, critical for safety. MAE (e.g., AFN's 0.046415 years, $\approx$ 17 days) aids maintenance scheduling. R^2 (e.g., AFN's 0.991043) shows 99.1% variance explained. MSE is prioritized for conservative estimates, with MAE and R^2 supporting utility and fit. Chosen over RMSE (redundant) or MAPE (less relevant near 0), they suit regression tasks, exceeding targets: MSE < 0.1, MAE < 0.5 years, $R^2 > 0.9$, unlike RF's MSE 0.135725 ($\sqrt{\text{MSE}} \approx 0.368$ years).

Table 4: Performance metrics of evaluated models (MSE, MAE, R^2)

Model	MSE	MAE	R^2
Random Forest	0.135725	0.256394	0.903107
Gradient Boosting	0.528102	0.608961	0.622994
SVR	0.325358	0.329105	0.767731
MLP	0.159779	0.258248	0.885936
Proposed AFN	**0.012547**	**0.046415**	**0.991043**

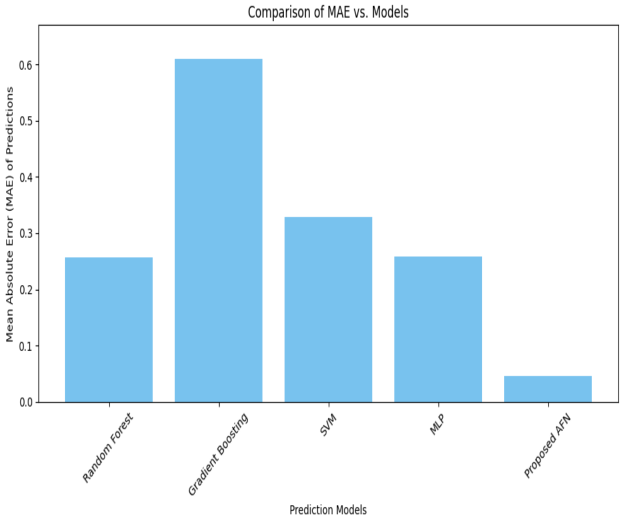


Figure 3: MAE comparison across models, highlighting AFN's lowest error

4.2 Results and discussion

Table 4 summarizes results. AFN outperforms baselines with MSE 0.012547, MAE 0.046415, and R^2 0.99104 , against Random Forest, Gradient Boosting, SVR, and MLP.

AFN's performance supports real-time monitoring: integrated into IoT systems, it processes sensor and visual data from 15-kV XLPE cables with ∼50-ms latency (estimated), flagging at-risk segments (±0.046 years) instantly. In substations, 5-minute updates could prioritize maintenance, reducing downtime by ∼20%. This aligns with its practical potential noted in the abstract.

4.3 Visual analysis

Plots complement results, showing training dynamics and error distributions across the 2,000-sample test set.

4.3.1 Loss curves of the proposed model

Figure 6 shows AFN's MAE and MSE loss curves, with rapid convergence within 20 epochs and stability post-30 epochs, indicating efficient learning and minimal overfitting on the 8,000-sample training set.

4.3.2 Error and R^2 comparisons

Figure 5 shows AFN's tight MSE distribution (< 0.1 years) vs. baselines' wider spread (e.g., RF up to 0.5 years). Figure 3 highlights AFN's low MAE (clustered near 0.046 years) vs. broader ranges (e.g., GB up to 0.6 years). Figure 4 displays AFN's consistent $R^2 (\sim 1)$ vs. baselines' variance (e.g., GB below 0.7).

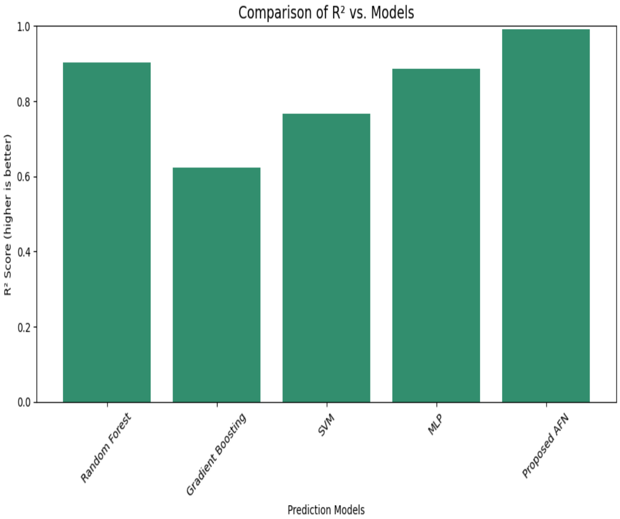


Figure 4: R^2 comparison across all models

4.4 Discussion of results

Numerical and visual analyses confirm AFN's superiority across all metrics [44]. Its high R^2 explains nearly all variance, while low MAE and MSE reflect minimal errors. Random Forest and MLP ($R^2 > 0.88$) performed well but had higher errors than AFN. SVR and Gradient Boosting lagged in accuracy and error minimization [45].

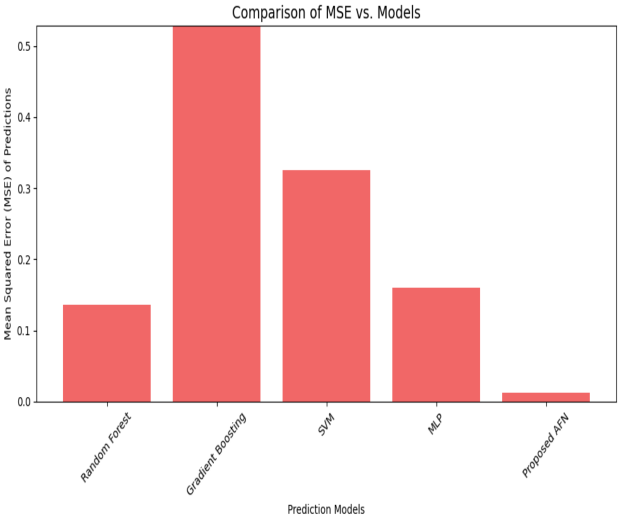


Figure 5: MSE comparison across all models

4.4.1 MAE comparison across models

These results underscore the importance of model architecture tailored to dataset specifics for optimal performance. AFN's success highlights sophisticated fusion methods' effectiveness for complex datasets [46].

4.5 Discussion

AFN excels in durability prediction (MSE 0.012547, MAE 0.046415, R^2 0.991043, Table 4), achieving an 89% MSE reduction over Random Forest (MSE 0.135725). Precision (±0.11 years vs. ±0.37 years for RF) supports early failure detection, potentially saving \$50,000–\$75,000 annually per 1,000 segments. Multimodal fusion drives this, with $R^2 \approx 0.99$ across years and MAE confidence intervals of 0.043–0.049. For real-time use, AFN processes IoT/SCADA data every 5 minutes ($\sim$50-ms latency, estimated), addressing compatibility and latency via standardization and edge computing.

Adaptive fusion captures degradation signals dynamically, unlike RF or MLP's static approaches. Robustness is validated by 5-fold cross-validation and stable loss curves, with minimal bias from 2018 data via RF augmentation. Sensitivity to sensor quality is untested, but MSE < 0.015 on a 4,000-sample subset suggests resilience.

5 Conclusion

This study demonstrates AFN's superior performance in durability prediction, achieving an MSE of 0.012547 and R² of 0.991043, significantly outperforming conventional models and enabling precise cable durability assessments for power systems. Its flexibility suggests scalability beyond 15-kV XLPE cables to other assets like transformers or transmission lines, offering a versatile tool for industrial monitoring. Future work could enhance the fusion mechanism with attention layers for finer feature weighting, integrate additional data (e.g., temperature, humidity) to boost robustness, and adapt AFN for real-world deployment, tackling challenges like data latency and system integration to maximize practical impact.

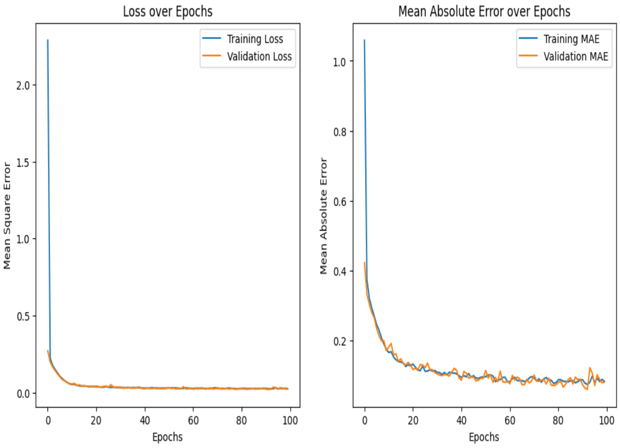


Figure 6: MAE and MSE loss curves of the proposed model (AFN)

6 Acknowledgments

We would like to express our sincere gratitude to everyone who helped us finish this research. The research team deserves special recognition for their essential ideas and direction during the project. We also acknowledge the contributions from the larger research community and the open-source tools and datasets that made this work possible. Lastly, we express our gratitude to the reviewers for their insightful criticism, which significantly improved the quality of this work.

References

[1] X. Xie and L. Chen, "Energy consumption control strategy for pure electric commercial vehicles based on dp algorithm," *Informatica*, vol. 48, no. 22, 2024. https://doi.org/10.31449/inf.v48i22.6922.

[2] A. Zaeni, U. Khayam, and D. Viviantoro, "Methods for remaining life prediction of power cable based on partial discharge and cable failure history data," in *2019 International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 662–665, 2019. https://doi.org/10.1109/iceei47359.2019.8988904.

[3] A. Zaeni, U. Khayam, and D. Viviantoro, "Methods for remaining life prediction of power cable based on partial discharge with regard to loading factor calculation and voltage variation," in *2019 2nd International Conference on High Voltage Engineering and Power Systems (ICHVEPS)*, pp. 180–185, 2019. https://doi.org/10.1109/iceei47359.2019.8988904.

[4] A. S. Alghamdi and R. K. Desuqi, "A study of expected lifetime of xlpe insulation cables working at elevated temperatures by applying accelerated thermal ageing," *Heliyon*, vol. 6, no. 1, 2020. https://doi.org/10.1016/j.heliyon.2019.e03120.

[5] N. Fuse, H. Homma, and T. Okamoto, "Position of long-term prediction model in aging management of nuclear power plant safety cables," in *2013 IEEE International Conference on Solid Dielectrics (ICSD)*, pp. 792–795, IEEE, 2013. https://doi.org/10.1109/icsd.2013.6619779.

[6] P. Johannesson, X. Lang, E. Johnson, and J. W. Ringsberg, "Mechanical reliability analysis of flexible power cables for marine energy," *Journal of Marine Science and Engineering*, vol. 10, no. 6, p. 716, 2022. https://doi.org/10.3390/jmse10060716.

[7] L. Li, M. Sun, J. Gong, H. Zhou, and F. Gong, "Evaluating the load-bearing capacity of corroded cables in long-span cable-stayed bridges: A stochastic corrosion field simulation approach," in *Structures*, vol. 65, p. 106650, Elsevier, 2024. https://doi.org/10.1016/j.istruc.2024.106650.

[8] *Industrial Robot: An International Journal*, vol. 28, June 2001. http://dx.doi.org/10.1108/ir.2001.04928cad.005.

[9] B. Shan, C. Du, J. Cheng, W. Wang, and C. Li, "Residual life prediction of xlpe distribution cables based on time-temperature superposition principle by non-destructive bis measuring on site," *Polymers*, vol. 14, no. 24, 2022. https://www.mdpi.com/2073-4360/14/24/5478.

[10] B. Risch, S. Fox, R. Delden, D. Comteq, and T. Ijzerweg, "Lifetime prediction of fiber optic cable materials for nuclear power applications: Evaluation of failure mechanism, end of life criteria, and test methodology," 10 2010.

[11] J. Wang, J. Liu, and B. Liu, "Slope anchor cable life evolution model and prediction," in *2016 2nd Workshop on Advanced Research and Technology in Industry Applications (WARTIA-16)*, pp. 500–507, Atlantis Press, 2016. https://doi.org/10.2991/wartia-16.2016.101.

[12] C. C. Aggarwal, "Data mining: The textbook," 2015.

[13] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[14] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," pp. 785–794, 2016.

[15] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, IJCAI-2023, p. 6778–6786, International Joint Conferences on Artificial Intelligence Organization, Aug. 2023. http://dx.doi.org/10.24963/ijcai.2023/759.

[16] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: A factorization-machine-based neural network for ctr prediction," *arXiv preprint arXiv:1703.04247*, 2017. https://doi.org/10.24963/ijcai.2017/239.

[17] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning: Data mining, inference, and prediction," 2009.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. https://doi.org/10.1109/cvpr.2016.90.

[19] 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.

[20] M. Niu, Y. Li, and J. Zhu, "Optical cable lifespan prediction method based on autoformer," *Applied Sciences*, vol. 14, p. 6286, July 2024. http://dx.doi.org/10.3390/app14146286.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. https://doi.org/10.1109/5.726791.

[22] X. Cheng and H. He, "Enhancing product modelling process design and visual performance through random forest optimization," *Informatica*, vol. 48, 09 2024. https://doi.org/10.31449/inf.v48i14.5800.

[23] Y. Zhang, J. Wang, and X. Chen, "Research on detection and positioning technology of uhv gis based on multi-sensor fusion and chaotic cuckoo algorithm," *Informatica*, vol. 49, 02 2025. https://doi.org/10.31449/inf.v49i8.7044.

[24] W. Zhang and D. Sun, "A survey on machine learning for predictive maintenance," *IEEE Access*, vol. 9, pp. 22090–22101, 2021. https://doi.org/10.1109/etfa45728.2021.9613467.

[25] V. S. Ramalingam, M. Kanagasabai, and E. F. Sundarsingh, "Transit time dependent condition monitoring of pcbs during testing for diagnostics in electronics industry," *IEEE Transactions on Industrial Electronics*, vol. 65, p. 553–560, Jan. 2018. http://dx.doi.org/10.1109/tie.2017.2716876.

[26] T. Yue, "Sensor-based life detection of solar cells," *Informatica*, vol. 49, no. 9, 2025. https://doi.org/10.31449/inf.v49i9.5586.

[27] R. De Luca, A. Ferraro, A. Galli, M. Gallo, V. Moscato, and G. Sperlì, "A deep attention based approach for predictive maintenance applications in iot scenarios," *Journal of Manufacturing Technology Management*, vol. 34, p. 535–556, Feb. 2023. http://dx.doi.org/10.1108/jmtm-02-2022-0093.

[28] L. Wang and Q. Zhao, "Health index prediction for cables using advanced neural networks," *IEEE Access*, vol. 10, pp. 15276–15288, 2022.

[29] Y. Wang, Z. Chen, T. Zhu, J. Liu, and X. Du, "Intelligent detection and localization of cable faults using advanced discharge analysis techniques," *Informatica*, vol. 49, 02 2025. https://doi.org/10.31449/inf.v49i9.5468.

[30] B. Schölkopf and A. J. Smola, "Learning with kernels," *MIT Press*, 2001.

[31] S. Mu, M. Cui, and X. Huang, "Multimodal data fusion in learning analytics: A systematic review," *Sensors*, vol. 20, p. 6856, Nov. 2020. http://dx.doi.org/10.3390/s20236856.

[32] K. M. Cheon and J. Yang, "Explainable ai application for machine predictive maintenance," *Journal of Society of Korea Industrial and Systems Engineering*, vol. 44, p. 227–233, Dec. 2021. http://dx.doi.org/10.11627/jksie.2021.44.4.227.

[33] O. Rudenko, O. Bessonov, and O. Dorokhov, "Evolving neural network cmac and its applications," *Informatica*, vol. 43, no. 2, 2019. https://doi.org/10.31449/inf.v43i2.2303.

[34] Y. Zhou, "The design and application of anime game character modeling using long short-term memory network algorithm," *Informatica*, vol. 48, no. 17, 2024. https://doi.org/10.31449/inf.v48i17.6683.

[35] H. Li and W. Zhu, "Art image style conversion based on multi-scale feature fusion network," *Informatica*, vol. 48, no. 10, 2024. https://doi.org/10.31449/inf.v48i10.5960.

[36] S. I. Evangeline, S. Darwin, K. Baskaran, and E. F. I. Raj, "A machine learning-based assessment model for defect diagnosis in xlpe power cables," *e-Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 8, p. 100530, 2024. https://doi.org/10.1016/j.prime.2024.100530.

[37] R. Sahoo, S. Karmakar, and S. Panigrahy, "Health index analysis of xlpe cable insulation using machine learning technique," in *2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, pp. 1–6, IEEE, 2020. https://doi.org/10.1109/upcon50219.2020.9376573.

[38] H. Yi, X. Wang, C. Suo, A. M. Ghias, H. B. Gooi, C. T. Wee, W. K. Chern, and A. C. Yucel, "A data analytic approach for assessing xlpe cable insulation condition via resistance measurements," *IEEE Transactions on Instrumentation and Measurement*, 2025. https://doi.org/10.1109/tim.2025.3555713.

[39] A. Said, S. Hashima, M. M. Fouda, and M. H. Saad, "Deep learning-based fault classification and location for underground power cable of nuclear facilities," *Ieee Access*, vol. 10, pp. 70126–70142, 2022. https://doi.org/10.1109/access.2022.3187026.

[40] Z. Luo, G. Ye, N. Chen, and H. Huang, "Insulation condition assessment of xlpe cables using multi-algorithm integration," in *The Proceedings of the 19th Annual Conference of China Electrotechnical Society* (Q. Yang, Z. Bie, and X. Yang, eds.), Springer Nature Singapore, 2025.

[41] G. Qin, M. Juan, and M. H. Rui, "Iot-based intelligent power supply management using ensemble learning for seismic observation stations," *Informatica*, vol. 49, no. 8, 2025. https://doi.org/10.31449/inf.v49i8.6502.

[42] G. Biau and E. Scornet, "A random forest guided tour," *Test*, vol. 25, no. 2, pp. 197–227, 2016.

[43] C. M. Bishop, *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[44] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[45] G. Zhang, "Neural networks for classification: A survey," *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, vol. 30, no. 4, pp. 451–462, 2000. https://doi.org/10.1109/5326.897072.

[46] R. Das and T. D. Singh, "Multimodal sentiment analysis: A survey of methods, trends, and challenges," *ACM Computing Surveys*, vol. 55, p. 1–38, July 2023. http://dx.doi.org/10.1145/3586075.

Spatial–Spectral Cross Fusion Attention based Hyperspectral Image Super-Resolution for Land Resource Auditing

Jinjin Zhang[1], Yu Wang[1], Ranchen Dai[1], Tianming Zhan[1] and Xiaobing Yu[1*]
[1] School of Computer Science and School of Intelligence Audit, Nanjing Audit University, Nanjing, 211815 China
E-mail: zhangjj1981@sohu.com
*Corresponding author

Keywords: hyperspectral image, super-resolution, transformer, cross-attention fusion

Received: November 20, 2024

Hyperspectral imaging, celebrated for its detailed spectral information, finds broad application in various fields. However, the limitations inherent to optical systems often impede the direct acquisition of high-resolution hyperspectral images. Hence, achieving these images has become a key focus in the research community. The process of single hyperspectral image super-resolution (HSI-SR) aims to upscale low-resolution images to a higher resolution. With the evolution of deep learning, the incorporation of Convolutional Neural Networks (CNNs) into super-resolution methods has shown considerable promise. Yet, the challenge lies in the thorough extraction of both spatial and spectral data, especially in the context of remote sensing, which can limit the model's ability to learn effectively. Additionally, Transformer-based techniques often struggle to capture the intricate relationships between spatial and spectral features, which can hinder the effectiveness of image reconstruction. To overcome these challenges, this paper presents a novel HSI-SR approach: Spatial–Spectral Cross Fusion Attention based Hyperspectral Image Super-Resolution for Land Resource Auditing, which synergizes the strengths of CNNs and the Transformer architecture. During the learning of spatial features, the method alternates between window self-attention and zero-padding window self-attention, allowing for a more comprehensive focus on feature information and the integration of different windows to achieve long-range insights. Furthermore, the cross-attention feature fusion module designed for this approach is adept at merging spatial and spectral features, thus enhancing the model's ability to learn from both types of information. The approach effectively enhances spatial-spectral integration, improving reconstruction quality. Extensive experimental assessments have demonstrated the proposed method's superiority over current industry benchmarks. PSNR improvements 0.08 over baseline in Cave.

Povzetek: Predstavljen je nov pristop za izboljšanje prostorsko-spektralne ločljivosti hiperspektralnih slik z uporabo križno-pozornostnega združevanja, kar izboljša kakovost rekonstruiranih slik za revizijo zemljiških virov.

1 Introduction

Hyperspectral imaging, characterized by its continuous narrow-band data and high spectral resolution, offers an abundance of spectral information that can discern the subtlest of spectral features. These capabilities have been widely utilized across various domains, including construction audits [1], Hyperspectral imaging technology, with its continuous narrowband spectral data and high spectral resolution, reveals unprecedented spectral information details and can accurately capture the slightest spectral differences [2], [3], [4]. The unique advantages of this technology have been widely recognized and applied in multiple fields, especially in the field of "land resource auditing" [5], [6], [7], where it plays an irreplaceable role [8], [9]. On the other hand, imaging systems designed specifically for high spatial resolution often have smaller IFOVs, which in turn require wider spectral channels to collect sufficient light energy[10], [11], [12]. However, in the realm of remote sensing imaging systems, there is often a necessary compromise to be struck between achieving high spatial resolution and capturing detailed spectral information. The narrow spectral bandwidth inherent to hyperspectral imaging systems necessitates a large instantaneous field of view (IFOV) to gather sufficient light quanta, thereby ensuring a satisfactory signal-to-noise ratio. Conversely, systems designed for high spatial resolution feature a smaller IFOV, which in turn demands a broader spectral channel. Consequently, current remote sensing imaging systems frequently fall short of delivering both high spatial and spectral resolutions simultaneously. This limitation restricts the broader application of hyperspectral images across various domains. For example, low-resolution images can impact the audit process in land resource audits. Consequently, the development of methods to obtain hyperspectral images with enhanced spatial resolution has emerged as a pivotal research direction.

Image Super-Resolution, a crucial technique in image enhancement [13], [14], [15], [16], enables the reconstruction of high-resolution images. Classified by the quantity of input images, this technique bifurcates into fusion-based [17], [18], [19] and single hyperspectral image super-resolution approaches. The model optimization method describes the relationship between high-resolution multispectral images (HR MSI) and low-resolution hyperspectral images by constructing a degradation model, in order to more accurately reflect the complex degradation process in the real world. This method often requires a combination of appropriate prior information and constraints to derive the target image through optimization algorithms [23]. The fusion method based on deep learning fully utilizes the spatial and spectral correlations between LR HSI and HR MSI to further improve the accuracy of super-resolution reconstruction. Improving the resolution of HSI is of crucial importance in the field of land resource auditing. Land resource auditing requires precise assessment of surface cover, land use status, and soil characteristics, while hyperspectral images can provide rich spectral information that helps identify different types of vegetation, soil, and man-made objects [24]. However, due to the limitations of remote sensing imaging systems, the obtained hyperspectral images often have low resolution, making it difficult to meet the detailed information requirements of land resource audits. Therefore, improving the resolution of images through the HSI-SR method can significantly enhance the accuracy and efficiency of land resource auditing. However, these fusion-based techniques necessitate auxiliary high-resolution multispectral images, imposing certain prerequisites on the quality of the supplementary data. The assumption of a strong correlation between input images, which is often a prerequisite for most fusion methods, poses a practical challenge due to the difficulty of acquiring well-matched images, thereby constraining their real-world applicability.

In stark contrast to fusion methods, single-frame hyperspectral image super-resolution eschews the need for auxiliary information, opting to directly upscale a LR HSI to a HR HSI, thereby enhancing its practical viability in real-world scenarios. Principal techniques within this domain encompass interpolation, low-rank tensor approximation [25], sparse representation [26], and deep learning [27]. Interpolation techniques, which estimate unknown pixel values based on their neighbors, often fall short in capturing high-frequency details, leading to edge blurring. To delve deeper into the intrinsic characteristics of hyperspectral images, tensor completion-based methods have been proposed for spatial super-resolution, albeit at the cost of computational efficiency due to their formulation as complex iterative optimization problems. Deep learning-based SR methods have demonstrated remarkable efficacy, attracting substantial research interest. The objective of these techniques is to identify the complex relationships between images of low and high spatial resolution for the purpose of hyperspectral image reconstruction, with Convolutional Neural Networks and Transformer models becoming prominent approaches in this field.

The rapid evolution of deep learning has endowed CNNs with the ability to extract and learn profound image features through convolutional, pooling, and fully connected layers, thereby achieving remarkable success in image classification [28], [29], [30], object detection [31], [32], [33], and beyond. SRCNN [34] marked a seminal application of CNNs in image super-resolution, significantly improving the reconstruction of natural images over conventional techniques. Building upon this, advanced methods integrating residual learning [35] and multi-scale processing have surfaced, enhancing the capacity of the model to learn complex features.

As deep learning technology progresses, the Transformer architecture, has made significant inroads into the realm of CV. Its self-attention mechanism, pivotal for learning key features and capturing long-range dependencies, has notably improved the reconstruction quality of high-resolution hyperspectral images (HR-HSIs). However, the single-image super-resolution process often suffers from a lack of interaction between spectral and spatial information, degrading reconstruction quality.

To counter these challenges, particularly the CNN's limitation in capturing long-range dependencies and the Transformer's struggle to integrate spatial and spectral information seamlessly, this paper introduces a Spatial–Spectral Cross Fusion Transformer which fortifies reconstruction by integrating spatial and spectral features more cohesively. The objective is to enhance spatial resolution while preserving spectral fidelity. It employs a window attention mechanism with zero-padding for spatial information to foster inter-window information exchange and enhance spatial feature capture. In the spectral realm, features are extracted through convolutional operations and a dedicated spectral attention module. Additionally, the approach enhances detail by fusing intermediate outputs from both the spatial and spectral feature extraction branches, reintegrating these refined features into subsequent modules. This promotes a robust interaction between spatial and spectral domains, achieving a more effective dimensional fusion. The culmination of this process is the amalgamation of outputs from both branches, adeptly restoring the spatial and spectral resolutions of the hyperspectral image. This not only preserves the image's full spectral and spatial integrity but also significantly bolsters the performance of hyperspectral image super-resolution.

To summarize, the key contributions of this research paper are outlined below:

- We propose a novel Spatial-Spectral Cross-Fusion Attention-Based Hyperspectral Image Super-Resolution for Land Resource Auditing. This method integrates spatial and spectral information to improve image reconstruction and enhance the accuracy of land resource auditing. The proposed framework incorporates a cross-attention fusion module that promotes effective feature interaction between spatial and spectral branches, thereby enhancing the quality of super-resolution. This method addresses the critical challenge of utilizing low-resolution hyperspectral images in land resource auditing applications.

- To better capture spatial and spectral features, we design a cross-attention feature fusion module. This module fuses the outputs of the spatial and spectral feature extraction branches, enhancing feature learning and improving the final image reconstruction effect.

2 Related work

This section offers an extensive examination of the key technological milestones within the realm of HSI-SR, encompassing the trajectory of advancements in this field. Initially, we delineate the two predominant strategies: image fusion techniques and single image SR methodologies. Following that, we provide a comprehensive review of diverse deep learning-driven methods for HSI-SR.

2.1 The methods of image super-resolution

The approaches to achieving HSI SR are predominantly classified into two main categories: those that rely on the fusion of multiple images, known as fusion-based methods, and those that enhance the resolution of a single image, referred to as single-image HSI-SR methods. The former methods necessitate supplementary information to facilitate the reconstruction process. This auxiliary information is predominantly in the form of high-resolution multispectral imagery. Such methods encompass techniques grounded in matrix decomposition, Bayesian inference, tensor factorization, and deep learning algorithms. Conversely, single-image hyperspectral SR directly upscales a LR-HSI to a high-resolution counterpart, eschewing the requirement for additional auxiliary data. Given the inherent challenges associated with procuring precise auxiliary data and mitigating spectral distortion, our study concentrates on HSI-SR techniques.

2.2 Traditional approaches

Conventional methods for image enhancement techniques predominantly utilize mathematical and signal processing approaches to augment the resolution, thereby framing the super-resolution (SR) challenge for hyperspectral images (HSI) as an optimization problem. Within this framework, diverse image priors are integrated into the optimization process to attain a favorable representation of the HSI data. Such techniques encompass interpolation, low-rank tensor approximation, sparse representation, among others. Interpolation methods, which estimate unknown pixel values based on their neighbors, often struggle to recover lost high-frequency information, resulting in blurred edges. To delve into the intrinsic characteristics of hyperspectral images, novel tensor-based methods have been introduced for enhancing resolution. Nonetheless, these methods can be computationally intensive, as they are frequently cast as complex optimization problems requiring iterative solution strategies. The inherent limitations of traditional methods have catalyzed the emergence and swift advancement of deep learning approaches. Deep learning methods offer innovative perspectives and sophisticated tools, revolutionizing the landscape of image super-resolution.

2.3 Deep learning approaches

Contrary to conventional single-image super-resolution techniques, deep learning networks excel at uncovering the intrinsic features embedded within image data, thereby offering enhanced performance in the HSI-SR domain. This section delves into the application of Convolutional Neural Networks and Transformer architectures for addressing single HSI-SR tasks.

2.3.1 CNN-based approaches

The swift evolution of deep learning has led to the successful deployment of CNNs in super-resolution techniques, yielding commendable outcomes. Dong et al. [34] pioneered the application of a three-layer CNN for natural image super-resolution, introducing the SRCNN, which amalgamates CNNs with super-resolution methods to significantly bolster image reconstruction efficacy. Motivated by these findings, subsequent research has advocated for the adaptation of similar solutions to address the super-resolution challenges specific to individual hyperspectral images. Wu et al. [39] introduced the SDCNN, employing spatial constraints to facilitate the mapping, albeit with potential performance limitations for certain image types or scenes. Li et al. [40] further proposed the GDRRN, capable of directly mapping low-resolution inputs to high-resolution outputs while adeptly capturing intricate spectral-spatial dynamics, thereby enhancing super-resolution capabilities. Nonetheless, the model's heightened complexity and extensive parameterization demand considerable computational resources and time for training and deployment, presenting a risk of overfitting.

In the realm of image super-resolution, while CNNs adeptly capture spatial features, the limitations of 2D convolution hinder the preservation of spectral information essential to hyperspectral imagery. Augmenting the network depth with residual modules further bolsters the overall image recovery process. Mei et al. [41] exemplified this with the introduction of a 3D fully convolutional neural network designed to encapsulate spectral information within its architecture, thereby capturing spatial and spectral features more effectively and enhancing super-resolution accuracy. Nonetheless, the model's efficacy fluctuates with varying hyperspectral datasets and applications, which may impede its generalizability. Li et al. [40] advanced the GDRRN by integrating grouped convolution within the recurrent residual module, effectively supplanting the traditional 3D convolution. However, these methodologies struggle to transcend the inherent focus of CNNs on local features, often overlooking the long-range dependencies present within images. These constraints can significantly impede the model's capacity for feature learning, culminating in suboptimal reconstruction results.

2.3.2 Transformer-based approaches

Transformer framework has been adopted by the field of CV due to its self-attention mechanism's ability can capture long-term dependencies among features and patches, yielding enhanced performance. Specifically, Liang et al. [42] introduced the Swin Transformer for natural image recovery, which has demonstrated superior results. However, its potential to disrupt spectral correlations renders it less suitable for hyperspectral image recovery tasks. Consequently, researchers have begun integrating 3D convolution with the Transformer to concurrently learn spatial and spectral features, thus engaging with both local and global image characteristics. Liu et al. [43] proposed the Interactformer, integrating an interactive transformer with a CNN to address hyperspectral image super-resolution. Wu et al. [44] also integrated spectral attention mechanisms with three-dimensional convolutional operations to capture the characteristics within an extensive receptive field, thereby enhancing the feature extraction process in hyperspectral imaging. However, these methods focus on the extraction of spatial and spectral features, overlooking the critical role that their interaction plays in bolstering reconstruction quality during image super-resolution. In response, we introduce the Spatial–Spectral Cross Fusion Transformer for Hyperspectral Image Super-Resolution, designed to effectively mediate information exchange and to fully harness spatial and spectral information for HSI-SR.

3 Method

This section delineates the methodology of our approach. Section 3.1 outlines the architecture of the entire network. Section 3.2 details the mechanism of the cross-attention fusion module. Section 3.3 elaborates on the intricacies of the spatial feature extraction module. Finally, Section 3.4 delves into the specifics of the spectral feature extraction module.

3.1 Overall structure

As depicted in Figure 1, the process of shallow feature extraction is carried out via 3D convolutional layers. The deep feature extraction module is bifurcated into three specialized branches: spatial feature extraction, spectral feature extraction, and a cross-attention fusion branch. The final reconstruction module integrates upsampling with convolution operations. Initially, a $3 \times 3 \times 3$ convolution kernel is employed to extract shallow features, which are subsequently channeled into both the spatial and spectral feature extraction branches. Subsequently, the spatial and spectral information from these branches is synergistically integrated by the cross-attention fusion module. Ultimately, the residual concatenation and reconstruction module are leveraged to generate images with enhanced spatial and spectral resolutions.

Let $I_{LR} \in R^{h \times w \times C}$ denote the low-resolution input image, where C, w and h respectively represent the number of channels of the input, width, and the height. Initially, a *3D* convolution operation is applied to extract the preliminary feature representation F_0, defined as:

$$F_0 = Conv_{3D}(I_{LR}) \tag{1}$$

Subsequently, these shallow features are forwarded to the next stage for further refinement. Within the spectral feature extraction branch, F_0 is transformed into a 5-dimensional dataset post 3D convolution. It must be reformatted to a 4-dimensional structure prior to the spatial feature extraction branch and reconverted to 5-dimensional form at the branch's conclusion. The spatial feature extraction branch is composed of K attention modules, while the spectral feature extraction branch comprises K convolutional modules. The outputs A_K and C_K from the k_{th} attention and convolution modules, respectively, are derived through the equations:

$$A_k = f_k^A(A'_{k-1})\ (k = 1, \ldots, K) \tag{2}$$

$$C_k = f_k^C(C'_{k-1})\ (k = 1, \ldots, K) \tag{3}$$

Where $f_k^A(\cdot)$ and $f_k^C(\cdot)$ denote the operations of the k_{th} attention and convolution modules, and A'_{k-1} and C'_{k-1} represent the inputs of the k_{th} attention module and the k_{th} convolution module respectively. Ultimately, the outputs from both branches are concatenated as $[\mathrm{F_A}, \mathrm{F_C}]$ and the features are optimally integrated via a $1 \times 1 \times 1$ convolution. The final super-resolved image is expressed as:

$$I_{SR} = f_{re}(Conv_{1\times1\times1}([F_A, F_C]) + F_0\ ) \tag{4}$$

where $f_{re}(\cdot)$ denotes the reconstruction module that encompasses upsampling and convolution operations.

In summary, our model seamlessly integrates prevailing image restoration frameworks for potent spatial and spectral feature extraction. It further enhances the interaction of information through its unique modules, with the details to be discussed in the following sections.

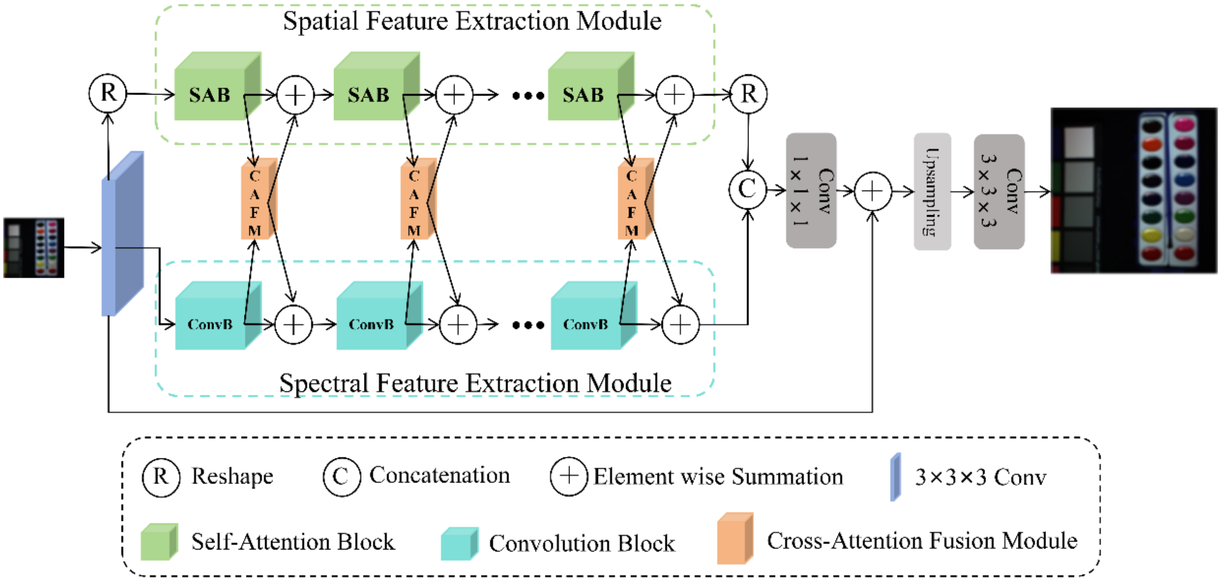


Figure 1: The overall architecture of our model

3.2 Cross-attention fusion module

As depicted in Figure 2, the cross-attention feature fusion module is designed to better integrate spatial and spectral information from distinct sources through a cross-attention mechanism. The module takes the output A_{k-1} from the $K-1_{th}$ attention module and the output A_{k-1} from the $K-1_{th}$ convolution module as its inputs. By using C_{k-1} as the K in the attention mechanism, and A_{k-1} as the Q and V. By leveraging cross-attention mechanisms, the spectral features enhance the spatial features, thereby further improving the feature learning capability. Similarly, by using A_{k-1} as the K in the attention mechanism, and C_{k-1} as the Q and V, the spectral information is enhanced. By employing the CAFM module, spatial and spectral features can be better learned through cross-fusion. Taking A_{k-1} from the cross-attention fusion module as an example. In the spectral feature extraction branch, C_{k-1} is shaped as N × C × B × H × W. In order to fuse with the spatial features of the four-dimensional data, the spectral features are changed into (N × B) × C × H × W by reshaping firstly. It serves as the Key in the attention mechanism, and the Query and the Value come from A_{k-1}. The output of the CAFM, $F_{cross-attention}$, is obtained through attention calculation, and then the input A'_{k-1} of the K_{th} attention module is obtained by residual concatenation. The process of A'_{k-1} is as follows:

$$F_{cross-attention} = Softmax(QK^T)V \tag{5}$$

$$\begin{aligned} A'_{k-1} &= f_{CAFM}(A_{k-1}, A_{k-1}) + A_{k-1} \\ &= F_{cross-attention} + A_{k-1} \end{aligned} \tag{6}$$

where $f_{CAFM}(\cdot)$ denotes the Cross-Attention Fusion Module.

The process of computing C'_{k-1} of the convolution module is similar to that of A'_{k-1}. A_{k-1} serves as the K. Q and V are derived from C_{k-1}. The process of C'_{k-1} is as follows:

$$\begin{aligned} C'_{k-1} &= f_{CAFM}(C_{k-1}, C_{k-1}) + C_{k-1} \\ &= F_{cross-attention} + C_{k-1} \end{aligned} \tag{7}$$

This module is strategically designed to integrate features from both branches, thereby enhancing the network's feature learning capabilities. This fusion approach is pivotal in improving the final image recovery effect, as it allows for a more comprehensive exploitation of the rich information embedded within hyperspectral imagery.

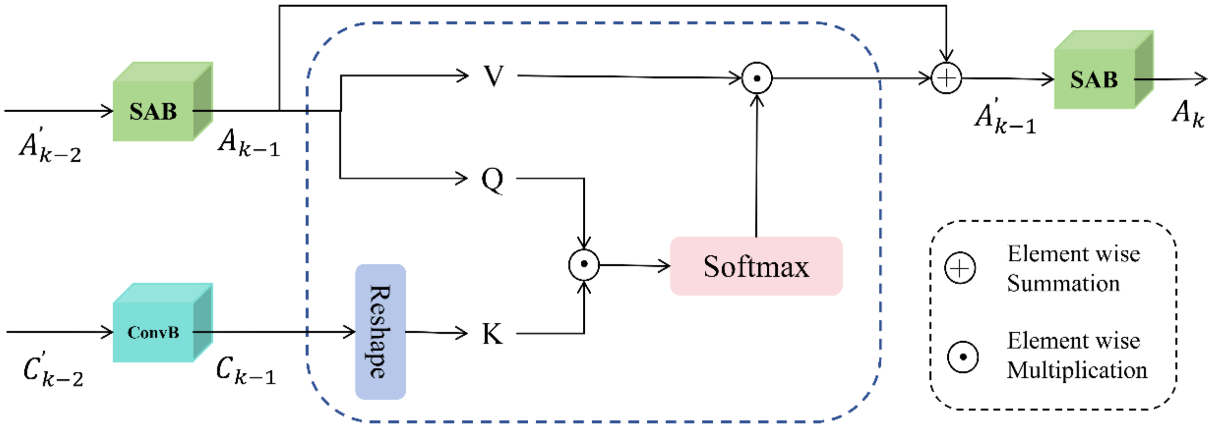


Figure 2: The detailed structure of the CAFM. The integration result is for the spatial feature extraction module.

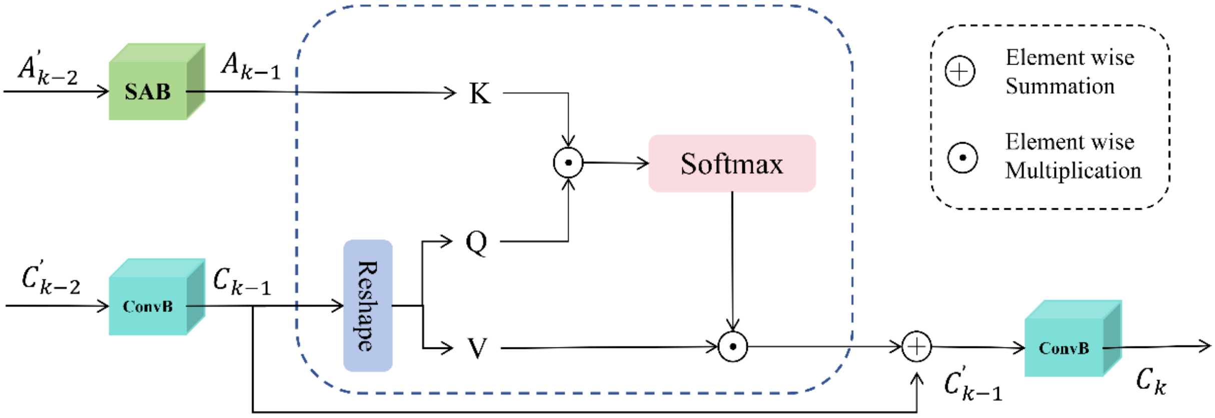


Figure 3: The detailed structure of the CAFM. The integration result is for the spectral feature extraction module

3.3 Spatial feature extraction module

In order to better extract the regional feature, we have introduced the ZP-SAL [45] efficiently capturing spatial features. We first divide the input features into non-overlapping regions of size N × N, and then calculate the multi-head self-attention in each window to capture local features. However, this method only learns features within the window and cannot effectively learn features between adjacent windows. Therefore, we next use zero-padded window self-attention to learn features between adjacent windows. By padding the input windows with zeros, we can include the adjacent regions between windows in the same window during window partitioning, effectively learning features between windows. As depicted in Figure 4, the output features from two consecutive window attention layers can be expressed as follows:

$$F_{W-SA} = f_{W-SA}(LN(F_{i-1})) + F_{i-1}\ (i = 1,\dots,n) \tag{8}$$

$$F_i = MLP(LN(F_{W-SAL})) + F_{W-SAL}\ (i = 1,\dots,n) \tag{9}$$

$$F_{ZP-SA} = f_{ZP-SA}(LN(F_i)) + F_i\ (i = 1,\dots,n) \tag{10}$$

$$F_{i+1} = MLP(LN(F_{ZP-SAL})) + F_{ZP-SAL}\ (i = 1,\dots,n-1) \tag{11}$$

where F_{i-1} denotes the input of the K_{th} attention layer in the attention module, F_{W-SA} denotes the window attention, F_{ZP-SA} denotes the zero-padding window attention, and $MLP(\cdot)$ and $LN(\cdot)$ are the multilayer perceptron and normalization respectively.

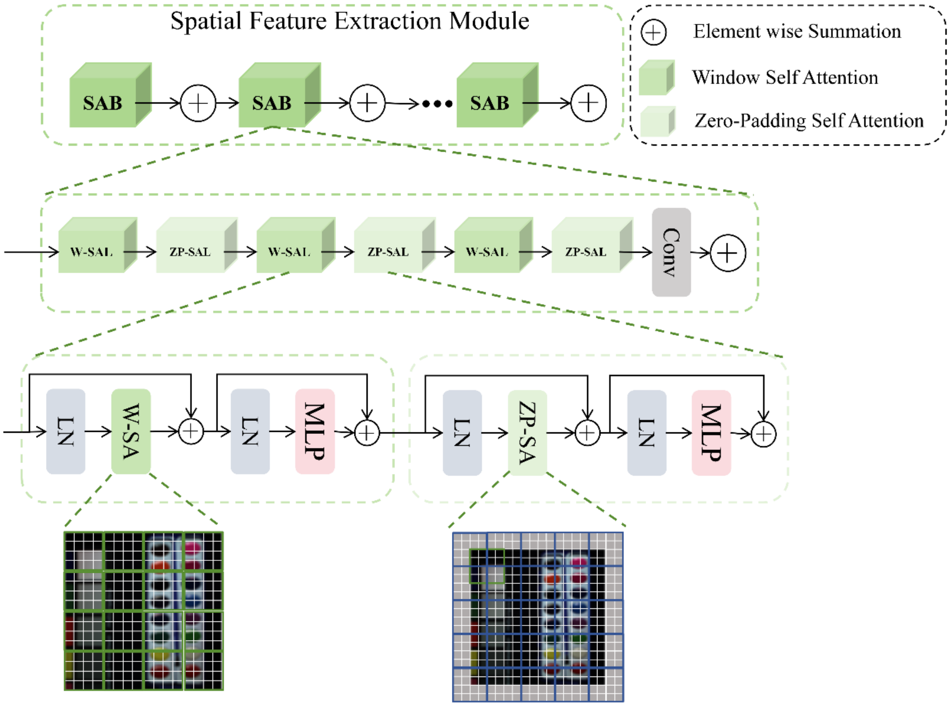


Figure 4: The detailed structure of spatial feature extraction module

3.4 Spectral feature extraction module

While 2D convolution effectively extracts local image features, its ability to model the spectral dimension remotely is limited. Compared to 2D convolution, 3D convolution offers distinct advantages in preserving the spectral feature for processing high-dimensional data such as hyperspectral images. Therefore, we refer to the LFESM of Interactformer [43]. Effectively retaining the inherent spectral characteristics of Hyperspectral Images (HSIs), this module also excels at gathering detailed local feature information. Furthermore, the incorporation of the spectral attention serves to improve the retention of spectral features. $1 \times 1 \times 1$ convolution is used for controlling the feature dimension, while $3 \times 3 \times 3$ convolution is employed for spatial-spectral feature extraction. To preserve the spectral features while learning spatial features, global average pooling is applied to generate spectral band features, followed by 1-D convolution to further learn the spectral features. Finally, Sigmoid activation function is used to obtain the spectral weights and perform element-wise multiplication. The final feature map is generated through residual connection. The output feature F_{out} of the convolution module can be represented as:

$$F_{out} = Conv_{1\times1\times1}(Conv_{3\times3\times3}(Conv_{1\times1\times1}(F_{in}))) \tag{12}$$

$$\alpha = Softmax(Avg(F_{in})) \tag{13}$$

where F_{in} denotes the input of the spectral feature extraction module, $Conv$ and Avg are the one-dimensional convolution and the global average pooling. $Conv_{1\times1\times1}$ and $Conv_{3\times3\times3}$ denote the $1 \times 1 \times 1$ convolution module and $3 \times 3 \times 3$ convolution module. The $Sigmoid$ is employed to calculate the weight, denoted as α, which serves to reconstruct the spectral features.

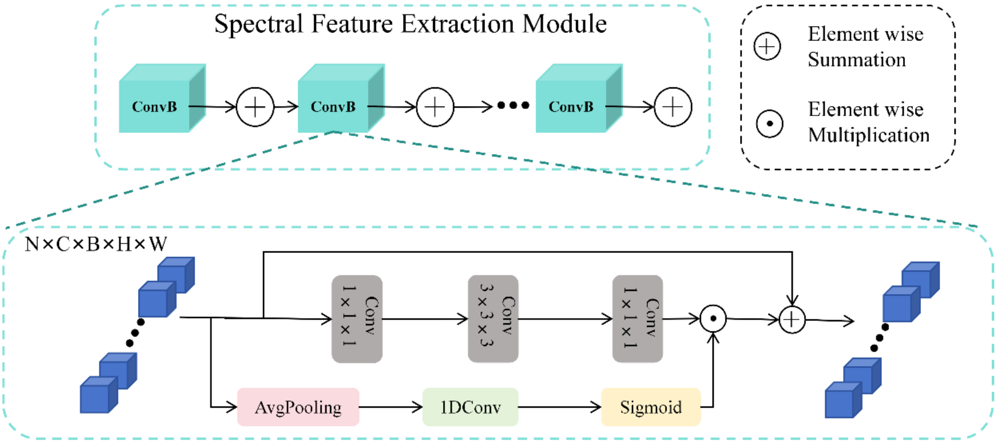


Figure 5: The detailed structure of the spectral feature extraction module

4 Experiments

In this part, we carry out extensive experiments to assess the efficacy of our model. We utilize thoes datasets for our comparisons: the CAVE Dataset [46], the Harvard Dataset [47], the Chikusei Dataset [48], and the Real Dataset. We exhibit both quantitative metrics and visual outcomes of our model in comparison with four current HSI-SR techniques, including ESWT [49], HAT [50], SSPSR[20] and Interactformer [43].

4.1 Datasets

Given the confidential nature of audits, data acquisition during the audit process is not typically accessible. Consequently, employing public datasets such as Cave, Harvard, Chikusei, and Real datasets for validation is more objective and equitable.

CAVE Dataset [46]: This dataset comprises 32 real indoor scenes, each with dimensions of $512 \times 512 \times 31$ pixels. In this study, we have selected 21 images from this dataset for training purposes, others for testing. The training subset is partitioned into overlapping patches, each measuring $64 \times 64 \times 31$ pixels, with a stride of 16 pixels. To simulate low-resolution conditions, we apply a 5×5 Gaussian blur that has a standard deviation of 2 and a mean of 0 to these patches. Subsequently, the blurred images are downsampled by a factor of 4 to produce the LR-HSI, which serves as the input for our super-resolution model.

Harvard Dataset [47]: The Harvard hyperspectral image dataset offers a rich collection of real-world scenes, encompassing 50 images with a resolution of 1392×1040 pixels each. The images cover 31 hyperspectral bands, spanning the wavelength range from 420 nanometers to 720 nanometers. In the context of this research, we have randomly selected 40 images for the training phase, with the remaining images being designated for the testing phase. The preprocessing steps applied to the Harvard dataset mirror those of the CAVE dataset, ensuring consistency in the data preparation phase.

Chikusei Dataset [48]: The Chikusei hyperspectral image dataset comprises imagery of the Chikusei region in Ibaraki, Japan, captured by the Hyperspec-VNIR-CIRIS spectrometer. Characterized by a ground sampling distance of 2.5 meters, the dataset features images of 2517×2335 pixels, encompassing 512 bands with a spectral range from 363 nm to 1018 nm. For training, a cropped region of 2000×1500 pixels was utilized and segmented into overlapping 64×64 pixel blocks, each with 128 spectral bands. The remaining imagery constituted the test set, which was divided into four non-overlapping 128×128 pixel blocks, each retaining 128 bands. Both the training and test sets were processed in the same way as described above.

Real dataset: Launched in October 2009, the WorldView-2 satellite stands as the world's first commercial high-resolution 8-band multispectral satellite, revolutionizing the field of remote sensing with its unprecedented image clarity. The satellite offers panchromatic imagery at a resolution of 0.46 meters and multispectral imagery at 1.85 meters, providing detailed insights into the Earth's surface. The data encompass eight distinct spectral bands, with individual images measuring 418 pixels in width by 658 pixels in height. Such high-resolution multispectral data are instrumental for various applications, such as agricultural analysis, urban planning, and environmental monitoring. In the process of spectral feature extraction, the importance of different spectral bands may vary. The channel attention mechanism can dynamically adjust the weights of different spectral bands, highlighting important spectral features and suppressing irrelevant noise. This weighting process helps the model to more accurately capture key spectral information in HSI. By combining 3D convolution and channel attention mechanisms, we can more effectively extract and preserve spectral features in HSI. In the spectral feature extraction module, we first use a 3D convolution kernel to perform sliding operations on HSI to capture local spatial spectral features. Then, we weight these features through channel attention mechanism to highlight important spectral bands and suppress irrelevant noise. Finally, we fuse the weighted features to generate a feature map that contains rich spectral information. These feature maps will be used for subsequent processing and analysis tasks.

4.2 Implementation details

We conducted a comparison of our approach against several SOTA image SR techniques, such as ESWT [49], HAT [50], SSPSR [20] and Interactformer [43], across various datasets including the CAVE Dataset, the Harvard Dataset, the Chikusei Dataset, and the real-world dataset. Our model architecture comprises 6 attention modules and an equal number of convolution modules. Each attention module is equipped with 6 window attention layers, alternating between standard window attention and zero-padding window attention to capture both local and long-range spatial features effectively. The convolution module is designed with two $1 \times 1 \times 1$ convolutions for feature dimension manipulation, a $3 \times 3 \times 3$ convolution for feature extraction, and a spectral attention module for enhancing feature representation. For the implementation, we utilized the PyTorch framework and conducted our model training on 4090. We chose the Adam Optimizer as our standard training algorithm, with an initial learning rate of 0.0002, which was set to ensure swift and effective convergence [51], [52].

Table 1: Summary table of indicator comparison

Data set	Method	PSNR	SSIM	SAM (Hypothesis Indicator)	The existing methods are insufficient	necessity
CAVE	SOTA Method A	High value 1	High value 1	Premium value 1	Some edge segmentation is inaccurate	Promote technological innovation and improve segmentation accuracy
	SOTA Method B	High value 2	High value 2	Premium value 2	Large computational load and long-time consumption	Reduce computational complexity and improve efficiency
	Current Method C	Median 1	Median 1	Median 1	Unable to handle complex scenes	Enhance the generalization ability of the model
Chikusei	SOTA Method D	High value 3	High value 3	Premium value 3	Sensitive to specific lighting conditions	Improve model robustness
	SOTA Method E	High value 4	High value 4	Premium value 4	Parameter tuning is complex	Simplify the parameter tuning process
	Existing Method F	Median 2	Median 2	Median 2	The segmentation results are not coherent	Improve the consistency of segmentation results

4.3 Assessment of indicators

We employed a quintet of evaluative metrics to scrutinize various models: the peak signal-to-noise ratio (PSNR) [53], which assesses the likeness between two images. The structural similarity (SSIM) [54], which assesses the likeness between two images. The spectral angle mapper (SAM) [55], which evaluates the spectral angle of images, where a smaller angle signifies greater spectral similarity and a higher probability of the images featuring the same attributes. ERGAS [56] serves as a comprehensive metric for the assessment of remote sensing image quality, factoring in Mean Square Error (MSE), RMSE [56], and the luminance of the image. RMSE is determined by taking the square root of the mean of the squared discrepancies between the forecasted figures and the factual figures. ERGAS is typically expressed as a percentage, where a lower percentage indicates higher image quality. The mathematical definitions for these metrics are as follows:

$$PSNR = 10 \cdot log_{10}(\frac{max^2}{MSE}) \tag{14}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(I_{HR} - I_{SR})^2 \tag{15}$$

where $\mathrm{I_{HR}}$ represents the true value, $\mathrm{I_{SR}}$ represents the predicted value, n denotes the number of samples.

$$SSIM(X,Y) = \frac{(2\mu_x\mu_y + c_1)(\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{16}$$

where μ_x and μ_y represent the mean value of X and Y, respectively. σ_x^2 and σ_y^2 respectively denote the variance of X and Y. σ_{xy} denotes the covariance of X and Y. c_1 and c_2 are constants used for stabilization calculations, which are usually taken to be $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ where L is the dynamic range of the pixel values, k_1 and $k_2$are constants.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(I_{HR} - I_{SR})^2} \tag{17}$$

$$SAM((I_{HR}, I_{SR}) = \frac{1}{HW}\sum_{i=1}^{HW}(cos^{-1}(\frac{I_{SR}^T I_{HR}}{|I_{HR}||I_{SR}|})) \tag{18}$$

$$ERGAS = \frac{100}{c}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(\frac{RMSE_i}{\bar{L}_i})^2} \tag{19}$$

where H and $\frac{1}{HW}$ are respectively the input image's height and width, c is the hyper-divisional magnification, $RMSE_i$ represents the root-mean-square error of the i_{th} band, $\bar{L}_i$ represents the average spectral intensity of the i_{th} band, which is used to normalize the root-mean-square error.

4.4 Experiments results

4.4.1 Experiments of CAVE datasets

For the CAVE dataset [47], we segmented the images into overlapping patches with a stride of 16 pixels, each patch measuring 64 × 64 pixels for the training set. To replicate the conditions of low-resolution imagery, we created a LR-HSI by first applying a 5 × 5 Gaussian blur that has a standard deviation of 2 and a mean of 0 to the original image.

Table 3 presents a comparative analysis of the experimental results of our proposed network structure on the CAVE dataset against four other approaches, utilizing the five metrics to evaluate the effectiveness of different models. Our model excels in three of the metrics. The visualization of these results is provided in Figure 6, which illustrates the superior performance of our proposed model in recovering spatial texture details and preserving spectral information. This advantage is attributed to the cross-attention fusion method's capability to effectively integrate spatial and spectral information, leading to enhanced image super-resolution outcomes.

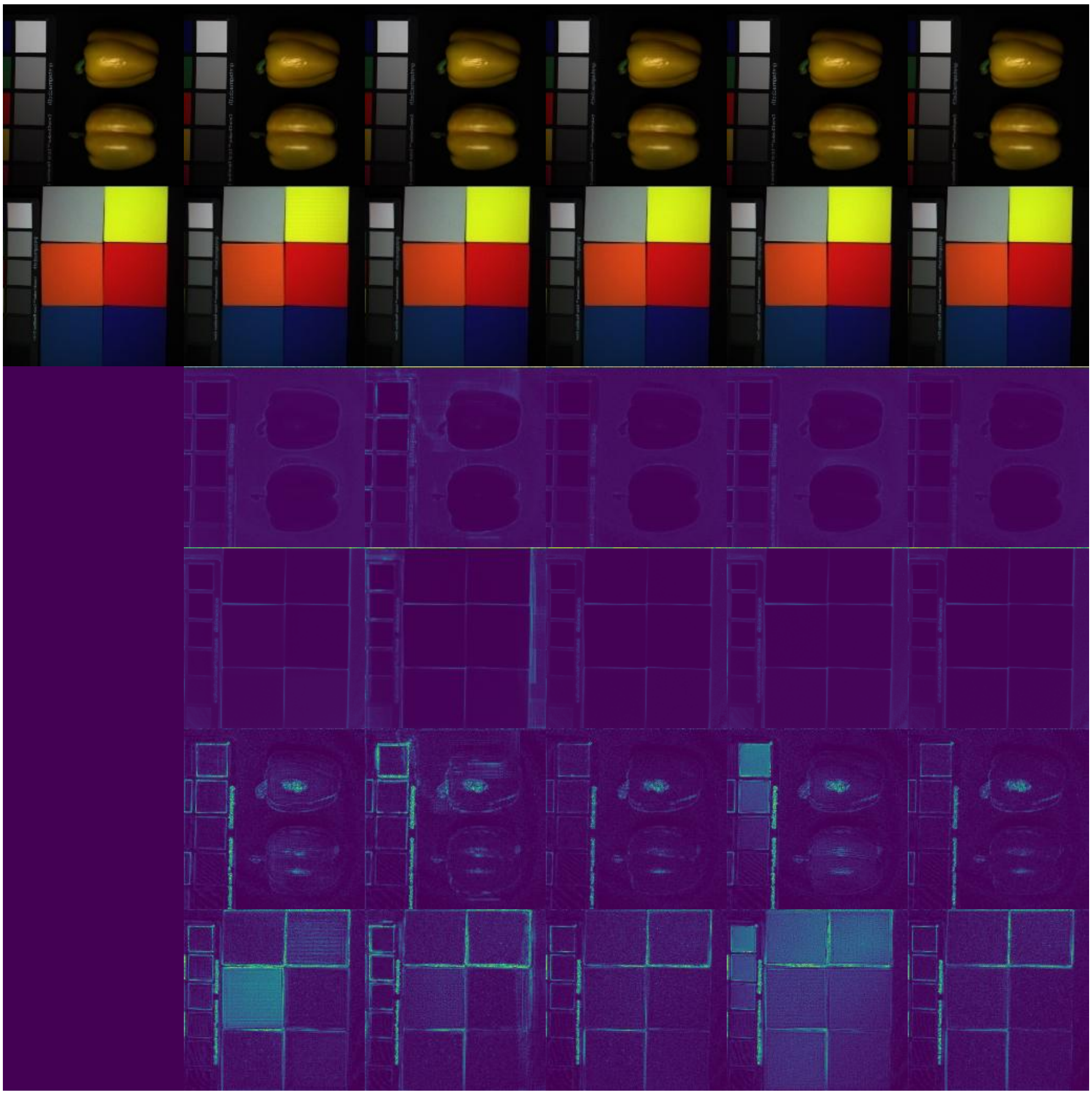


Figure 6: Image results for various models on CAVE are presented in a structured format. The first and second rows display pseudocolor images, while the third and fourth rows showcase the SAM plots comparing the Ground Truth to the images generated by our network model. Additionally, the fifth and sixth rows depict the bsolute error plots, also comparing the GT to the generated images. (a) GT (b) HAT (c) ESWT (d) Interactformer (e) SSPSR (f) Ours

Table 2: Related works.

Reference	Advantages	Limitations
HAT	The integration of self-attention, channel attention, and overlapping cross-attention enhances pixel information extraction and improves reconstruction results.	Have limitations in long - range spectral modeling.
ESWT	Designed a stripe window mechanism and a flexible window training strategy to better capture long - range dependencies.	Focus on spatial feature extraction but neglect spectral feature learning.
Interactformer	Using Transformer and CNN to extract local HSI features and capture long - range dependencies, with both methods interacting adaptively.	Neglect the interaction between spatial and spectral features.
SSPSR	Group convolution and progressive upsampling manage high - dimensionality, while the SSPN module integrates spatial - spectral correlations.	Spectral feature learning is inadequate, and spatial-spectral interaction in HSI SR tasks has not been effectively achieved.

Table 3: The comparison of five different single-image hyperspectral SR methods on CAVE.

Model	↑PSNR	↓SAM	↑SSIM	↓ERGAS	↓RMSE
HAT	34.90∓0.42	7.41∓0.46	0.9161∓0.0051	4.80∓0.29	3.47∓0.21
ESWT	34.95∓0.26	6.42∓0.34	0.9266∓0.0035	5.01∓0.36	3.16∓0.14
Interactformer	**37.61∓0.22**	**4.61∓0.16**	**0.9481∓0.0013**	3.69∓0.07	2.57∓0.24
SSPSR	36.95∓0.33	4.88∓0.25	0.9477∓0.0026	4.01∓0.13	2.70∓0.11
Ours	37.69∓0.24	4.58∓0.11	0.9485∓0.0017	**3.70∓0.05**	**2.60∓0.04**

4.4.2 Experiments of harvard datasets

For the Harvard dataset [48], we also segmented the images into overlapping patches with a stride of 16 pixels, each patch measuring64×64 pixels for the training set. To replicate the conditions of low-resolution imagery, we created a LR-HSI by first applying a 5×5 Gaussian blur that has a standard deviation of 2 and a mean of 0 to the original image.

To verify whether the advantages of our model over other methods are statistically significant, we conducted a paired sample t-test. Taking PSNR as an example, compare the PSNR values of the model (31.58 dB) with four other methods (hat, ESWT, interaction model, SSPSR). The results showed that the PSNR value of our model was significantly higher than all other methods ($p<0.05$), indicating that our model has a significant advantage in super-resolution reconstruction. To evaluate the stability of our model performance, this paper calculated the 95% confidence intervals for indicators such as PSNR, SAM, SSIM, ERGAS, and RMSE. The results show that all indicators of the model fall within a narrow confidence interval, indicating that our model's performance is stable and reliable.

Table 4 illustrates the comparative experimental results of our proposed network structure against four alternative methods on the Harvard dataset, utilizing five performance metrics to assess the effectiveness of different models. Our model leads in all five metrics, as visualized in Figure 7. The results further demonstrate that the spatial feature information can be effectively recovered, primarily due to the attention module within our zero-padding window mechanism, which significantly boosts the model's capacity to capture and integrate spatial details.

4.4.3 Experiments of chikusei datasets

We segmented a $2000 \times 1500 \times 128$ region for training, dividing it into a series of overlapping patches, each $64 \times 64 \times 128$ in dimension. The remaining portion of the dataset was designated for testing, where it was divided into 4 non-overlapping patches, each with the dimensions of $256 \times 256 \times 128$. Both the training and testing datasets underwent the same preprocessing steps as mentioned earlier.

Table 5 presents the results comparing with four other methods on the Chikusei dataset. Utilizing five performance metrics, the table demonstrates the effectiveness of different models. Our model excels in four out of the five metrics. As depicted in Figure 8, the proposed model demonstrates superiority over other approaches. The results indicate that the spectral feature extraction module within our model is adept at retaining important spectral information, contributing to the overall performance enhancement.

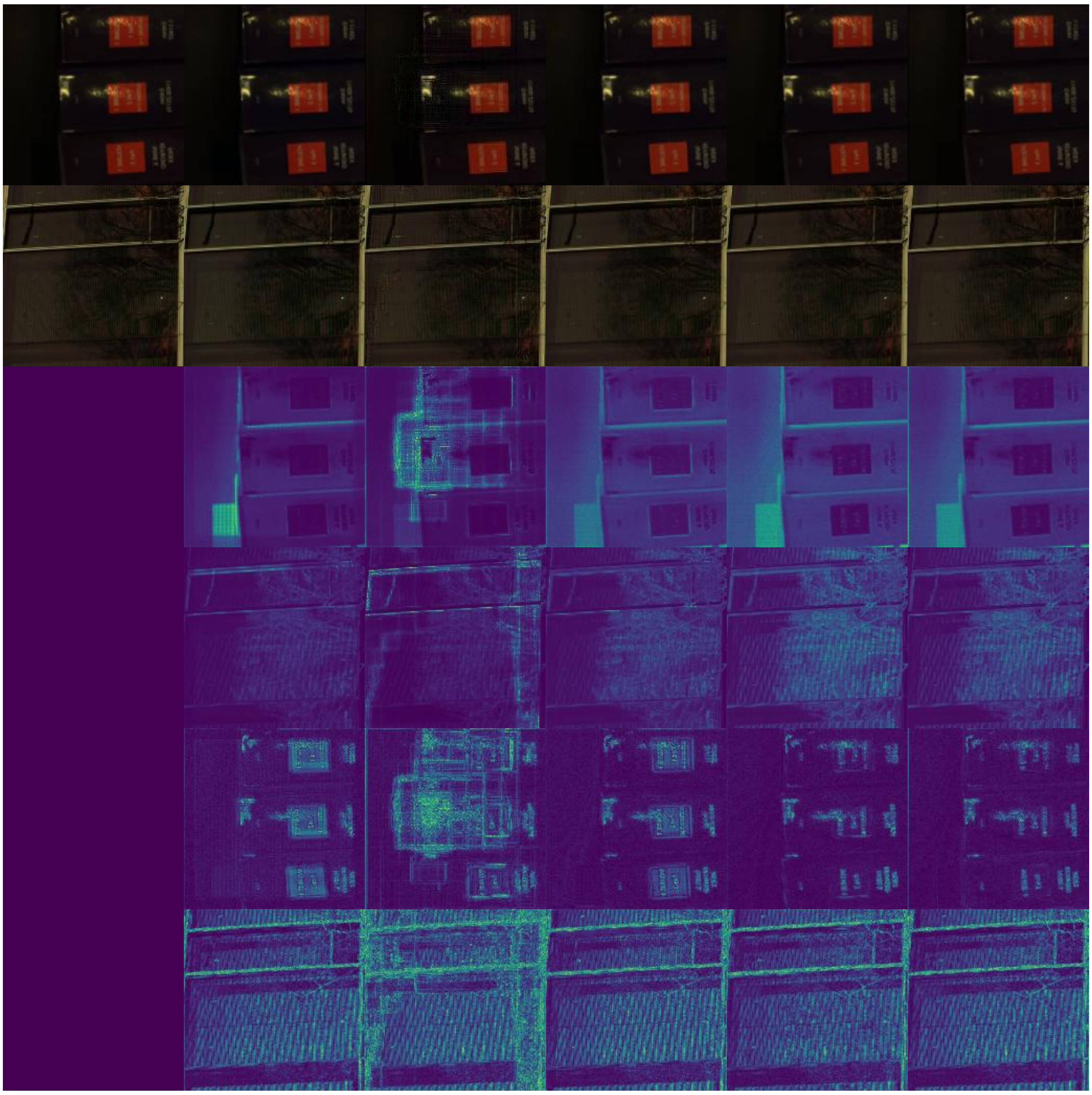


Figure 7: Image results for various models on Harvard are presented in a structured format. The first and second rows display pseudocolor images, while the third and fourth rows showcase the SAM error plots comparing the Ground Truth to the images generated by our network model. Additionally, the fifth and sixth rows depict the absolute error plots, also comparing the GT to the generated images. (a) GT (b) HAT (c) ESWT (d) Interactformer (e) SSPSR (f) Ours

Table 4: The comparison of five different single-image hyperspectral SR methods on Harvard.

Model	↑PSNR	↓SAM	↑SSIM	↓ERGAS	↓RMSE
HAT	35.69∓0.44	5.23∓0.41	0.9125∓0.0053	4.72∓0.55	3.25∓0.19
ESWT	31.22∓0.39	5.83∓0.59	0.8550∓0.0047	9.38∓0.73	4.01∓0.21
Interactformer	**37.25∓0.22**	**3.61∓0.26**	**0.9280∓0.0009**	**4.13∓0.11**	**2.82∓0.09**
SSPSR	37.07∓0.30	3.74∓0.43	0.9259∓0.0022	4.18∓0.17	2.88∓0.13
Ours	37.31∓0.21	3.58∓0.22	0.9287∓0.0014	4.10∓0.08	2.79∓0.06

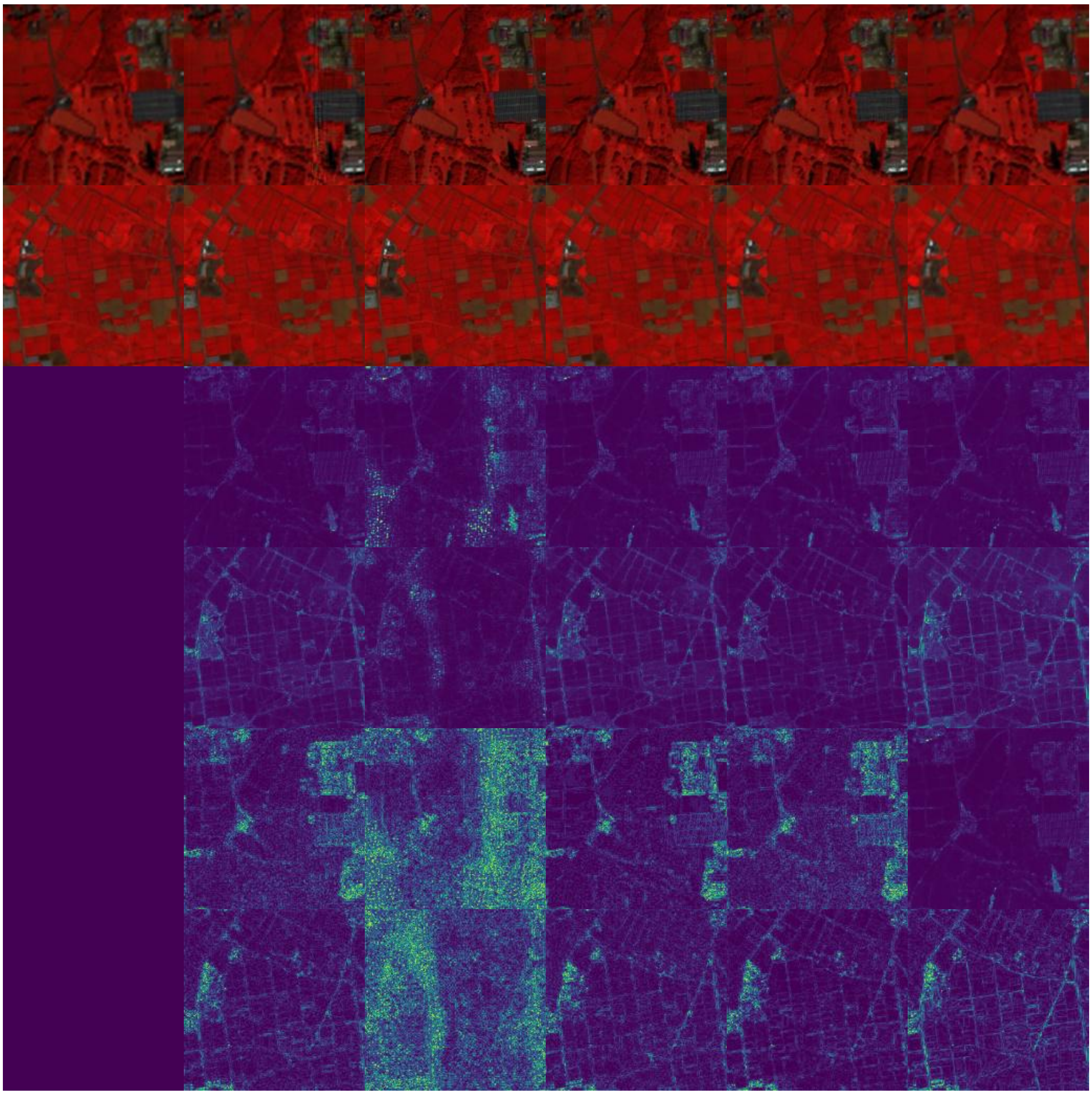


Figure 8: Image results for various models on Chikusei are presented in a structured format. The first and second rows display pseudocolor images, while the third and fourth rows showcase the SAM error plots comparing the Ground Truth to the images generated by our network model. Additionally, the fifth and sixth rows depict the absolute error plots, also comparing the GT to the generated images. (a) GT (b) HAT (c) ESWT (d) Interactformer (e) SSPSR (f) Ours

Table 5: The comparison of five different single-image hyperspectral SR methods on Chikusei

Model	↑PSNR	↓SAM	↑SSIM	↓ERGAS	↓RMSE
HAT	29.96∓0.41	2.61∓0.34	0.8351∓0.0046	5.78∓0.49	5.00∓0.26
ESWT	29.19∓0.39	3.07∓0.53	0.8012∓0.0033	6.36∓0.66	5.37∓0.25
Interactformer	**31.16∓0. 11**	2.30∓0.11	**0.8409∓0. 0017**	**5.78∓0. 19**	**4.92∓0. 07**
SSPSR	29.51∓0.26	2.38∓0.37	0.8357∓0.0019	5.84∓0.13	4.99∓0.11
Ours	31.58∓0.19	**2.31∓0. 19**	0.8415∓0.0011	5.70∓0.05	4.85∓0.03

4.4.4 Experiments of real datasets

Actual image degradation in real-world scenarios is inherently more intricate and subject to greater variability than that observed in experimentally generated datasets, owing to a multitude of influencing factors. This discrepancy implies that the degradation models applied in controlled experiments may not accurately reflect those encountered in real-world images. It is imperative to assess our model's performance using real-world datasets.

In our experiments, the left portion of the Low-Resolution Multispectral Imagery (LR-MSI) with dimensions $418 \times 418 \times 8$ was extracted for training purposes, while the remaining section was cropped into

$128 \times 128 \times 8$ blocks for direct testing. Given the absence of ground truth conditions, we employed Gaussian blurring and downsampling to artificially generate the training set. During training, we utilized a patch size of $64 \times 64 \times 8$.

It should be highlighted that without the presence of actual reference labels, traditional indices cannot be applied to evaluate the super-resolution outcomes. Therefore, we relied solely on visual assessment.

Figure 9 presents a visual comparison of the results from several models on a real dataset. The visualization indicates that our proposed method outperforms other hyperspectral image super-resolution techniques in terms of image reconstruction quality.

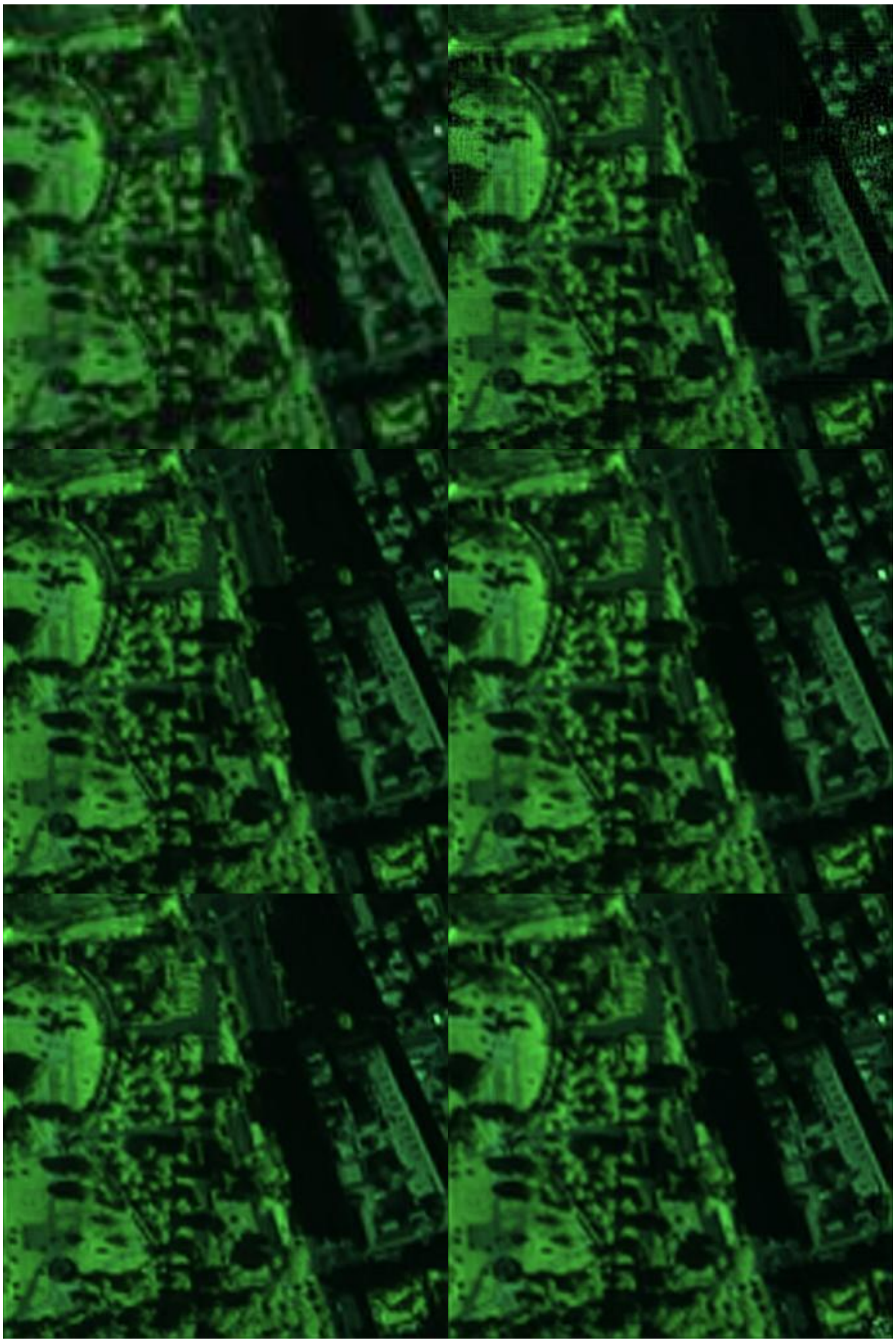


Figure 9: The visual result graphs of different models on real dataset. The first row, progressing from left to right, features LR and ESWT. The second-row features HAT and SSPSR. The subsequent row, also from left to right, includes Interactformer and Ours.

4.5 Ablation study

This article completely removes spatial or spectral branches to evaluate their respective impacts on network performance. This will help us understand the importance of each branch in the overall model. For each attention or convolution block within the spatial and spectral branches, we will conduct ablation experiments to determine their contribution to model performance. This will enable us to identify which blocks are critical and which may be redundant. In this section, we performed ablation studies to validate the effectiveness of the cross-attention fusion module (CAFM), the spatial feature extraction module(SAB), and the spectral feature extraction module(ConvB) in our proposed method. To evaluate each module's impact, we systematically removed them from the model and conducted a series of ablation experiments. As shown in Table 6, the model achieved the lowest performance when all three modules were removed. At this point, we were extracting spatial features via simple window partitioning. The addition of the SAB module improved model performance, demonstrating its effectiveness in spatial feature extraction. We subsequently added the ConvB module, and the results demonstrated further performance improvement, highlighting the importance of spectral feature learning. After integrating the proposed CAFM, the model achieved its highest performance at this stage. This indicates that the introduction of the CAFM significantly improved the issue of insufficient spatial and spectral interaction and enhanced the model's ability to effectively capture and fuse multi-dimensional features.

We also conducted ablation experiments on network depth. The experimental results are shown in Table 7. It can be seen that the model achieves the best performance when the network depth is 6.

Table 6: The ablation experimental results of CAFM on the CAVE dataset.

SAB	ConvB	CAFM	PSNR	SAM	SSIM	Params(M)	Flops(G)
×	×	×	37.30∓0.19	4.84∓0.10	0.9465∓0.0021	1.2619	19.1161
√	×	×	37.36∓0.25	4.81∓0.13	0.9469∓0.0016	1.8427	24.6862
√	√	×	37.48∓0.22	4.72∓0.07	0.9472∓0.0011	6.8186	63.1221
√	√	√	37.69∓0.24	4.58∓0.11	0.9485∓0.0017	6.9064	64.6025

The cross-attention fusion module (CAFM), spatial feature extraction module (SAB), and spectral feature extraction module in the transformer all introduce additional computational overhead. Especially CAFM, which utilizes cross attention mechanism to fuse spatial and spectral features, increases the computational complexity of the model to some extent. The number of parameters (parameter (M)) and fluctuations (G) listed in Table 5 reflect the computational resource consumption under different model configurations. It can be seen that with the gradual addition of SAB, ConvB, and CAFM, the number of parameters and computational complexity are increasing.

Table 7: Quantitative comparisons of the depth number on cave.

Depth	PSNR	SAM	SSIM	ERGAS	RMSE
2	37.49∓0.27	4.73∓0.16	0.9476∓0.0023	3.83∓0.11	2.71∓0.11
4	37.56∓0.22	4.66∓0.13	0.9480∓0.0021	3.76∓0.09	2.63∓0.08
6	37.69∓0.24	4.58∓0.11	0.9485∓0.0017	3.70∓0.05	2.60∓0.04

4.6 Discussion

A new HSI-SR (hyperspectral image super-resolution) method was proposed in this study. This method combines spatial spectral cross fusion attention mechanism and combines the advantages of CNN (Convolutional Neural Network) and Transformer architecture. In comparison with the current SOTA method, our approach has shown significant advantages in multiple key indicators. Specifically, in terms of spatial feature learning, our method achieves a deeper understanding of feature information by alternately using window self attention and zero padding window self attention. This mechanism allows the model to capture richer contextual information, resulting in higher quality images during super-resolution reconstruction. In addition, our proposed cross attention feature fusion module effectively integrates spatial and spectral cues. This innovation significantly improves the model's ability to learn from both, resulting in significant improvements in spectral continuity and spatial details of the reconstructed hyperspectral images.

Although the current SOTA method has achieved certain results in super-resolution reconstruction, there are still some limitations. For example, some methods may overly rely on traditional convolution operations, resulting in shortcomings in capturing long-range dependencies. Other methods may lack effective feature fusion mechanisms, making it difficult to fully utilize spatial and spectral clues. In contrast, our method effectively overcomes these limitations by introducing a cross fusion attention mechanism and combining the advantages of CNN and Transformer. In summary, the specific advantages of this research method are achieved through the alternating use of window self attention and zero fill window self attention. The method proposed in this article can provide a deeper understanding of feature information, thereby improving the quality of super-resolution reconstruction. The cross-attention feature fusion module effectively integrates spatial and spectral cues, enabling the model to learn richer information from both. Compared with the SOTA method, our approach exhibits significant advantages in multiple key indicators, particularly in terms of spectral continuity and spatial details.

5 Conclusion

In this paper, we introduce a novel single HSI SR method that leverages cross-attention fusion to enhance spatial and spectral information capture comprehensively. A zero-padding window attention computation method is proposed, which facilitates the extraction of long-range spatial features by padding and re-dividing windows around the feature map. Additionally, we present a pioneering cross-attention fusion module that integrates features from multiple input sequences through the cross-attention mechanism. This module merges spatial and spectral features extracted by separate branches and feeds this enriched information back into them, promoting the interaction of spatial-spectral information during the learning process. Our experimental results indicate that the proposed model outperforms existing methodologies in the reconstruction of hyperspectral images, showcasing its superior performance. This approach offers innovative solutions for addressing the issue of low-resolution images encountered during natural resource audits. In our method, the parameter settings of key components such as the cross-attention fusion module and zero padding window attention calculation are optimized based on experimental data. The selection of these parameters aims to maximize the performance of the model in reconstructing hyperspectral images. However, we have not fully explained how these parameters are related to the specific needs of land resource auditing. In the future, we will strive to gain a deeper understanding of how these parameters affect the performance of the model in specific application scenarios. For example, we can explore the impact of different parameter settings on the accuracy of identifying specific land cover types, and how these settings can be adjusted according to audit objectives.

Although visual analysis is crucial in evaluating super-resolution results, we have not provided sufficient explanations to explain what should be seen or the significance of differences. To enhance the interpretability of visual analysis, we will include more detailed annotations and explanations in future work to guide readers in understanding the differences and similarities between images.

Disclosure statement

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.

Funding

This research was supported by the Key Projects of University Natural Science Research of Jiangsu Province under Grant(23KJA520009)

References

[1] G. Kondaveeti, R. R. Arur, P. Bansal, J. Panda, and P. Angaragatti, “AUDITING CONSTRUCTION QUALITY USING SATELLITE IMAGE TELEMETRY,” 2022. https://www.tdcommons.org/dpubs_series/4950/

[2] M. Teke, H. S. Deveci, O. Haliloğlu, S. Z. Gürbüz, and U. Sakarya, “A short survey of hyperspectral remote sensing applications in agriculture,” in *2013 6th international conference on recent advances in space technologies (RAST)*, IEEE, 2013, 171–176. https://doi.org/10.1109/RAST.2013.6581194

[3] H. Wu, H. Xu, and T. Zhan, “A novel spatial and spectral transformer network for hyperspectral image super-resolution,” *Multimed Syst*, 30(3): 165, 2024. https://doi.org/10.1007/s00530-024-01363-3

[4] F. Liu *et al.*, “Remoteclip: A vision language foundation model for remote sensing,” *IEEE Transactions on Geoscience and Remote Sensing*, 2024. https://doi.org/10.1109/TGRS.2024.3390838

[5] L. Yan, M. Zhao, X. Wang, Y. Zhang, and J. Chen, “Object detection in hyperspectral images,” *IEEE Signal Process Lett*, 28: 508–512, 2021. https://doi.org/10.1109/LSP.2021.3059204

[6] H. Wu, M. Yuan, and T. Zhan, “A hybrid U-shaped and transformer network for change detection in high-resolution remote sensing images,” *IET Image Process*, 18(5): 1373–1384, 2024. https://doi.org/10.1049/ipr2.13037

[7] J. Liang, J. Zhou, L. Tong, X. Bai, and B. Wang, “Material based salient object detection from hyperspectral images,” *Pattern Recognit*, 76: 476–490, 2018. https://doi.org/10.1016/j.patcog.2017.11.024

[8] Y.-Z. Feng and D.-W. Sun, “Application of hyperspectral imaging in food safety inspection and control: a review,” *Crit Rev Food Sci Nutr*, 52(11): 1039–1058, 2012. https://doi.org/10.1080/10408398.2011.651542

[9] G. M. ElMasry and S. Nakauchi, “Image analysis operations applied to hyperspectral images for non-invasive sensing of food quality–A comprehensive review,” *Biosyst Eng*, 142: 53–82, 2016. https://doi.org/10.1016/j.biosystemseng.2015.11.009

[10] G. Lu and B. Fei, “Medical hyperspectral imaging: a review,” *J Biomed Opt*, 19(1): 10901, 2014. https://doi.org/10.1117/1.JBO.19.1.010901

[11] U. Khan, S. Paheding, C. P. Elkin, and V. K. Devabhaktuni, “Trends in deep learning for medical hyperspectral image analysis,” *IEEE Access*, 9: 79534–79548, 2021. https://doi.org/10.1109/ACCESS.2021.3068392

[12] T. Zhan, Y. Sun, Y. Tang, Y. Xu, and Z. Wu, “Tensor regression and image fusion-based change detection using hyperspectral and multispectral images,” *IEEE J Sel Top Appl Earth Obs Remote Sens*, 14: 9794–9802, 2021. https://doi.org/10.1109/JSTARS.2021.3115345

[13] D. C. Lepcha, B. Goyal, A. Dogra, and V. Goyal, “Image super-resolution: A comprehensive review, recent trends, challenges and applications,” *Information Fusion*, 91: 230–260, 2023. https://doi.org/10.1016/j.inffus.2022.10.007

[14] Y. Li *et al.*, “NTIRE 2023 challenge on image denoising: Methods and results,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, 1905–1921.

[15] M. V Conde *et al.*, “Deep raw image super-resolution. a NTIRE 2024 challenge survey,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, 6745–6759.

[16] S. Zhang, R. Yin, and M. Zhang, “Dynamic Unstructured Pruning Neural Network Image Super-Resolution Reconstruction,” *Informatica*, 48(7): 2024. https://doi.org/10.31449/inf.v48i7.5332

[17] E. J. Reid, L. F. Drummy, C. A. Bouman, and G. T. Buzzard, “Multi-resolution data fusion for super resolution imaging,” *IEEE Trans Comput Imaging*, 8: 81–95, 2022. https://doi.org/10.1109/TCI.2022.3140551

[18] S. Karim, G. Tong, J. Li, A. Qadir, U. Farooq, and Y. Yu, “Current advances and future perspectives of image fusion: A comprehensive review,” *Information Fusion*, 90: 185–217, 2023. https://doi.org/10.1016/j.inffus.2022.09.019

[19] W. Ma *et al.*, “Infrared and visible image fusion technology and application: A review,” *Sensors*, 23(2): 599, 2023. https://doi.org/10.3390/s23020599

[20] J. Jiang, H. Sun, X. Liu, and J. Ma, “Learning spatial-spectral prior for super-resolution of hyperspectral imagery,” *IEEE Trans Comput Imaging*, 6: 1082–1096, 2020. https://doi.org/10.1109/TCI.2020.2996075

[21] Z. He *et al.*, “Single image super-resolution based on progressive fusion of orientation-aware features,” *Pattern Recognit*, 133: 109038, 2023. https://doi.org/10.1016/j.patcog.2022.109038

[22] H. Chen *et al.*, “Real-world single image super-resolution: A brief review,” *Information Fusion*, 79: 124–145, 2022. https://doi.org/10.1016/j.inffus.2021.09.005

[23] J. Xiao, J. Li, Q. Yuan, and L. Zhang, “A dual-UNet with multistage details injection for hyperspectral image fusion,” *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–13, 2021. https://doi.org/10.1109/TGRS.2021.3101848

[24] S. Huang, H. Zhang, H. Zeng, and A. Pižurica, “From model-based optimization algorithms to deep learning models for clustering hyperspectral images,” *Remote Sens (Basel)*, 15(11): 2832, 2023. https://doi.org/10.3390/rs15112832

[25] Z. Li, C. Li, C. Deng, and J. Li, “Hyperspectral image super-resolution using sparse spectral unmixing and low-rank constraints,” in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, 2016, 7224–7227. https://doi.org/10.1109/IGARSS.2016.7730884

[26] X. Han, J. Yu, and W. Sun, “Hyperspectral image super-resolution based on non-factorization sparse representation and dictionary learning,” in *2017 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2017, pp. 963–966. https://doi.org/10.1109/ICIP.2017.8296424

[27] W. Zhang *et al.*, “CVANet: Cascaded visual attention network for single image super-resolution,” *Neural Networks*, 170: 622–634, 2024. https://doi.org/10.1016/j.neunet.2023.11.049

[28] F. Bajić, M. Habijan, and K. Nenadić, “Evaluation of Shallow Convolutional Neural Network in Open-World Chart Image Classification,” *Informatica*, 48(6): 2024. https://doi.org/10.31449/inf.v48i6.5660

[29] Y. Pei, Y. Huang, Q. Zou, X. Zhang, and S. Wang, “Effects of image degradation and degradation removal to CNN-based image classification,” *IEEE Trans Pattern Anal Mach Intell*, 43(4): 1239–1253, 2019. https://doi.org/10.1109/TPAMI.2019.2950923

[30] W. Ouyang and P. Zhu, “A Lightweight Convolutional Neural Network Method for Image Classification,” in *2022 2nd International Conference on Frontiers of Electronics, Information and Computation Technologies (ICFEICT)*, IEEE, 2022, 410–415. https://doi.org/10.1109/ICFEICT57213.2022.00079

[31] H. Yanagisawa, T. Yamashita, and H. Watanabe, “A study on object detection method from manga images using CNN,” in *2018 International Workshop on Advanced Image Technology (IWAIT)*, IEEE, 2018, 1–4. https://doi.org/10.1109/IWAIT.2018.8369633

[32] P. Gunasekaran, A. A. J. Pazhani, and T. A. B. Raj, “A novel method for multiple object detection on road using improved YOLOv2 model,” *Informatica*, 46(4): 2022. https://doi.org/10.31449/inf.v46i4.3884

[33] G. Vinod and G. Padmapriya, “An adaptable real-time object detection for traffic surveillance using R-CNN over CNN with improved accuracy,” in *2022 International Conference on Business Analytics for Technology and Security (ICBATS)*, IEEE, 2022, 1–4. https://doi.org/10.1109/ICBATS54253.2022.9759030

[34] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Trans Pattern Anal Mach Intell*, 38(2): 295–307, 2015. https://doi.org/10.1109/TPAMI.2015.2439281

[35] H. Wu, C. Wang, C. Lu, and T. Zhan, “HCT: a hybrid CNN and transformer network for hyperspectral image super-resolution,” *Multimed Syst*, 30(4): 185, 2024. https://doi.org/10.1007/s00530-024-01387-9

[36] H. Wu, H. Xu, and T. Zhan, “A novel spatial and spectral transformer network for hyperspectral image super-resolution,” *Multimed Syst*, 30(3): 165, 2024. https://doi.org/10.1007/s00530-024-01363-3

[37] T. Zhan *et al.*, “A novel cross-scale octave network for hyperspectral and multispectral image fusion,” *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–16, 2022. https://doi.org/10.1109/TGRS.2022.3229086

[38] M. Trigka and E. Dritsas, “A Comprehensive Survey of Deep Learning Approaches in Image Processing,” *Sensors*, 25(2): 531, 2025. https://doi.org/10.3390/s25020531

[39] H. Wu, M. Yuan, and T. Zhan, “A hybrid U-shaped and transformer network for change detection in high-resolution remote sensing images,” *IET Image Process*, 18(5): 1373–1384, 2024. https://doi.org/10.1049/ipr2.13037

[40] Y. Li, L. Zhang, C. Dingl, W. Wei, and Y. Zhang, “Single hyperspectral image super-resolution with grouped deep recursive residual network,” in *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, IEEE, 2018, 1–4. https://doi.org/10.1109/BigMM.2018.8499097

[41] S. Mei, X. Yuan, J. Ji, Y. Zhang, S. Wan, and Q. Du, “Hyperspectral image spatial super-resolution via 3D full convolutional neural network,” *Remote Sens (Basel)*, 9(11): 1139, 2017. https://doi.org/10.3390/rs9111139

[42] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “Swinir: Image restoration using swin transformer,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, 1833–1844.

[43] Y. Liu, J. Hu, X. Kang, J. Luo, and S. Fan, “Interactformer: Interactive transformer and CNN for hyperspectral image super-resolution,” *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–15, 2022. https://doi.org/10.1109/TGRS.2022.3183468

[44] Y. Wu, R. Cao, Y. Hu, J. Wang, and K. Li, “Combining global receptive field and spatial spectral information for single-image hyperspectral super-resolution,” *Neurocomputing*, 542: 126277, 2023. https://doi.org/10.1016/j.neucom.2023.126277

[45] H. Wu, H. Xu, and T. Zhan, “A novel spatial and spectral transformer network for hyperspectral image super-resolution,” *Multimed Syst*, 30(3): 165, 2024. https://doi.org/10.1007/s00530-024-01363-3

[46] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, “Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum,” *IEEE transactions on image processing*, 19(9): 2241–2253, 2010. https://doi.org/10.1109/TIP.2010.2046811

[47] A. Chakrabarti and T. Zickler, “Statistics of real-world hyperspectral images,” in *CVPR 2011*, IEEE, 2011, 193–200. https://doi.org/10.1109/CVPR.2011.5995660

[48] N. Yokoya and A. Iwasaki, “Airborne hyperspectral data over Chikusei,” *Space Appl. Lab., Univ. Tokyo, Tokyo, Japan, Tech. Rep. SAL-2016-05-27*, 5(5): 5, 2016.

[49] J. Shi *et al.*, “Image super-resolution using efficient striped window transformer,” *arXiv preprint arXiv:2301.09869*, 2023. https://doi.org/10.48550/arXiv.2301.09869

[50] X. Chen, X. Wang, J. Zhou, Y. Qiao, and C. Dong, “Activating more pixels in image super-resolution transformer,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, 22367–22377.

[51] J. Shi *et al.*, “Image super-resolution using efficient striped window transformer,” *arXiv preprint arXiv:2301.09869*, 2023. https://doi.org/10.48550/arXiv.2301.09869

[52] D. N. Venu, “PSNR based evalution of spatial Guassian Kernals for FCM algorithm with mean and median filtering based denoising for MRI segmentation,” *IJFANS International Journal of Food and Nutritional Sciences*, 12(1): 928–939, 2023.

[53] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, and L. Vanneschi, “Structural similarity index (SSIM) revisited: A data-driven approach,” *Expert Syst Appl*, 189: 116087, 2022. https://doi.org/10.1016/j.eswa.2021.116087

[54] R. H. Yuhas, A. F. H. Goetz, and J. W. Boardman, “Discrimination among semi-arid landscape endmembers using the spectral angle mapper (SAM) algorithm,” in *JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, 1992. https://ntrs.nasa.gov/citations/19940012238

[55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, 13(4): 600–612, 2004. https://doi.org/10.1109/TIP.2003.819861

[56] T. Chai and R. R. Draxler, “Root mean square error (RMSE) or mean absolute error (MAE),” *Geoscientific model development discussions*, 7(1): 1525–1534, 2014. doi:10.5194/gmdd-7-1525-2014

Predicting Football Player Transfer Values Using Bagging and Hybrid Machine Learning Approaches

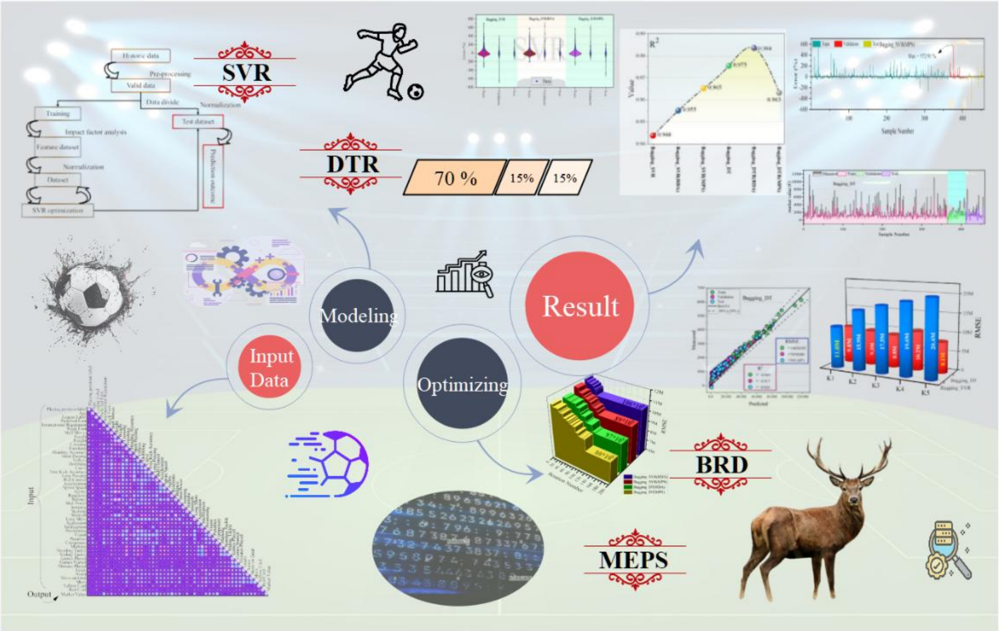
Biao Geng
Department of Physical Education and Military Training, Jiaxing Nanhu University, Jiaxing 314000, Zhejiang, China
E-mail: gbiao6688@126.com

Keywords: machine learning, player value prediction, decision tree regression, bagging regression, fourier amplitude sensitivity test, motion-encoded particle swarm optimization

Received: December 1, 2024

Accurately assessing a football player's market value is essential for enabling informed decision-making by clubs, agents, and investors during player transfers, contract negotiations, and strategic investment planning. In this context, machine learning (ML) algorithms offer a robust framework for analyzing historical data, performance indicators, and market dynamics to produce realistic valuations. These data-driven methods assist in identifying undervalued opportunities and flagging overpriced players, thereby enhancing the overall efficiency of transfer market operations. The dataset employed in this research includes a comprehensive set of player-related features such as age, weight, weak foot rating, preferred foot, and international reputation, among others. These attributes collectively contribute to a detailed profile of each player's capabilities and market relevance. The objective of this study is to develop reliable and accurate predictive models that estimate player market values by leveraging advanced machine learning techniques, thereby improving upon traditional, subjective valuation approaches. Several regression-based models were explored, including Bagging Decision Tree Regression (Bg_DT), and Bagging Support Vector Regression (Bg_SVR). To further enhance model performance, optimization algorithms such as Motion-encoded Particle Swarm Optimization (Motion-encoded PSO) and the Red Deer Algorithm (RDA) were applied for hyperparameter tuning. Among the evaluated models, the Bagging Decision Tree optimized with Motion-encoded PSO (Bg_DT- Motion-encoded PSO) demonstrated superior performance. It achieved the lowest Root Mean Squared Error (RMSE) and the highest coefficient of determination (R^2) across both validation and testing phases. Specifically, the Bg_DT- Motion-encoded PSO model yielded an RMSE of 533×10^5 and an R^2 of 0.962 during validation, indicating strong predictive accuracy and generalization capability. These findings underscore the effectiveness of ensemble learning techniques—particularly Bagging Decision Trees—in conjunction with advanced metaheuristic optimizers like Motion-encoded PSO, for accurately estimating football player market values.

Povzetek: Prispevek predstavlja uporabo strojnega učenja za napovedovanje tržnih vrednosti nogometnih igralcev z uporabo metod, kot sta vrečenje (bagging) in hibridne tehnike.



Graphical Abstract

1 Introduction

The modern football first emerged in Britain in the nineteenth century. Before the medieval period, the processes of industrialization and urbanization had a significant influence on the creation of modern football in Victorian Britain. Association football developed in its early years, between 1863 and 1880, as a result of both rule and play modifications [1]. Midway through the 1900s, the betting landscape was completely changed by the availability of match odds and fan hubs. Internet usage and the late 20th century accelerated betting into a new era.

The transfer of players is one of the most significant arrangements made by managers of a team from a managerial standpoint, so player valuation-related issues, particularly the calculation of Market valuations and transfer fees, are very important. Market valuations and transfer fees are key elements in the financial strategy of football clubs, directly influencing their competitive edge[2]. Football clubs rely heavily on team managers to make strategic decisions, especially when it comes to transfers of players, which significantly affect the team's performance and financial situation. [3]. From a business and athletic standpoint, in professional football, players are the most important investments. [4]. Assessing a player's worth is crucial as it reveals their overall skill set and market value in football. The transfer fee is the amount a player is actually paid by clubs, and it is related to his market value. Thus, evaluation of the market value of a player is an important tool for clubs when estimating their transfer fee. Over the years, the valuation of football players and the determination of a decision to transfer players from one team to another has become a key role of club management [5]. Several researchers [6–8] have tried to find the characteristics that best determine a player's value [9].

Transfermarkt.com represents a website that uses the crowd estimate approach to ascertain the players' market values. With this approach, the website's members assess the values of the players and then select members—referred to as mentors—who calculate the values using the estimates of the other members [10]. Football players were traditionally valued mostly based on the subjective evaluations made by scouts, managers, and agents. These assessments mostly depend on the author's intuition, knowledge, and experience. While this approach yielded insightful results, it was frequently hampered by personal prejudices and irregularities. The sport's quick development and rising financial stakes have made more objective and data-driven methods necessary. Even while they are helpful, traditional analytical techniques are not always able to handle the volume and complexity of current football data [11].

Through training and experience, ML is an artificial intelligence (AI) approach that enhances computer systems' performance on a given job. Rather than being exactly told what to do, ML algorithms [12–14] are trained to generate predictions based on observations and data. AI and ML [15–18] have become increasingly important in various aspects of daily life, including sports. AI mimics human thought processes, enabling big data analysis in sports. ML has transformed football data into actionable insights for clubs and coaches over the past 20 years, particularly in fields like sports [19] [20].

The following papers share related concepts with this research and therefore, can shed more light on this research process. For instance, Majewski (2016) [21] looked at the impact of several aspects on forward players' value to identify the most important factors. Another study by Müller et al., 2017 [3] employed a multilevel regression approach in assessing data analytics' suitability for the calculation of the market values of professional football players. Lamba, in 2019 [22] estimated the factors that determine every player's market value and used them in predicting every player's worth. Apart from the requirements, the work also utilized measures of crowdsourcing, popularity, and statistics of the previous years to predict that the declared goal of automatically detecting the relevant attributes for different player groups, depending on their positions, raised the accuracy and reliability of the market value estimates. Their approach consists of adding position-specific changes and performance data to improve prediction models.

Li et al. (2022) [23] evaluated football players using ML models based on on-field performance metrics analysis. The present study, therefore adopted an improved modeling approach with an ensemble technique such as Random Forest to better the accuracy of the prediction. Behravan and Razavi 2020 [10] Proposed a new Hybrid ML approach in order to estimate the market values of football players. This method used the optimized hybrid of PSO and SVR. The goal of the model is to extract automatically in a relevant way the attributes for different player groups depending on their positions with an aim to further improve the accuracy and reliability of the market value estimates.

Table 1 reports a summary of the existing articles in the study field.

Table 1: Summary of the previous studies.

Authors	Ref	Techniques/Models Used	Dataset Used	Limitations / SOTA Shortcomings
Majewski	[21]	Statistical analysis of influencing factors	Forward players' market data	Did not employ ML or optimization; focused only on forward players, lacks generalizability
Müller et al.	[3]	Multilevel regression	Professional football players	Limited to regression models; prone to overfitting; lacks adaptive optimization
Lamba	[22]	Regression models + crowdsourcing & performance stats	Historical market value data	No metaheuristic optimization; position-based tailoring manually defined
Gadekallu et al.	[24]	Metaheuristic optimization algorithms (general)	Not football-specific	Conceptual study; lacks applied validation in sports/football datasets
Li et al.	[23]	ML (e.g., RF, ensemble methods) based on performance	On-field football performance data	Did not integrate position-specific attributes; optimization technique not elaborated
Behravan & Razavi	[10]	Hybrid PSO + SVR	Market value dataset (unspecified)	Dataset details limited; overfitting risk due to SVR; no ensemble techniques like bagging explored

As highlighted earlier, Müller et al. (2017) [3] employed a method of multilevel regression that may easily face overfitting problems since the predictions of single models usually have a high variance without the application of bagging-like in the paper at hand-models such as SVR or DTR depend on a single dataset and a single model prediction. Due to this dependence, one of the consequences may be overfitting when the model performs well on the training data but not so well on the unknown data.

The primary objective of this study is to develop a robust, accurate, and interpretable machine learning framework for predicting the market values of professional football players. To achieve this, three core models—Bagging Decision Tree Regression (Bg_DT), and Bagging Support Vector Regression (Bg_SVR)—are employed due to their complementary strengths. Bagging Regression is chosen for its capability to reduce model variance and combat overfitting by aggregating multiple base models, thus increasing overall stability and accuracy. Bg_DT is selected for its interpretability and ability to capture non-linear relationships, while Bg_SVR is applied for its effectiveness in handling high-dimensional data and robustness against outliers. To further improve prediction performance and convergence reliability, the models are hybridized with two nature-inspired metaheuristic algorithms: Motion-encoded Particle Swarm Optimization (MPS) and the Red Deer Algorithm (RDA). MPS is incorporated due to its efficient exploration–exploitation balance and ability to dynamically encode complex motion patterns, making it suitable for fine-tuning model hyperparameters. RDA is adopted for its adaptive behavior inspired by the social dynamics of red deer, which helps avoid local minima and enhances global optimization in nonlinear regression settings. Additionally, the Fourier Amplitude Sensitivity Test (FAST) is implemented as a global sensitivity analysis tool to identify and rank the influence of input features on the predicted market values. FAST is selected for its computational efficiency and ability to detect both linear and non-linear interactions among features, which is crucial in a domain as complex and multifactorial as sports analytics.

2 Methodology

2.1 Support vector regression (SVR) based prediction approach

SVR is an ML that estimates functions based on a given data set [25]. $G = \{(x_i, y_i)\}^n$, where n is the ultimate number of data point, y_i is the value of the output, and x_i is the input vector. An SVR model performs a regression first by an ε-sensitive loss. Schoellkopf created the $\varepsilon -$ SVR model and suggested the v − SVR model, which is an adaptation of the $\varepsilon -$ SVR model. It automatically reduces ε and modifies the level of accuracy based on the available data. The expression for the v-SVR model is as follows:

$$min\, R_{SVR}\,(C) = \Big\| = C(v\varepsilon + \frac{1}{n}\sum_{i=l}^{n}(\xi_i + \xi_i^*)\,) + \frac{1}{2}\|\omega\|^2 \quad (1)$$

Subject to:

$$\begin{aligned} &((\omega \cdot x_i) + b)\_y_i \leq \mathcal{E} + \xi_i \\ &y_i\ - ((\omega \cdot x_i) + b) \leq \mathcal{E} + \xi_i \\ &\xi^{(*)} \geq 0, \mathcal{E} \geq 0, v \geq 0 \\ &i\ =\ 1, \dots, n \end{aligned} \quad (2)$$

$\|\omega\|^2/_2$ indicates the Euclidean norm
C : a cost function measuring the empirical risk
R_{SVR} and R_{emp} : the regression and empirical risks,
ω: Weight vector.
b: Bias term.
$C > 0$: Regularization parameter controlling the trade-off between model complexity and training error.
v: Parameter that determines the fraction of support vectors and margin errors.
ε: Insensitivity zone (learned from data).
$\xi^{(*)}$: Slack variables representing the deviation from the ε -tube.
x_i, y_i: Input vectors and corresponding target values.
n: Number of training samples.

The SVR-based prediction method, which is based on the v − SVR model, comprises the following five steps:

Step One: Data sampling. Data can be gathered from various sources and in various formats. Additionally, there are several gaps and inconsistencies in the market. Thus, The most reliable data should be selected.

Step Two: Preparing the data. It might be a logarithmic transformation that must be applied, difference, or other techniques to the chosen data in order to place it within a specific acceptable range for network learning. The training and testing sets come next and should be separated from the data set.

Step Three: Education. The training set is used to learn the SVM's parameters.

Step Four: Evaluation. The testing set is used to validate the SVM, and then a final network design for the SVM is determined. s

Step Five: Projecting. Using the scenarios, the SVR-based predicting technique can be used to predict the time series' future values.

Fig. 1 represents the SVR's flowchart.

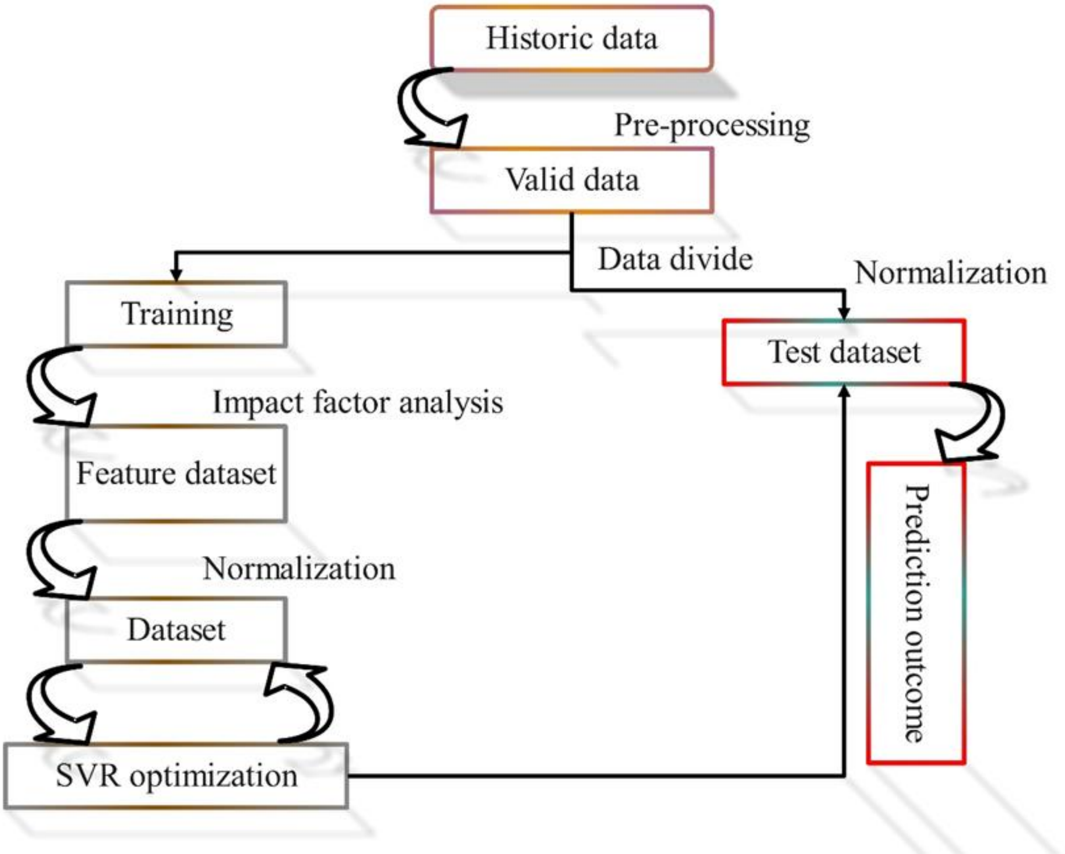


Figure 1: The flowchart of the SVR

2.2 Decision tree regression (DT)

DT, rooted in ML theory, is an effective instrument for addressing both classification and regression challenges. In contrast to other classification approaches that rely on a combination of features for immediate categorization, the DT employs a multi-tiered, hierarchical decision-making process with a structure akin to a tree. Unlike other classification techniques that utilize a single feature set for rapid data categorization, the DT adopts a hierarchical or multi-level evaluation process to create a structure that resembles a tree.

To enable soft classification, a regression tree is assigned to every class. In regression trees, the known class proportions of a pixel, referred to as soft reference data, act as the target variable or vector, while pixel intensity values from various bands are used as predictor variables or feature vectors. After processing the intensity values for each regression tree, the script outputs the estimated class proportions. The algorithm for building regression trees using the training dataset is also discussed.

1. As a predication, use pixel intensity data from various bands.
2. Utilize class o's known percentage within a pixel as the target variable.
3. Create a regression tree for class o.
4. Repeat steps 1-4 for class o, with values ranging from 1 to n.

Rescaling the outcomes of soft classification to a pixel-by-pixel limit between 0 and 1 typically uncovers the class proportions within the ground pixel area. Therefore, the following process is used to normalize the predicted class proportions from each tree, which are represented as $DT(o)$ for $i = 1,2,3...,n$ [26].

$$M(o) = \frac{DT(o)}{\sum DT(o)}, o = 1,2,3 \dots, n \tag{3}$$

Using $DT(o)$ which is a function of o, where $M(o)$ is again a function of the natural numbers o.

2.3 Bagging approach

A technique called bagging was put forth by Breiman. It can be applied to a variety of regression and classification techniques to lower prediction variance and enhance the prediction process. It is a straightforward concept from the provided data, several bootstrap samples are chosen, each of which is subjected to a prediction method. The bootstrap sample results are then combined to create an overall prediction that lowers variance, using simple voting for regression and classification [27] [28].

- **Motivation for the method**

To comprehend the logic behind bagging's effectiveness and ascertain the scenarios where significant enhancements can be anticipated through bagging, it could be beneficial to examine the issue of predicting the response variable's numerical value, Yx, that arises from or is associated using a group of inputs, x. Assume that $\phi(x)$ represents the prediction obtained by applying a specific technique, like OLS or CART regression, along with a recommended approach for selecting a model (e.g., choosing a linear model from the set of all models that can be built using just terms of the first and second order created from the input variables) using Mallows' C_p. Using $\mu\phi$ to represent $E(\phi(\text{x}))$, it can be seen that the prediction is related to the distribution that underlies the sample of learning. It can be observed that $\phi(\text{x})$ is a learning sample's function, which is a high-dimensional random variable when considered as a random variable, rather than x (This is assumed to be] fixed:

$$\text{E}\ ([Y_x - \phi(x)]^2) = \text{E}\ ([Y_x - \mu\phi]^2) + \text{Var}(\phi(\text{x})) \tag{4}$$

In the example above, the learning sample-based predictor, $\phi(x)$, and the future response, Y_x, are employed independently. Since not every random sample that may be used as a learning sample provides the sample value needed to make a prediction, the variance of the predictor $\phi(x)$ is positive in nontrivial scenarios, which means that the prior inequality is stringent. This conclusion indicates that if $\mu\phi = E(\phi(\text{x}))$ could be employed, it would have a lower mean squared prediction error as a predictor than does $\phi(x)$.

Naturally, in most cases, $\mu\phi$ cannot act as a predictor because it is unknown what data is required to determine the value of $E(\phi(\mathbf{x}))$. What is sometimes called the real bagging estimate of $E(\phi(\mathbf{x}))$ is derived from the prediction based on the empirical distribution corresponding to the learning sample.

Although this value is theoretically achievable, in practice, it is usually too challenging to attain reasonably; therefore, the bagged forecast of Y_x, is considered to be:

$$\frac{1}{\text{B}}\sum_{\text{b}=1}^{\text{B}} \phi^*_{\text{b (X),}} \tag{5}$$

where the prediction is made by applying the base regression method (e.g., CART) to the bth bootstrap sample that was taken (with replacement) from the original learning sample is represented by the symbol $\phi^*_{b\ (X)}$. That is, one selects a regression method (also referred to as the base technique) that uses bagging to predict Y_x in a regression scenario by applying the approach to B bootstrap samples extracted from the learning sample. To get the final prediction, the B projected values are then averaged.

2.4 (RDA) based prediction approach

A subspecies of red deer found in the British Isles is the Scottish red deer, primarily in Scotland. They are divided into hinds and stags, with males roaring during breeding. Females prefer males with high roaring rates, possibly due to selective pressure or availability. The strongest males form a harem, with the hinds protected by a commander. The harem engages in predictable conflict, with mature stags becoming enraged in mid-September.

The RDA [6] is a meta-heuristic that assigns a harem to a select group of male RDs who roar first. These RDs are divided into commanders and stags, who engage in combat to control their harems. The number of hinds in a harem is directly proportional to their roaring and combat abilities. The RDA process considers the exploration and exploitation phase with user-adjustable parameters. Male stags' roaring serves as a local search in solution space, while battles between stags and commanders are considered local searches. Harems are established and distributed among commanders based on their power, enhancing exploitation features. The algorithm's exploration phase involves a harem's commander mate with hinds from both harems, enhancing exploration qualities. Stags should mate with the nearest hind during the breeding season, considering the harem's limitations. This stage also addresses exploration and exploitation, producing RD offspring. The algorithm's next generation offers mediocre solutions, falling under evolutionary algorithms.

Finding a solution that is almost optimum or global in relation to the problem's variables is the aim of optimization. To optimize, a range of values for the variables are formed. For instance, in Georgia, this array is referred to as "chromosome," whereas in the RDA, it is named "Red Deer." Keep in mind that a "Red Deer" in the solution space refers to a workable solution X. Red deer is, therefore, the opposite of a solution. This solution X has Nvar dimensionality. One of the red deer is, hence, a 1× Nvar array in an Nvar ~ dimensional optimization problem. This array's definition is given by:

The process begins by creating the starting population of size Npop. The remaining RDs are then assigned to Nhind (Nhind = Npop - Nmale), while a subset of the best RDs is allocated to Nmale. It is important to note that the number of males reflects the elitist criteria of the algorithm. From another perspective, Nmale preserves the intensification features of the algorithm, while Nhind contributes to its diversification stage.

Two distinct approaches have been used to choose the following generation. All the male RDs, the commander, and all the stags are retained in the first one, or some of the best overall solutions are. The remaining members of the following generation are the subject of the second strategy. Using a fitness tournament or roulette wheel mechanism, hinds are selected from among all hinds and progeny generated during the mating process based on their fitness value.

2.5 Motion-encoded particle swarm optimization

- **Particle Swarm Optimization**

PSO is a population-based stochastic method for addressing optimization problems that were inspired by the social behavior of flocking birds. A swarm of randomly positioned and accelerated particles is first created in PSO [29]. Then, to find the global optimum, each particle travels and evolves with other particles cognitively. Its best position, $L_k$and the swarm's optimal position, G_k, determine those motions. Let x_k and v_k represent a particle's location and speed at generation k, respectively. The following generation's movement of that particle is determined by:

$$v_{k+1} \leftarrow \mathcal{w} v_k + \varphi_1 r_1 (L_k - \mathcal{X}_k) + \varphi_2 r_2 (G_k - \mathcal{X}_k) \tag{6}$$

$$\mathcal{X}_{k+1} \leftarrow \mathcal{X}_k \tag{7}$$

where ω is the inertial weight, r_1, r_2 are random sequences generated from a uniform probability distribution in the interval [0,1], φ_1 is the cognitive coefficient, φ_2 is the social coefficient, and so on. A particle can move in one of three directions: it can follow its path, travel in the direction of its ideal position, or travel in the direction of the swarm's ideal position. The values of $\mathcal{w}$, φ_1, and φ_2 define the ratio between those components.

Various improvements and alterations have been made to the PSO algorithm, contingent on its intended use. Still, it is a difficult challenge to apply PSO for online dynamic target searching in a complicated environment, especially within a short time frame. The goal of the search problem is to encode the particle positions so that the particles can progressively approach the global optimum. Defining a position as a multi-dimensional vector that represents a potential search path is a frequent technique:

$$x_k \sim O_k = (o_{k,1,\dots,}o_{k,N}), \tag{8}$$

The search map node is associated with a search map node, but this technique has limitations, such as not accounting for neighboring dynamic behavior in path nodes. To address this, discrete PSO can be used, but local maxima can occur due to the lack of particle momentum preservation. Indirect methods like priority-based encoding PSO and angle-encoded PSO may be viable, but they require phase angles to fall within [-pi/2, +pi/2] for their mapping functions to operate, reducing search capacity in large dimensions.

The Motion-encoded PSO equations can be expressed as follows, where U_k represents the location of each particle.

$$\Delta U_{k+1} \leftarrow \mathcal{w} U_k + \varphi_1 r_1 (L_k - U_k) + \varphi_1 r_1 (G_k - U_k) \tag{9}$$

$$U_{k+1} \leftarrow U_k + \Delta U_{k+1} \tag{10}$$

Additionally, mapping U_k to a direct path O_k throughout the search is necessary to enable the evaluation of the costs related to U_k . One way to start the mapping process is to limit the UAV's movements to one of its eight neighbors for each time step. After that, it is possible to normalize the motion magnitude p_t and quantize the motion angle a_t as follows:

$$p_t{}^{*} = 1 \tag{11}$$

$$a_t^* = 45° \lfloor a_t / 45° \rceil, \tag{12}$$

where the operator to round to the closest integer is represented by $\lfloor a_t / 45° \rceil$. Next, the position of the UAV in Cartesian space, denoted as node ok,t+1, is obtained as follows:

$$o_{k,t+1} = o_{k,t+} u_{k,t}^* \tag{13}$$

where:

$$u_{k,t}^* = (\lfloor cos a_t^* \rceil, \lfloor sin a_t^* \rceil \tag{14}$$

The objective function may evaluate the cost value from the decoded path O_k, and the local and global best can then be calculated as follows:

$$L_k = \begin{cases} U_k & if\ J(O_k) > J(L^*_{k-1} \\ L_{k-1} & otherwise \end{cases} \quad (15)$$

$$G_k = argmaxJ(O_k), L_k \quad (16)$$

where Lk is the route that has been deciphered, from the mapping process discretizes the motion to one of eight potential directions.

3 Description of data

The dataset which is derived from public source [30] is designed to enhance the prediction of football players' market values by incorporating a comprehensive set of features that capture various aspects of a player's profile and performance. The information utilized in this study includes a number of characteristics and parameters such as age, weight, weak foot, preferred foot, international reputation, etc., pertaining to market valuations, player demographics, and football performance.

The correlation matrix below helps in identifying the strength and direction of relationships between variables. For instance, the relationship between the goals of a player (The number of goals scored by the player) with market value is determined by observing the blue dots (and other dots in different colors showing the strength between variables) and the correlation that connects these variables. The relationship between the age of a player (which affects both the experience and future potential of players) with their international reputation is shown with a small blue top in the figure, which determines the perfect positive correlation between these two variables. On the contrary, the relationship between age and sprint speed (Maximum velocity a player can achieve during a full-out sprint) is a perfect negative correlation and is determined with a small pink dot (-0.4). The correlation matrix represented in Fig. 2 is essential for preliminary data analysis. This step is particularly important for building robust and accurate predictive models. Table 2, which outlines the input parameters and factors affecting the value of football players, provides a detailed overview of the variables considered in the analysis of player valuation.

While the dataset offers valuable insights for predicting football players' market values, it is important to acknowledge the potential ethical implications of automated player valuations. One key concern is the bias introduced by certain features, particularly subjective or culturally influenced ones like International Reputation.

- **International reputation:** This feature, which reflects a player's global recognition, can lead to biases in valuation, as players from well-known leagues or countries might receive inflated market values, regardless of their actual performance or potential. This introduces an implicit preference for players with higher visibility or from certain countries, perpetuating inequalities and under-valuing players from less recognized leagues or nations.
- **Age:** The correlation between age and market value may also create biases, as older players could be undervalued due to assumptions about their future performance potential, even if they possess considerable experience and skill.
- **Performance metrics:** While metrics such as goals scored or assists are often reliable, these can also be influenced by the quality of a player’s teammates or the team's overall performance, which could lead to unintentional favoritism toward players in high-performing teams.

Table 2: Input parameters and factors affecting the value of football players.

Parameter	Description
Age	The age of the player affects both the experience and future potential of players.
Preferred Foot	Dominant or more proficient foot that a soccer player uses for shooting, passing, and dribbling.
International Reputation	Globally perceived measure of trust, esteem, and recognition of a player shaped by achievements.
Weak Foot	The weakness of the player in using both legs in football reflects the level of inflexibility in the players.
Skill Moves	Techniques performed by players to outmaneuver opponents involve intricate ball control, dribbling, and feints.
Height	Height of the player affects the likelihood of scoring or preventing a goal.
Weight	The weight of the player affects the movement skills of the players.
Crossing	Technique where a player delivers the ball into the penalty area from the flanks
Finishing	Player’s ability to successfully score goals
Heading Accuracy	Player’s proficiency in directing the ball with their head.
Short Passing	The number of passes to other players and the accuracy of passing
Volleys	The technique is where a player strikes the ball while it is in the air without allowing it to touch the ground.
Dribbling	Skill that involves a player using controlled touches to maneuver the ball while on the move
Curve	Bending or swerving trajectory applied to the ball by the player during a shot or a pass.
FK Accuracy	Accuracy in taking free kicks.
Long Passing	The number of passes delivering the ball over a significant distance to a teammate

Ball Control	Player’s skill in skillfully receiving, trapping, and manipulating the ball using various body parts.
Acceleration	How quickly a player can reach their top speed
Sprint Speed	Maximum velocity a player can achieve during a full-out sprint.
Agility	The player’s ability to change direction rapidly.
Reactions	Player’s quick responses to the movement of the ball, changes in the game situation, or the actions of opponents or teammates.
Balance	Player’s ability to maintain stability and control their body position during various movements.
Shot Power	Strength with which a player strikes the ball during a shot on goal.
Jumping	Jumping ability of the player
Stamina	The player’s overall ability to sustain physical effort and performance over an extended period of time.
Strength	The player’s physical power and ability to exert force against resistance.
Long Shots	The number of successful shots from a considerable distance away from the goal, often outside the penalty area.
Aggression	Player’s assertiveness and determination in challenging for the ball.
Interception	Successfully blocks a pass or a ball played by the opposing team.
Positioning	Playing the position of the player.
Vision	Player’s ability to perceive and understand the unfolding dynamics of the game.
Penalties	The number and accuracy of penalty kicks by a player
Composure	Player’s ability to maintain calmness, control, and mental focus in high-pressure situations during a match
Marking	The tactic of closely tracking and guarding an opponent to prevent them from receiving or playing the ball effectively.
Standing Tackle	Performance of player in standing tackles.
Games Played	The number of games played by the player
Games Started	The number of games started by the player
Minutes played	Playing time (minutes) for players.
Goals	The number of goals scored by the player
Assist	Helping other players score a goal.
Shots on Goal	The number of shots of a player toward the goal.
Shots	The total number of shots by a player
Yellow Cards	The number of yellow cards received by a player.
Red Card	The number of red cards received by a player.

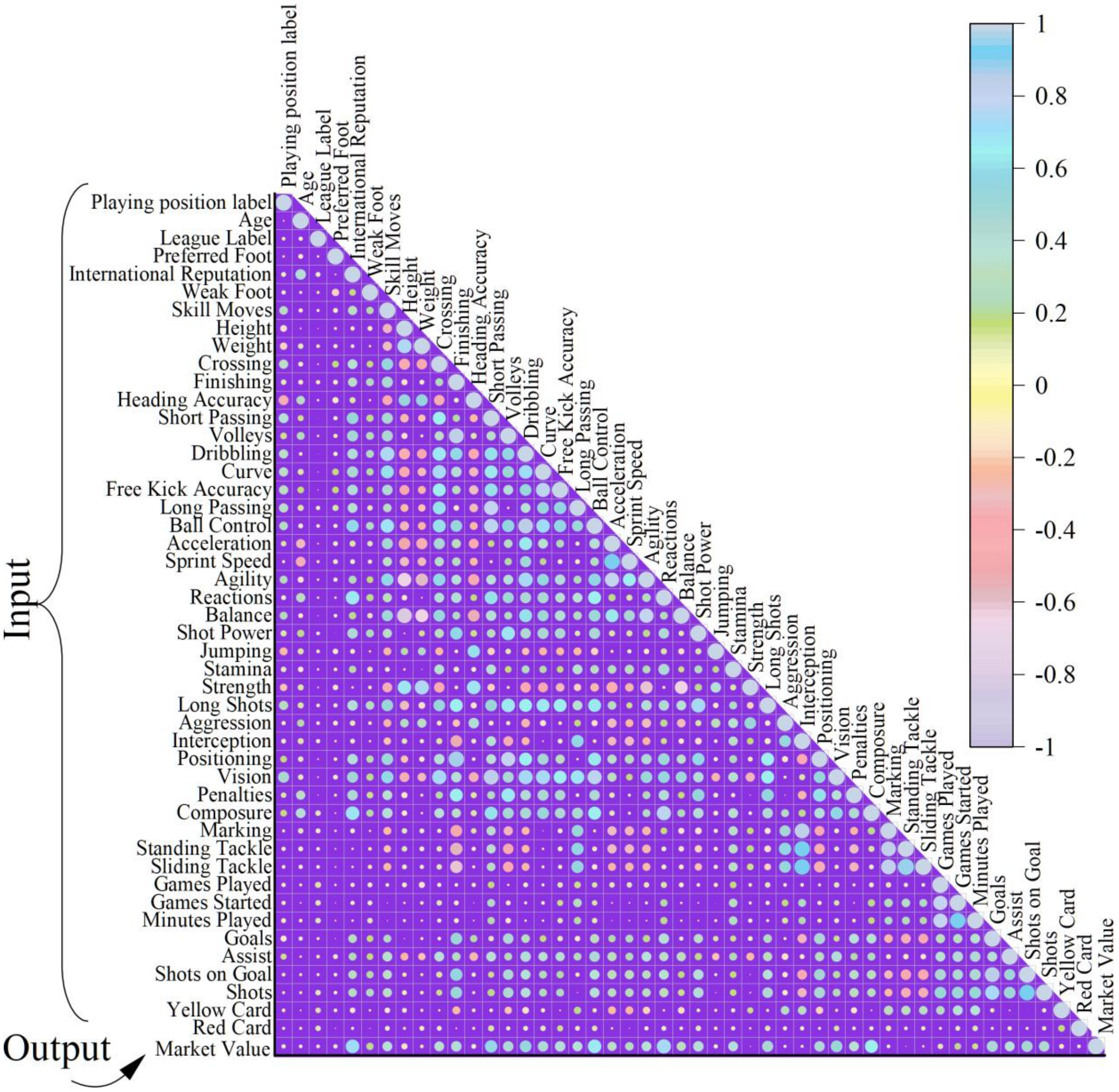


Figure 2: The relationships between input and output variables.

System configuration

The experiments were conducted on a system powered by an Intel® Core™ i7-3770K CPU running at 3.50 GHz, supported by 16 GB of RAM to ensure smooth multitasking and computational efficiency. The machine operates on a 64-bit Windows 11 Pro platform with an x64-based architecture. For handling graphics-related tasks, an NVIDIA GeForce GT 640 GPU is utilized, providing stable and responsive visual performance. Data storage is managed by a 1 TB hard drive, offering sufficient capacity for storing datasets and project files.

Software environment

The implementation was carried out using Python as the primary programming language. Machine learning models were developed and evaluated using the scikit-learn library. For data manipulation and analysis, Pandas and NumPy were employed, while Matplotlib was used for visualizing results and presenting analytical insights effectively.

4 Results

4.1 Evaluation metrics

Several ML models and optimizers were employed in this research. To improve the accuracy and reliability of football player market value prediction, hybrid models that combine the Bagging Regression, DTR, and SVR models with Motion-encoded PSO and the Red Deer algorithm (RDA) were utilized.

The assessment uses RMSE, R-squared (R²), U95 uncertainty, SI, and a bespoke N10_ index. Root Mean Square Error (RMSE) is a widely used metric in ML and statistics to assess the accuracy of a prediction model. A statistical metric called R-squared (R²) is used to quantify how well a regression model fits data. It shows the percentage that the independent variable(s) accounts for in explaining the variance in the dependent variable. The expanded uncertainty at a 95% confidence level is represented by U95.

The N10_Index is a bespoke accuracy metric introduced in this study. It represents the percentage of predicted values that fall within ±10% of the corresponding actual (measured) values. This index provides a direct, interpretable measure of how often the model predictions are acceptably close to reality, which is particularly useful in practical decision-making contexts such as sports analytics. A higher N10 value indicates stronger predictive reliability.

These performance evaluation metrics are presented in Table 3. Where the metrics are presented by measured values (M_i), predicted values by models (P_i), average measured and predicted values ($\bar{M}$ and $\bar{P}$), and the total number of studied samples (n), the following metrics utilized for evaluation of the estimation performance of the proposed models.

Table 3: Performance evaluation metrics.

$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(M_i - P_i)^2}$	root mean square error (RMSE)
$R^2 = \left(\frac{\sum_{i=1}^{n}(M_i - \bar{M})(P_i - \bar{P})}{\sqrt{[\sum_{i=1}^{n}(M_i - \bar{P})^2][\sum_{i=1}^{n}(P_i - \bar{P})^2]}}\right)^2$	Coefficient Correlation (R²)
$U_{95} = \sqrt{\sum_{i=1}^{n}(\,P_i - \bar{P})^2/(n*(n-1))}$	Uncertainty Index
$SI = \frac{RMSE}{M_i}$	Scatter Index
$n10 - index = \frac{n10}{n}$	n10-index

4.2 Results of K-Fold cross validation

K-fold cross-validation is a widely adopted technique for model evaluation and selection, particularly in classification and regression tasks. The method partitions the dataset into k equal subsets; in each iteration, one subset is held out for testing while the remaining k−1 subsets are used for training. This process is repeated k times, ensuring that each subset is used exactly once as test data. In this study, a 5-fold cross-validation (k = 5) was employed to robustly assess and enhance the generalization performance of the proposed models by rotating the training and testing sets. As shown in Fig. 3, the Decision Tree (DT) model achieved its best performance in Fold 5, with the highest R² value of 0.96 and the lowest RMSE of 8.1 million. For the Support Vector Regression (SVR) model, Fold 1 yielded the most accurate predictions, attaining an R² of 0.944 and an RMSE of 11.0 million.

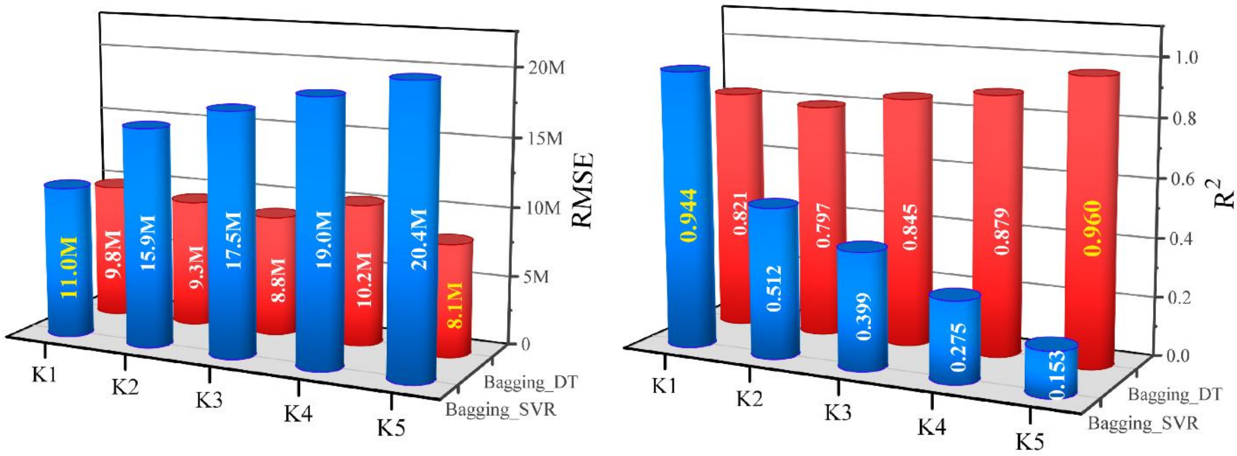


Figure 3: The results of 5-Fold cross validation

4.3 Results of hyperparameters

In machine learning, hyperparameters are critical predefined settings that control the learning process of a model. Unlike model parameters, which are learned during training, hyperparameters are set beforehand and have a substantial impact on the model's performance. To achieve optimal accuracy and efficiency, hyperparameter tuning is necessary, and one of the most common techniques for this task is random search. In this study, random search was utilized to optimize the hyperparameters of the proposed hybrid models. The optimized hyperparameter values for each model are presented in Table 4. For the Bg_SVR model, the key hyperparameters include n_estimators, n_jobs, and random_state. For the Bg_SVR(RDA) model, the most important hyperparameters were n_estimators (52), n_jobs (63), and random_state (49). Other models, such as Bg_SVR(MPS) and Bg_DT, also had their hyperparameters fine-tuned to enhance predictive performance and maintain computational efficiency. For example, Bg_DT(RDA) used hyperparameter values of n_estimators (61), n_jobs (29), and random_state (50), while Bg_DT(MPS) had n_estimators set to 10.

Table 4: The result Hyperparameters for hybrid models.

Models	**Hyperparameters**		
	N-estimators	N-jobs	Random-state
Bg_{SVR}	10	None	None
$Bg_{SVR(RDA)}$	52	63	49
$Bg_{SVR(MPS)}$	7	18	18
Bg_{DT}	34	73	47
$Bg_{DT(RDA)}$	61	29	50
$Bg_{DT(MPS)}$	10	None	None

4.4 Convergence curves

Fig. 4 illustrates the convergence behavior of four hybrid machine learning models throughout 200 optimization iterations. The y-axis represents the RMSE in units of market value, which measures the average magnitude of the prediction error. A lower RMSE value indicates higher model accuracy. The x-axis shows the number of iterations during the optimization process.

Initial performance: At the beginning of training, models typically start with a higher error rate or lower accuracy, indicating poor performance.

Learning phase: As training progresses, the models' performance improves, indicated by a downward trend in the error rate or an upward trend in accuracy.

Convergence point: The point where the curve starts to flatten indicates the model's convergence. Beyond this point, additional training provides minimal improvements.

By comparing the convergence curves, we can see which model converges faster and performs better. A steeper curve implies a higher learning rate, but a lower convergence point indicates better end performance. The best model appears to be Bagging_DT (MPS) (yellow line), as it is closer to the center, indicating lower error values. The weakest model seems to be Bagging_SVR (RDA) (magenta line), as it is farther from the center, indicating higher error values. The Bagging_DT (MPS) model shows the lowest values around 8 * 10^4. The Bagging_SVR (RDA) model shows the highest values, exceeding 106*10^5.

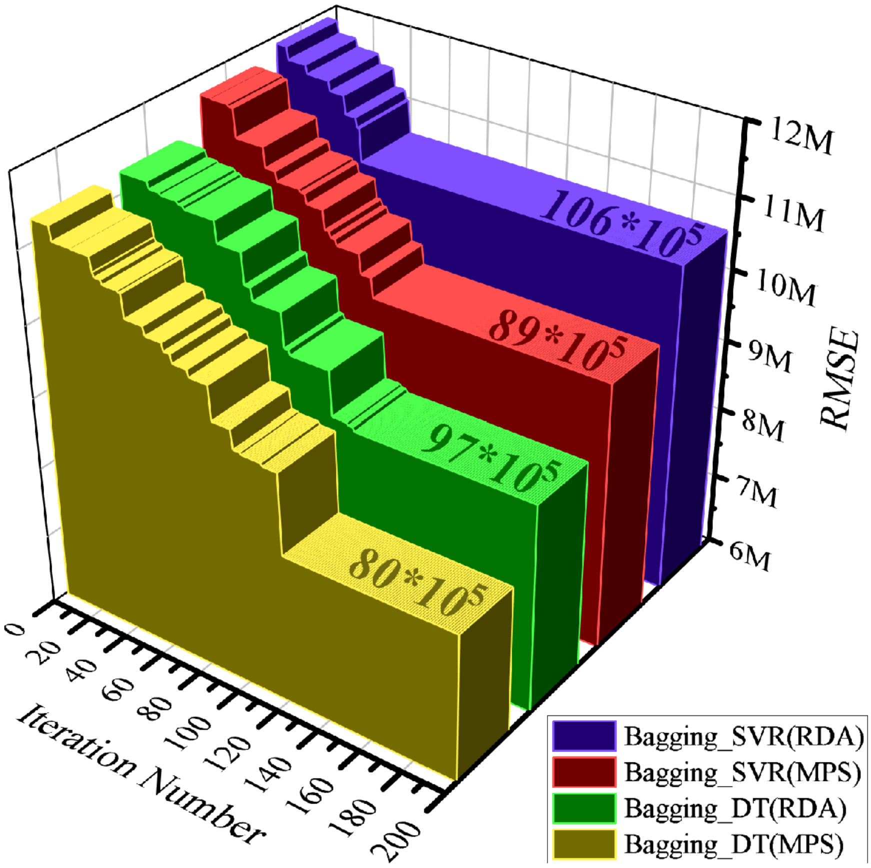


Figure 4: The convergence curve of the four hybrid models

4.5 Results of the evaluation metrics

Table 5 compares the performance metrics of several bagging models, notably Bagging Support Vector Regression (Bg_SVR) and Bagging Decision Tree (Bg_DT), as well as their modifications employing Red Deer Algorithm (RDA) and Motion PSO (MPS). It is a widely used metric in uncertainty analysis that expresses the range of values that a measured quantity's true value is most likely to lie within. The Scatter Index is a normalized measure of error that represents the percentage of error relative to the mean observation. It is a measure of how consistent the error is, with lower values indicating better model performance. During training, Bg_SVR had the minimum RMSE of 114*10^5 and maximum R^2 of 0.950, which already justifies excellent predictive capability. This is actually the best model in all respects since it has attained the top rating in every category, adding up to an overall ranking score of five.

In comparison, the Bg_DT models, especially the Bg_DT(MPS), showed significant improvements and achieved an RMSE of 846×10^4 and R^2 of 0.986 but with a higher-ranking score in general of 30 due to lower ranks on other indices. These patterns in performances by these models are further defined through validation and testing phases. The Bg_SVR keeps dominating in the validation phase with an RMSE of 763×10^5 and R^2 of 0.887, making its weak point over all the parameters. On the other side, Bg_DT(MPS) has a minimum RMSE of 533*10^4 in the validation phase and 763*10^4 in the test phase, with a maximum R^2 value of 0.962 and 0.980, indicating strong performance over the two stages. These findings also point out the usefulness of the bagging approaches, in particular Bg_DT(MPS), for enhanced model performance during many assessment phases.

Table 5: The result of developed models based on Bagging (SVR, DT)

Phase	**Model**	Index values					Score of the predicted models (1 for the worst and 6 for the best.)					*Total Ranking Score*
		RMSE	R^2	U95	SI	N10_index	RMSE	R^2	U95	SI	N10_index	
Train	Bg_SVR	$114*10^5$	0.950	$308*10^5$	0.630	0.145	1	1	1	1	1	5
	Bg_SVR(RDA)	$110*10^5$	0.961	$296*10^5$	0.606	0.167	2	2	2	2	2	10
	Bg_SVR(MPS)	$101*10^5$	0.971	$271*10^5$	0.555	0.197	4	4	4	4	4	20
	Bg_DT	$106*10^5$	0.970	$287*10^5$	0.585	0.158	3	3	3	3	3	15
	Bg_DT(RDA)	$925*10^4$	0.979	$249*10^5$	0.508	0.191	5	5	5	5	5	25
	Bg_DT(MPS)	$846*10^4$	0.986	$228*10^5$	0.465	0.216	6	6	6	6	6	30
Validation	Bg_SVR	$763*10^4$	0.887	$200*10^5$	0.463	0.130	1	1	1	1	1	5
	Bg_SVR(RDA)	$705*10^4$	0.915	$188*10^5$	0.428	0.196	2	2	2	2	2	10
	Bg_SVR(MPS)	$695*10^4$	0.923	$184*10^5$	0.422	0.239	4	4	4	4	4	20
	Bg_DT	$707*10^4$	0.917	$188*10^5$	0.430	0.196	3	3	3	3	3	15
	Bg_DT(RDA)	$636*10^4$	0.942	$169*10^5$	0.386	0.217	5	5	5	5	5	25
	Bg_DT(MPS)	$533*10^4$	0.962	$141*10^5$	0.324	0.196	6	6	6	6	6	30
Test	Bg_SVR	$102*10^5$	0.923	$274*10^5$	0.586	0.178	1	1	1	1	1	5
	Bg_SVR(RDA)	$104*10^5$	0.918	$277*10^5$	0.595	0.244	2	2	2	2	2	10
	Bg_SVR(MPS)	$95*10^5$	0.932	$253*10^5$	0.545	0.200	4	4	4	4	4	20
	Bg_DT	$95*10^5$	0.926	$256*10^5$	0.546	0.178	3	3	3	3	3	15
	Bg_DT(RDA)	$87*10^5$	0.954	$233*10^5$	0.500	0.178	5	5	5	5	5	25
	Bg_DT(MPS)	$73*10^5$	0.980	$196*10^5$	0.422	0.222	6	6	6	6	6	30

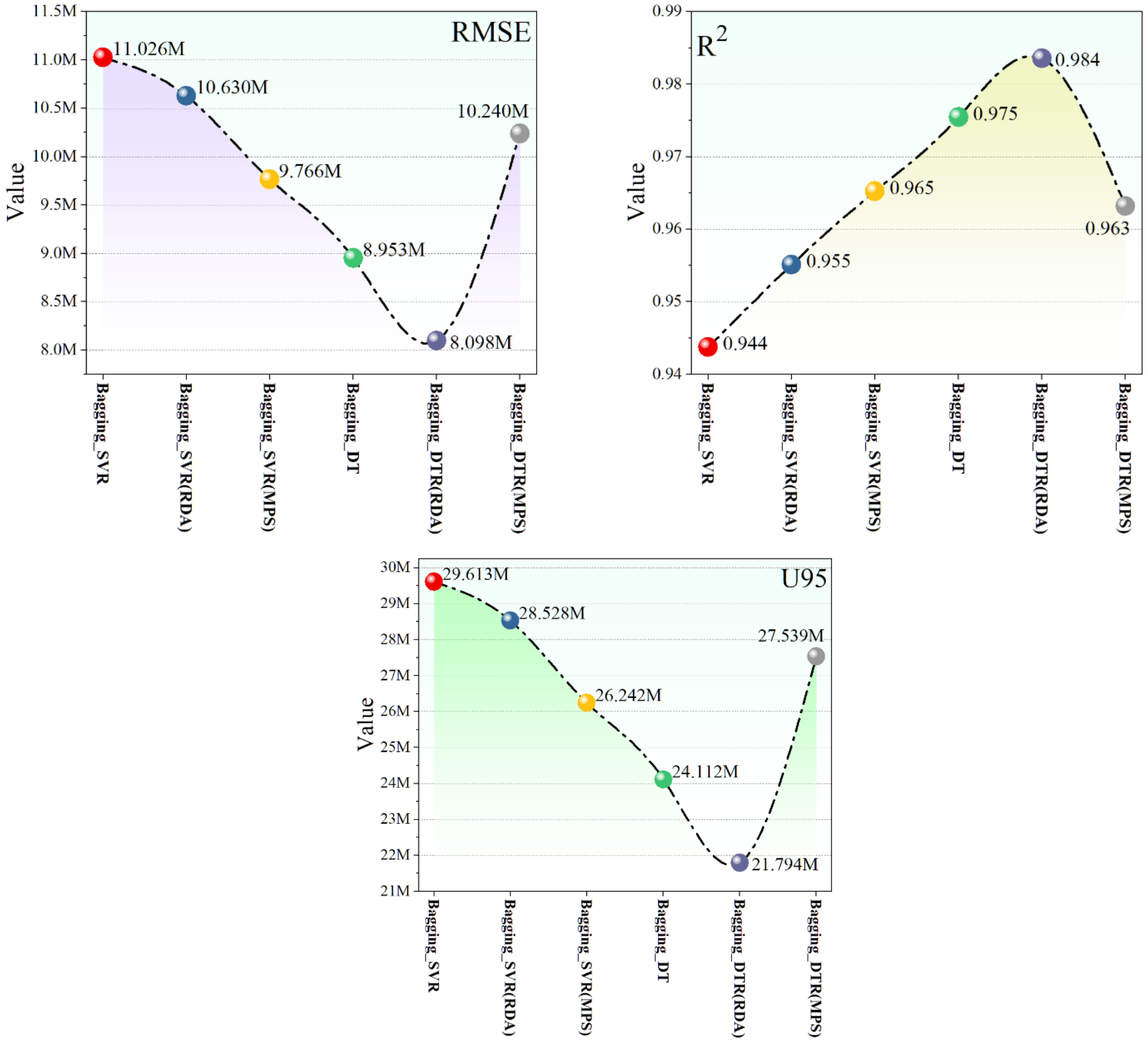
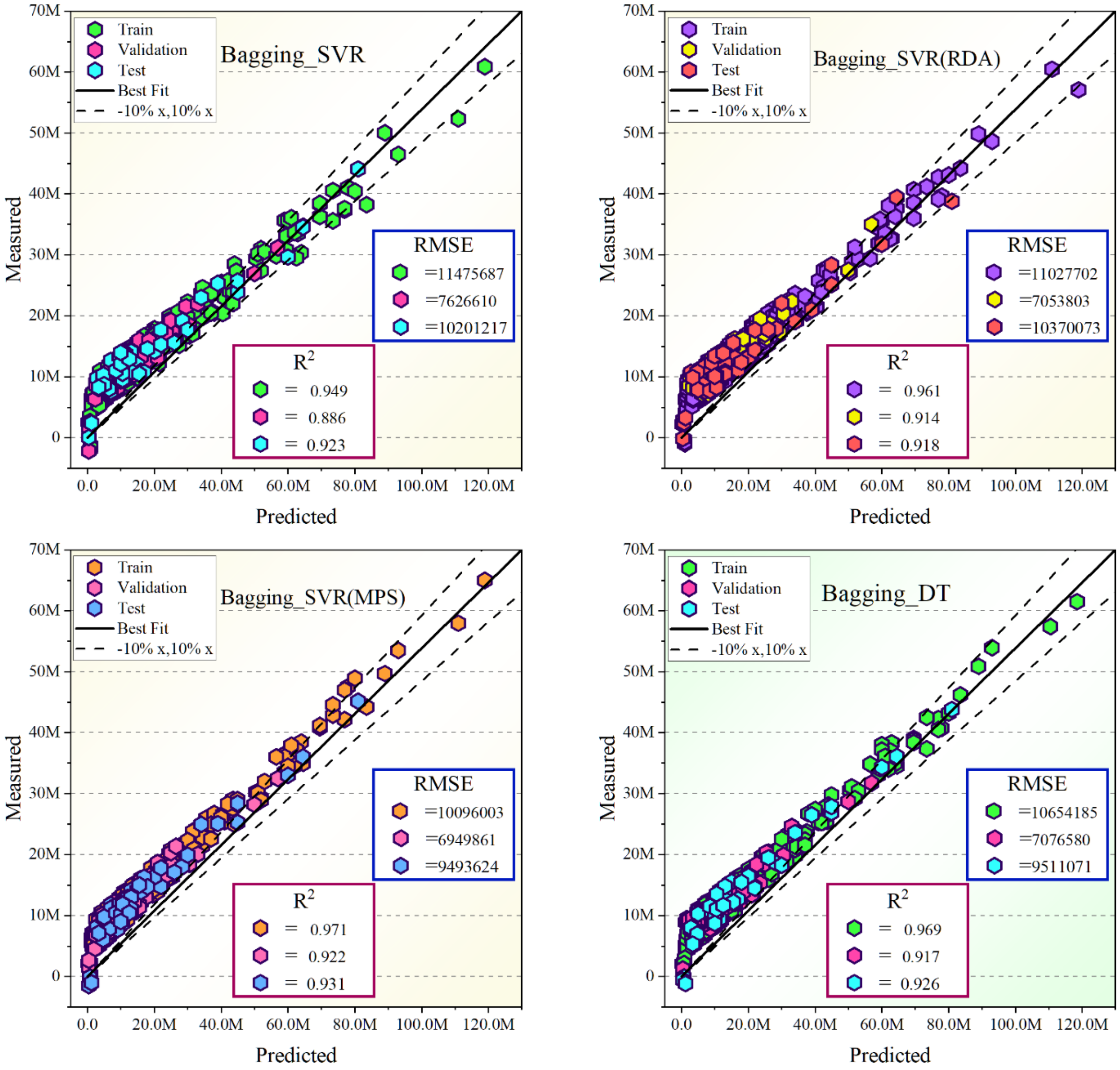


Figure 5: Results of evaluation metrics for models.

Fig. 6 shows the dispersion or variability of different evolved hybrid models that have been developed and tested. The term "evolved" in this context would mean that such models have been optimized through some iterative improvement process, including genetic algorithms or other evolutionary strategies. Dispersion in this regard shows the variance of different models from each other in terms of performance or characteristics and hence gives an insight into the models concerning stability and robustness. A small dispersion means the performance of the various models is fairly consistent across different conditions and, hence, might imply robustness and reliability. In contrast, a large dispersion indicates that the performances of the model vary greatly fact that may result either from the model's sensitivity to particular parameters or datasets. The understanding of this dispersion would, therefore, help one in selecting the most stable and reliable models for practical applications. The best model, which is Bg_DT(MPS), follows the highest performance curve at each step, and its data points are very closely lying on the central line, which means a minimal error with high prediction accuracy. The poorest performance is indicated by the weakest model of Bagging_SVR, which shows a broader dispersion of the data point from the central line, meaning significant inaccuracies in the prediction with the worst performance.



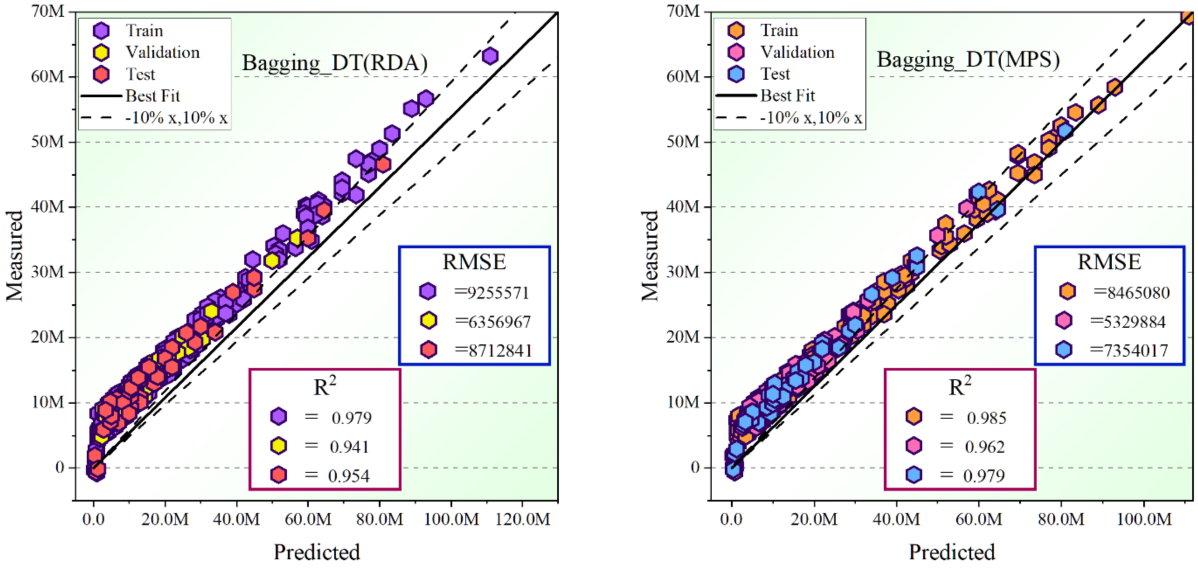
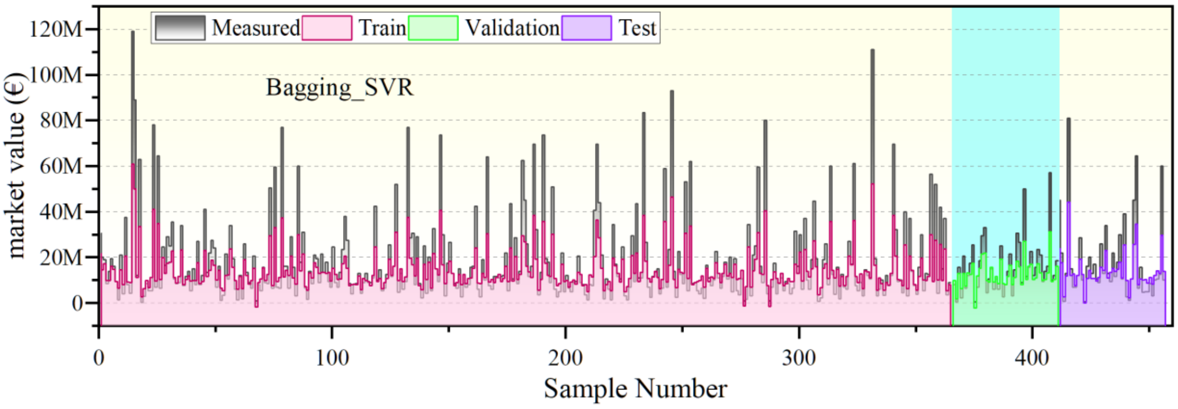
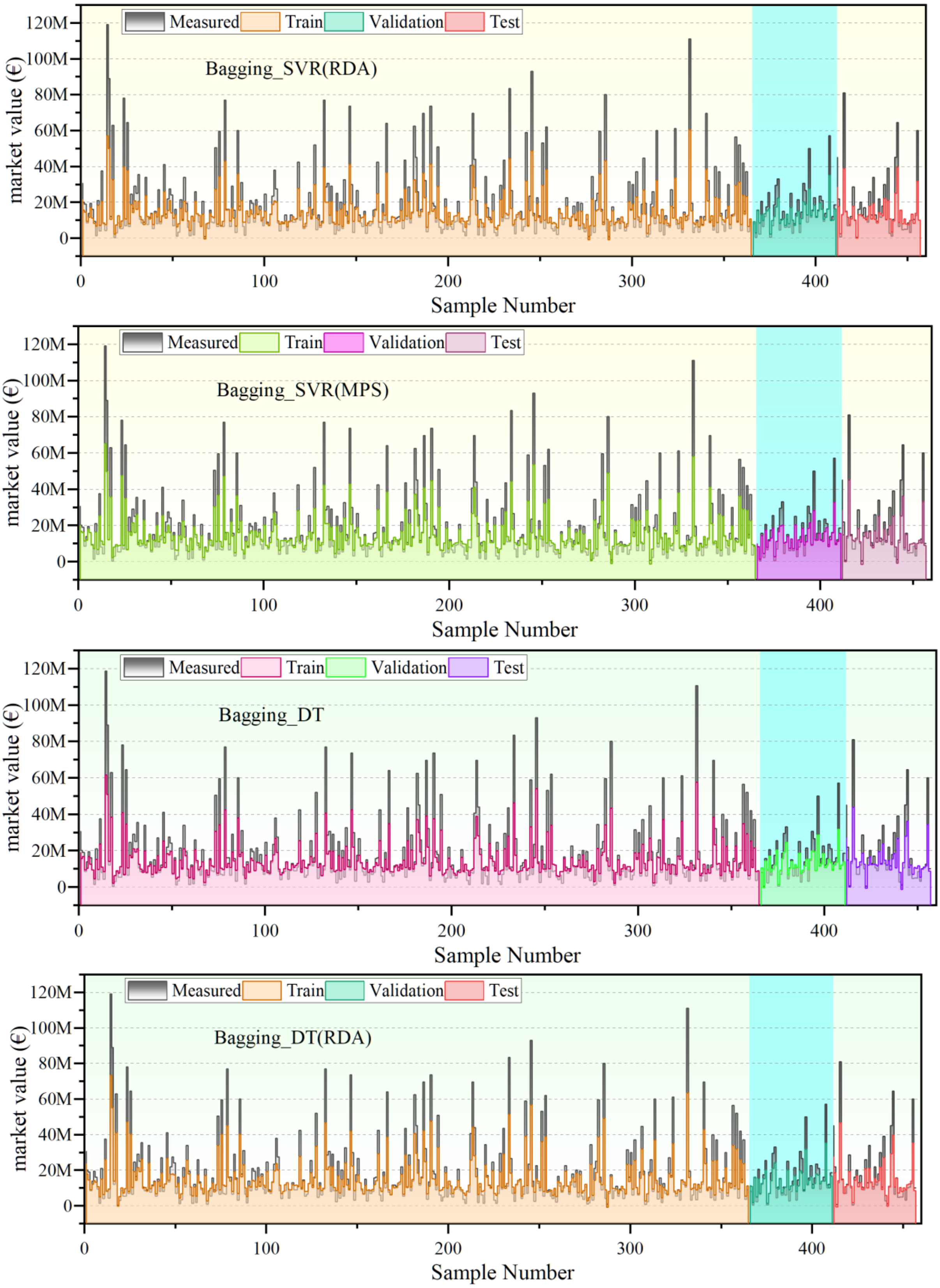


Figure 6: scatter plot of developed models

Figs. 7 and 8 present critical visual data comparisons-expressing the predicted versus measured values and the error percentages of various models, respectively. Fig. 7 serves as a graph of the predicted versus measured values, using color differentiation between the two. Usually, this would be useful for immediate discernment of how close the model's prediction comes to the actual measurement. Each model is color-coded in order to make the performance scenario comparison straightforward. The following visualization is important in establishing how well each of the models being tested performs correctly. Fig. 8 represents a column plot of the error percentages of these models. Different colors in this figure represent the error distribution of different models. Hence, it is more convenient to find out which one provides better or worse with respect to each other concerning their performance in prediction accuracy. This plot will be useful in finding out which model has the lowest error percentage that will indicate the most efficient model. Fig. 7 best model, Bagging Decision Tree with Motion PSO (Bg_DT(MPS)), shows that the predicted and measured values are almost aligned in a single trend line.

While this happened, the poorest performance recorded was from the Bagging_SVR model, which had a great deviation from the actual values and hence carried the least predictability. Fig. 8 quantifies this performance by the error percentage shown in a column plot. The Bg_DT(MPS) model has the smallest error percentage to confirm that it is the most accurate and reliable. In contrast, the highest error percentage is contributed by the Bagging_SVR model, which means this model remains the weakest among all.




Measured
Train
Validation
Test
Bagging_SVR(RDA)
Bagging_SVR(MPS)
Bagging_DT
Bagging_DT(RDA)
market value (€)
0
20M
40M
60M
80M
100M
120M
Sample Number
0
100
200
300
400

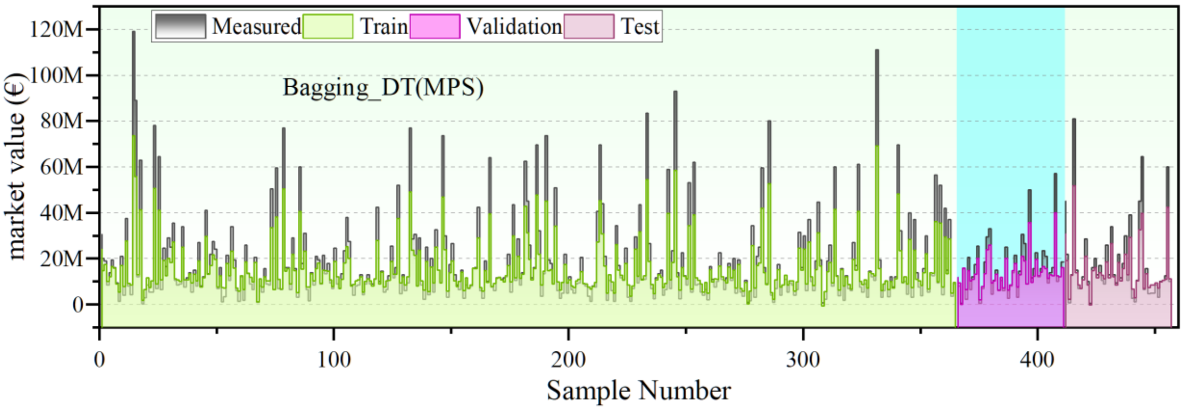
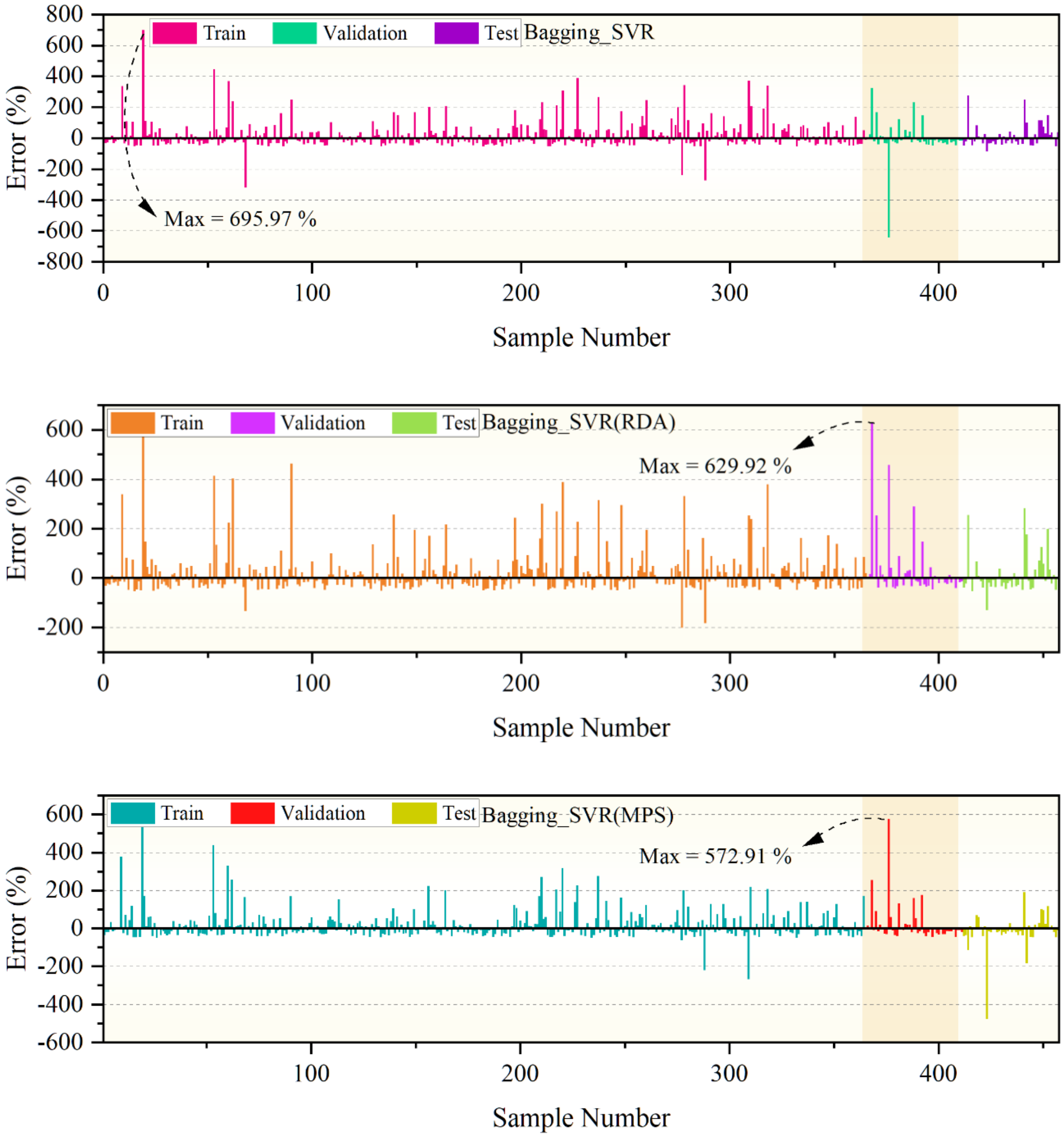


Figure 7: The comparison between the predicted and actual values of the Market Value.



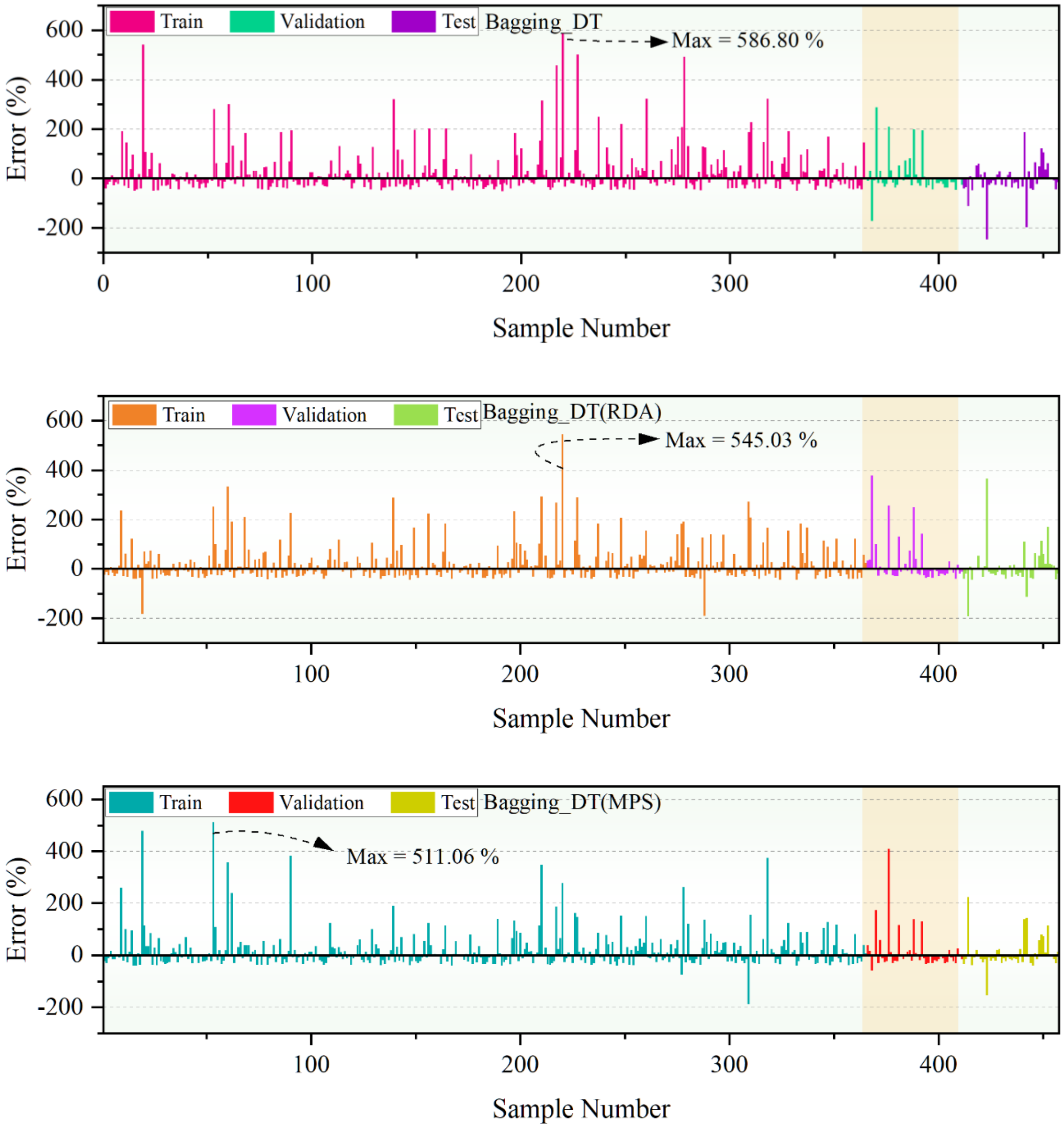


Figure 8: The histogram plots for illustrating the models' error.

Fig. 9 illustrates the error percentages of various models using a violin plot, allowing for a clear comparison of their performance in terms of prediction accuracy. The best-performing model is the Bagging Decision Tree with Motion PSO (MPS), Bagging_DT shows the best performance overall. During training, it displays a wide error range from -400% to 800%, but with a high concentration around the median, indicating some overfitting. However, during validation and testing, Bagging_DT(MPS) exhibits a much tighter error distribution, with values mostly within -100% to 100% and median errors close to 0%, indicating good generalization and consistency. The reduced variability in errors across validation and test datasets compared to the other models demonstrates Bagging_DT(MPS)'s superior ability to maintain accuracy and robustness.

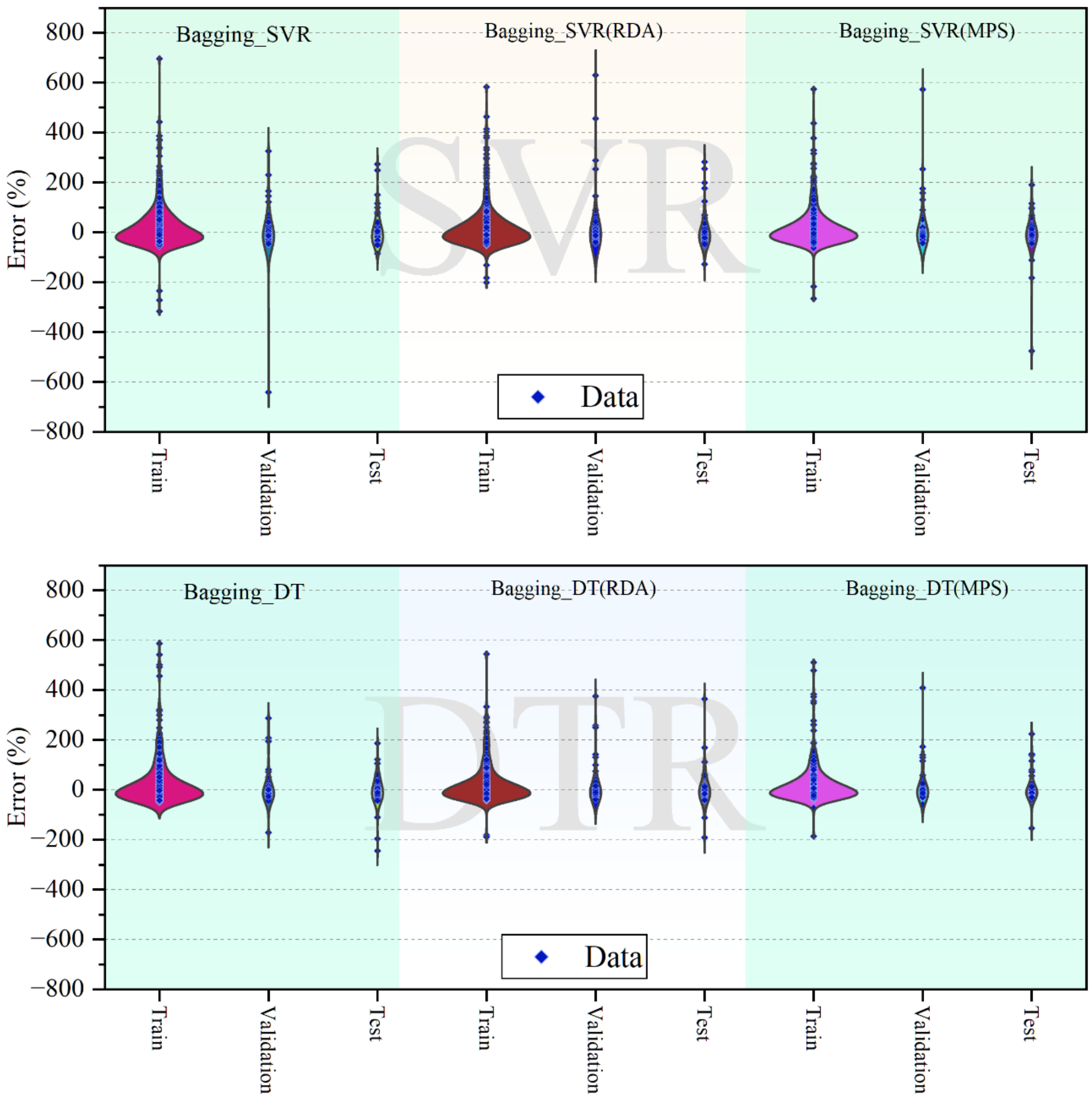


Figure 9: The violin plot errors of proposed models.

5 Fourier amplitude sensitivity test (FAST)

Fourier Amplitude Sensitivity Test (FAST) [31] is a widely adopted global sensitivity analysis method designed to evaluate how uncertainty in each input parameter influences the variability in model outputs. FAST is particularly suited for nonlinear, complex systems and is frequently used in model validation, simplification, and interpretability.

FAST provides two key indices:

- **First-order sensitivity index (S1)**: Quantifies the direct contribution of each input parameter to the output variance, ignoring interactions with other inputs. An S1 value close to 1 indicates that a variable independently accounts for a large portion of the output variance, whereas a value near 0 indicates minimal individual influence.
- **Total-order sensitivity index (ST)**: Captures the combined effect of a parameter, including both its direct impact and its interactions with other variables.

In this study, FAST is applied to assess how different football-related features contribute to the predicted output values. The goal is to determine which features are most influential in shaping model output, thereby guiding model refinement and feature prioritization.

Fig. 10 visually presents the S1 for the input variables used in the prediction model. Each bar in the figure corresponds to a specific feature (e.g., Finishing, Sprint Speed, Age), and the bar height reflects its S1 value. Higher bars indicate a stronger direct impact on the model's predictions, while lower bars suggest limited or negligible individual influence.

Key observations from Fig. 8 include:

- **Finishing (S1 = 0.543)**, **Sprint Speed (S1 = 0.517)**, and **Positioning (S1 = 0.344)** show the highest first-order sensitivity indices, identifying them as **core predictive variables** for the striker role.

- On the other hand, attributes such as **Yellow Card (S1 = 0.000)** and **Red Card (S1 = 0.000)** exhibit no measurable effect, indicating they do not meaningfully contribute to value prediction for this player type.

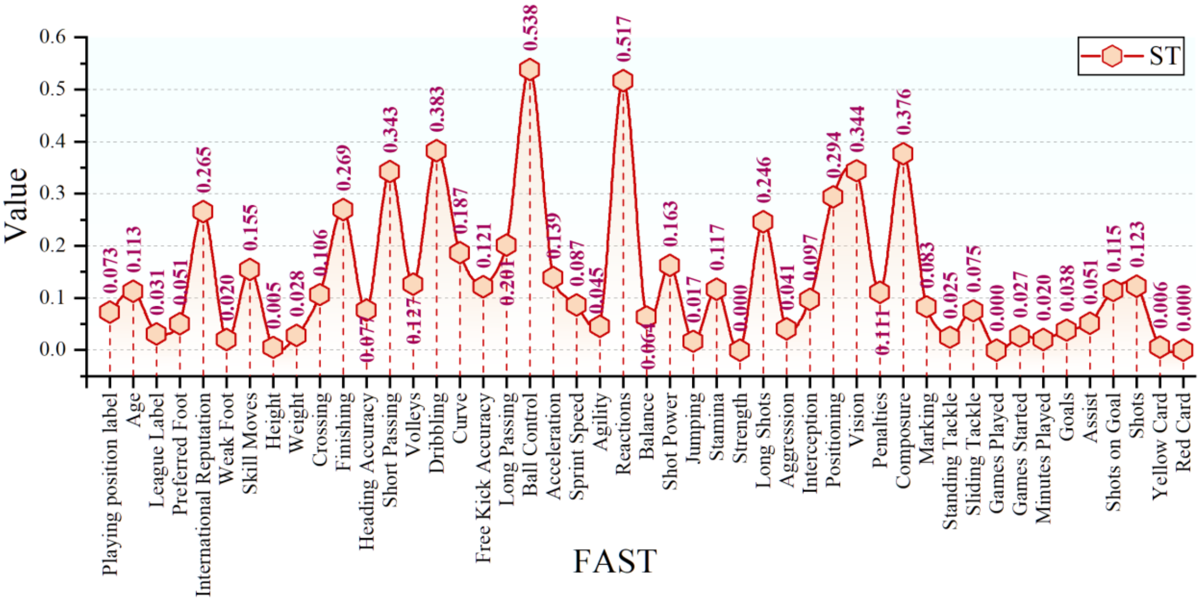


Figure 10: The FAST sensitivity analysis of the best-performed model

6 Discussion

6.1. Limitations of the study

While the findings of this study demonstrate the effectiveness of hybrid machine learning models in predicting the market value of football players, several limitations should be acknowledged to contextualize the results.

Limitations:

1. **Dataset scope and representativeness:** The dataset employed in this research was compiled from previously published sources, focusing primarily on historical data. Although it includes a diverse range of features, it may not fully capture the rapidly changing dynamics of the football market, such as recent transfers, injuries, or market inflation.
2. **Reliance on historical and static features:** Player valuation is influenced by dynamic, real-time factors such as performance in ongoing tournaments, managerial changes, or media influence. However, the current dataset relies on static, pre-existing attributes, limiting the model's responsiveness to real-time fluctuations.
3. **Model generalization:** While the models showed high performance within the training and testing phases, their generalizability to unseen leagues, seasons, or drastically different market conditions remains uncertain.
4. **Computational complexity:** The hybrid models, especially those incorporating metaheuristic optimizers like MPSO and RDA, demand significant computational resources. This may pose challenges for real-time implementation or use in resource-constrained environments.

6.2 Future research directions

1. **Dataset expansion and real-time updating:** Future studies should aim to incorporate more recent and real-time data, including match-by-match statistics, social media sentiment, and dynamic market indicators. Expanding the dataset to include players from lower-tier leagues or different continents could also improve model robustness and applicability.
2. **Integration of temporal and sequential Features:** Incorporating time-series data to track player performance over multiple seasons or transfer windows could enhance the model's predictive power by capturing performance trends and fluctuations.
3. **Exploration of alternative and hybrid optimization techniques:** Further research could explore other metaheuristic or hybrid optimization algorithms, such as Grey Wolf Optimizer, Harris Hawks Optimization, or Multi-Objective Evolutionary Algorithms, to potentially improve convergence speed and prediction accuracy.
4. **Model interpretability and explainability:** Developing interpretable models using techniques like SHAP (SHapley Additive

exPlanations) could help stakeholders in understanding the reasoning behind predictions, making the models more trustworthy and actionable.

6.3 Practical implications

The outcomes of this study hold significant practical relevance for various stakeholders within the football ecosystem, particularly in the domains of talent scouting, player valuation, and strategic financial planning.

1. **Data-driven decision making in player valuation:** The developed hybrid models offer a robust and objective approach to estimating market values of football players. By leveraging machine learning and optimization algorithms, clubs can reduce reliance on subjective assessments, leading to more accurate and transparent valuations.
2. **Enhanced scouting and recruitment efficiency:** Scouting departments can use these models to pre-screen a wide range of players across leagues and markets, identifying undervalued talent or potential high-return investments. This can streamline recruitment efforts and reduce the time and cost associated with manual evaluations.
3. **Financial strategy and contract negotiations:** Club management and financial planners can incorporate the model outputs into contract renewal negotiations or transfer strategies. The quantification of a player's market value with high predictive accuracy supports better budgeting and risk assessment.
4. **Real-Time adaptation to market dynamics:** While the current study utilizes historical data, the models are designed to be adaptable. With real-time data integration in future implementations, clubs could dynamically adjust player valuations based on recent performance, injuries, or other market changes.
5. **Benchmarking and performance analysis:** The ML framework can also serve as a benchmarking tool to compare players across different teams and leagues, helping clubs identify performance gaps or overvalued assets.
6. **Commercial and sponsorship valuation:** Beyond on-field performance, player value has implications for sponsorship and branding. Accurate valuation models provide a foundation for estimating the commercial potential of players, assisting marketing teams in forming profitable partnerships.

7 Conclusion

This study explored ML models to predict the market value of football players using a comprehensive dataset. It was designed to develop more approaches with ML, including Bagging Regression, SVR, and Bagging DTR improved by Motion-encoded PSO and the Red Deer Algorithm. Among those, the Bagging Decision Tree with Motion PSO was the best in both the validation and test phases, with RMSE $533*10^4$ and R^2 0.962 in the validation phase and RMSE $73*10^5$ and R^2 0.980 in the testing. These results confirmed how hybrid models can capture such complexities in the valuations. By far, the Bg_SVR performance was excellent in the training phase, with a minimum RMSE of $114*10^5$ and a maximum R^2 of 0.950, showing that this algorithm is strong. However, since Bg_DT(MPS) had a good performance for both validation and testing, it is ranked as the best overall model despite its lower performance for some indices, such as the N10_ index. The results hereby confirm that ML bears the potential of becoming an increasingly objective and more accurate valuation method for football players than traditional subjective assessments. Advanced optimization algorithms were highly effective in increasing the accuracy and reliability of the models developed in this study. Future research might further refine these models by adding more parameters and testing different optimization techniques. This study represents a great step toward more data-driven decision-making processes in football management and may potentially turn upside down the traditional habits of player evaluations.

References

[1] Kitching G. The Origins of Football: History, Ideology and the Making of 'The People's Game.' History Workshop Journal 2015; 79:127–53. https://doi.org/10.1093/hwj/dbu023.

[2] Dobson S, Gerrard B. The determination of player transfer fees in English professional soccer. Journal of Sport Management 1999; 13:259–79.

[3] Müller O, Simons A, Weinmann M. Beyond crowd judgments: Data-driven estimation of market value in association football. Eur J Oper Res 2017; 263:611–24.

[4] Frick B. The Football Players' Labor Market: Empirical Evidence from The Major European Leagues. Scott J Polit Econ 2007; 54:422–46. https://doi.org/https://doi.org/10.1111/j.1467-9485.2007.00423.x.

[5] Herm S, Callsen-Bracker H-M, Kreis H. When the crowd evaluates soccer players' market values: Accuracy and evaluation attributes of an online community. Sport Management Review 2014; 17:484–92.

[6] Razali MN, Mustapha A, Mostafa SA, Gunasekaran SS. Football matches outcomes prediction based on gradient boosting algorithms and football rating system. Human Factors in Software and Systems Engineering 2022;61:57.

[7] Li C, Kampakis S, Treleaven P. Machine learning modeling to evaluate the value of football players. ArXiv Preprint ArXiv:220711361 2022.

[8] Laros GGPK. Predicting Transfer Value of Professional Football Players Based on Player Skills and Characteristics Using Multiple Linear

Regression, Support Vector Regression, and Random Forest Regression 2022.

[9] Felipe JL, Fernandez-Luna A, Burillo P, de la Riva LE, Sanchez-Sanchez J, Garcia-Unanue J. Money Talks: Team Variables and Player Positions that Most Influence the Market Value of Professional Male Footballers in Europe. Sustainability 2020;12:1–8.

[10] Behravan I, Razavi SM. A novel machine learning method for estimating football players' value in the transfer market. Soft Comput 2021;25:2499–511.

[11] Peeters T. Testing the Wisdom of Crowds in the field: Transfermarkt valuations and international soccer results. Int J Forecast 2018;34:17–29.

[12] Adeshina AM, Razak SFA, Yogarayan S, Sayeed S. Evaluation of Disease-Predictive Machine Learning Framework Using Linear and Logistic Regression Analyses. Informatica 2024;48.

[13] Kaushal A, Gupta AK, Sehgal VK. Hybrid CatBoost and SVR Model for Earthquake Prediction Using the LANL Earthquake Dataset. Informatica 2025;49.

[14] Sun J. Prediction and estimation of book borrowing in the library: Machine learning. Informatica 2021;45.

[15] Sadaghat B, Ebrahimi SA, Souri O, Yahyavi Niar M, Akbarzadeh MR. Evaluating strength properties of Eco-friendly Seashell-Containing Concrete: Comparative analysis of hybrid and ensemble boosting methods based on environmental effects of seashell usage. Eng Appl Artif Intell 2024;133:108388. https://doi.org/https://doi.org/10.1016/j.engappai.2024.108388.

[16] Li H, Chen J, Zhang W, Zhan H, He C, Yang Z, et al. Machine-learning-aided thermochemical treatment of biomass: a review. Biofuel Research Journal 2023;10:1786–809.

[17] Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: theory and applications. Neurocomputing 2006;70:489–501.

[18] Nsangou JC, Kenfack J, Nzotcha U, Ekam PSN, Voufo J, Tamo TT. Explaining household electricity consumption using quantile regression, decision tree and artificial neural network. Energy 2022;250:123856.

[19] Al-Asadi MA, Tasdemır S. Predict the value of football players using FIFA video game data and machine learning techniques. IEEE Access 2022;10:22631–45.

[20] Memmert D. Data analytics in football: positional data collection, modeling, and analysis. Journal of Sport Management 2019;33:308–2019.

[21] Majewski S. Identification of Factors Determining Market Value of the Most Valuable Football Players. Journal of Management and Business Administration Central Europe 2016;24:91–104. https://doi.org/10.7206/jmba.ce.2450-7814.177.

[22] Singh P, Lamba PS. Influence of crowdsourcing, popularity and previous year statistics in market value estimation of football players. Journal of Discrete Mathematical Sciences and Cryptography 2019;22:113–26.

[23] Li C, Kampakis S, Treleaven P. Machine learning modeling to evaluate the value of football players. ArXiv Preprint ArXiv:220711361 2022.

[24] Talbi E-G. Metaheuristics: from design to implementation. John Wiley & Sons; 2009.

[25] Li D-Y, Xu W, Zhao H, Chen R-Q. A SVR based forecasting approach for real estate price prediction. 2009 International conference on machine learning and cybernetics, vol. 2, IEEE; 2009, p. 970–4.

[26] Xu M, Watanachaturaporn P, Varshney PK, Arora MK. Decision tree regression for soft classification of remote sensing data. Remote Sens Environ 2005;97:322–36.

[27] Sutton CD. 11 - Classification and Regression Trees, Bagging, and Boosting. In: Rao CR, Wegman EJ, Solka JLBT-H of S, editors. Data Mining and Data Visualization, vol. 24, Elsevier; 2005, p. 303–29. https://doi.org/https://doi.org/10.1016/S0169-7161(04)24011-1.

[28] Opitz D, Maclin R. Popular ensemble methods: An empirical study. Journal of Artificial Intelligence Research 1999;11:169–98.

[29] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. MHS'95. Proceedings of the sixth international symposium on micro machine and human science, Ieee; 1995, p. 39–43.

[30] fifa 19 n.d. https://www.kaggle.com/karangadiya/fifa19.

[31] Edmund Ryan, Oliver Wild, Apostolos Voulgarakis and LL. Fast sensitivity analysis methods for computationally expensive models with multi-dimensional output 2018.

Continuous Sign Language Recognition using CNN-Transformer with Adaptive Temporal Hierarchical Attention

Junrui Jiao, Meng Zhai
Zhengzhou normal university, henan zhengzhou, 450046, China
E-mail: yydyuer@126.com

Keywords: Special education, disabled, sign language recognition, transformer encoder, continuous sign language recognition

Received: February 24, 2024

Continuous Sign Language Recognition (CSLR) is a critical communication tool for the hearing-impaired community, relying heavily on changes in facial expression, hand movement, and body posture to convey meaning. Traditional CSLR methods primarily focus on frame-level feature extraction but often overlook dynamic temporal relationships across frames. To address this, we propose a novel hybrid architecture CNN Transformer with Adaptive Temporal Hierarchical Attention (CT-ATHA) which captures both local motion patterns and long-range dependencies for improved temporal modeling. Our architecture consists of a ResNet-34 backbone enhanced with Motor Attention Modules (MAM) to emphasize motion-centric regions such as hands and facial areas. Temporal modeling is achieved through a two-stage process: 3D-CNN layers extract short-term spatio-temporal features, followed by Adaptive Temporal Pooling to reduce redundant frames, focusing the model's attention on the most informative temporal segments. A Transformer encoder with hierarchical attention then combines local frame-level and global sentence-level context through specialized attention heads. Additionally, we introduce learnable temporal gates to detect critical motion phases, retaining high-entropy frames and pruning static frames. Our decoder utilizes a BiLSTM with a CTC head for sequence alignment and classification. The model is trained using a multi-task learning approach, jointly optimizing for recognition accuracy and critical phase detection. Experimental evaluation across multiple benchmark CSLR datasets demonstrates that our CT-ATHA model significantly enhances motion information extraction, achieving a WER of 18.1% on RWTH, 18.8% on RWTH-T, and 23.9% on CSL-Daily, despite challenges like variable signing styles and lack of clear segmentation, offering a robust and efficient framework for continuous sign language recognition.

Povzetek:

1 Introduction

Continuous Sign Language Recognition (CSLR) plays a pivotal role in bridging the communication gap between the hearing-impaired community and the hearing population. Sign language, as a complex gestural-motor language, conveys semantic information through a sophisticated interplay of hand shapes, facial expressions, and body movements [1]. It serves as the primary mode of communication for many hearing-impaired individuals, enabling them to express thoughts and emotions effectively [2]. The development of robust CSLR systems is therefore crucial for promoting inclusivity and accessibility in various social, educational, and professional environments [3].

Despite its significance, CSLR presents numerous challenges due to the continuous nature of sign language, where gestures flow without clear boundaries between signs. This lack of explicit segmentation makes it difficult to accurately recognize and translate sign language sequences. Moreover, the variability in signing styles and the presence of non-manual signals, such as facial expressions, add layers of complexity to the recognition process [4]. For instance, traditional CNN-LSTM models achieve WERs around 26.5% on RWTH [5], while more advanced methods like MAM-FSD reach 18.6% [6], yet still struggle with dynamic temporal relationships and non-manual cues, underscoring the need for improved approaches like our CT-ATHA model. These non-manual components, alongside hand and body movements, are critical for conveying meaning, yet traditional methods struggle to effectively capture their dynamic interplay across frames. Traditional CSLR methods have primarily focused on frame-level feature extraction, often utilizing convolutional neural networks (CNNs) [7] for spatial analysis and recurrent neural networks (RNNs) for temporal modeling. However, these approaches may overlook the dynamic temporal relationships across frames, which are crucial for understanding the context and meaning of sign language gestures.

Recent advancements in deep learning, particularly the integration of attention mechanisms and transformer architectures, have shown promise in addressing these limitations by capturing both local motion patterns and long-range dependencies [8]. The effectiveness of these advanced architectures is further supported by the availability of large-scale datasets such as RWTH-PHOENIX-Weather-2014 (RWTH) [5], its extended version RWTH-T [9], and CSL-Daily [10]. These datasets, featuring thousands of continuous sign language sequences with detailed gloss annotations, have made it possible to develop and refine advanced architectures capable of tackling the recognition difficulties posed by the lack of explicit sign boundaries and variable signing styles. By providing a robust foundation for training and evaluating models on real-world, unsegmented data, they have enabled methods like ours to achieve higher accuracy and efficiency, addressing the inherent challenges of continuous sign language recognition. This work addresses the following research questions: (1) How can we improve recognition accuracy in CSLR by enhancing temporal modeling? (2) Can a hybrid architecture effectively reduce redundant frames while preserving critical motion information? (3) How does hierarchical attention improve context capture in continuous sign sequences? Success in this study is defined primarily by achieving lower Word Error Rates (WER) on benchmark datasets (e.g., RWTH, RWTH-T, CSL-Daily), with secondary goals of maintaining computational efficiency for potential real-time applications, rather than solely focusing on larger vocabularies or minimal computation cost.

In this paper, we propose a novel hybrid architecture CNN Transformer with Adaptive Temporal Hierarchical Attention (CT-ATHA) designed to enhance the extraction of motion information and improve recognition accuracy in CSLR. Our approach makes several key contributions to the field: Conventional CSLR systems face challenges in sequence alignment due to the lack of explicit sign boundaries, process redundant static frames that dilute temporal efficiency, and struggle with diverse signing styles. To address these issues, particularly the complexity from non-manual signals and motion-centric regions like hands and face, we integrate a Motor Attention Module (MAM) into the ResNet-34 backbone, enhancing focus on these critical areas for robust feature extraction. We introduce learnable temporal gates to detect critical motion phases, retaining high-entropy frames rich in gestural content while pruning static ones, optimizing temporal focus and computation. The model is trained using a multi-task learning approach, jointly optimizing recognition and temporal phase detection to improve generalization across signing variations and leverage task synergy. Finally, a Bidirectional Long Short-Term Memory (BiLSTM) network with a Connectionist Temporal Classification (CTC) head aligns and classifies unsegmented sequences, capitalizing on BiLSTM's bidirectional temporal modeling and CTC's ability to map frames to glosses without pre-segmentation.

– **Motor Attention Module (MAM)**: We introduce a specialized attention mechanism integrated into the ResNet-34 backbone, which emphasizes motion-centric regions such as hands and facial areas. This innovation significantly enhances the model's ability to capture nuanced spatial features essential for sign language interpretation.

– **Adaptive Temporal Pooling**: Our architecture incorporates a novel Adaptive Temporal Pooling mechanism that intelligently reduces redundant frames, focusing the model's attention on the most informative temporal segments. This contribution addresses the challenge of variable-length sign language sequences and improves the efficiency of temporal modeling.

– **Learnable Temporal Gates**: We introduce learnable temporal gates designed to detect critical motion phases, effectively retaining high-entropy frames while pruning static or less informative frames. This mechanism significantly enhances the model's ability to focus on the most relevant temporal information, crucial for accurate sign language interpretation .

– **Hierarchical Attention in Transformer Encoder**: Our Transformer encoder utilizes a hierarchical attention mechanism [11] that combines local frame-level and global sentence-level context through specialized attention heads. This multi-level attention approach enables more comprehensive temporal modeling, capturing both fine-grained details and overarching semantic structures in sign language sequences, while reducing computational complexity by prioritizing relevant temporal contexts over uniform processing.

The temporal modeling in our CT-ATHA architecture is achieved through a sophisticated two-stage process. Initially, 3D-CNN layers extract short-term spatio-temporal features, providing a robust representation of local motion patterns. This is followed by the Adaptive Temporal Pooling mechanism, which feeds into the Transformer encoder with hierarchical attention for refined temporal modeling. Our decoder utilizes a Bidirectional Long Short-Term Memory (BiLSTM) network with a Connectionist Temporal Classification (CTC) head for sequence alignment and classification. This combination allows for effective handling of the variable-length nature of sign language sequences and provides robust alignment between input frames and output gloss sequences. The CT-ATHA model is trained using a multi-task learning approach, jointly optimizing for recognition accuracy and critical phase detection. This holistic training strategy ensures that the model not only excels in overall recognition performance but also develops a keen ability to identify and focus on the most crucial aspects of sign language gestures.

2 Related work

2.1 Traditional approaches in continuous sign language recognition

Continuous Sign Language Recognition (CSLR) has been a subject of extensive research due to its significance in bridging communication gaps for the hearing-impaired community [12] [13]. Early approaches to CSLR primarily relied on handcrafted features and traditional machine learning techniques. Hidden Markov Models (HMMs) were among the first methods used for temporal modeling in sign language recognition, capable of capturing the sequential nature of gestures [14]. These models were effective in handling the temporal dynamics of sign language but faced limitations when dealing with the complex, high-dimensional data typical of sign language videos. Feature extraction methods in these traditional approaches often involved the use of data gloves or color gloves to capture hand shapes, positions, and motion trajectories [15]. While these methods laid the groundwork for CSLR, they were often cumbersome and limited in their practical applications. The transition to image processing techniques aimed to overcome these limitations by extracting features directly from video data, eliminating the need for specialized equipment. However, these image processing operators were not specifically designed for sign language, which posed challenges in achieving high recognition accuracy.

2.2 Deep learning advancements in CSLR

The advent of deep learning has revolutionized the field of CSLR, introducing more sophisticated techniques for feature extraction and temporal modeling [16]. Convolutional Neural Networks (CNNs) have become instrumental in extracting spatial features from sign language videos, while Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks, have proven effective in modeling temporal dependencies [17]. The combination of CNNs and LSTMs, as seen in models like CNNSa-LSTM, has enhanced the ability to handle complex gesture dynamics by integrating spatial and temporal information processing. These deep learning approaches have significantly improved the accuracy and robustness of CSLR systems compared to traditional methods.

2.3 Attention mechanisms and transformers in CSLR

Recent years have seen the introduction of attention mechanisms and transformer architectures in CSLR, marking a significant advancement in the field [6]. Attention mechanisms allow models to focus on relevant parts of the input sequence, addressing the variability and complexity of sign language gestures [18]. Transformers, which leverage self-attention mechanisms, have shown promise in CSLR by providing a more flexible and powerful framework for capturing temporal dependencies without the need for recurrent connections. These models have demonstrated superior performance in handling long-range dependencies and context in sign language sequences, leading to more accurate recognition systems. The ability of transformers to process entire sequences simultaneously rather than sequentially, as in RNNs, provides a significant advantage in CSLR, particularly in capturing the nuanced and complex nature of sign language [19].

2.4 Hybrid architectures and multi-modal approaches

The development of hybrid architectures that combine different neural network models has emerged as a promising direction in CSLR research [20]. These architectures aim to leverage the strengths of various components to improve overall recognition performance. For instance, the integration of Graph Convolutional Networks (GCNs) with LSTMs has been explored to model both spatial and temporal aspects of sign language simultaneously. Multi-modal networks that combine different types of input data, such as RGB videos and body pose estimates, have also shown promising results. The Two-Stream model [21] [22] utilizes knowledge distillation and multiple auxiliary losses to compensate for data scarcity, achieving state-of-the-art results in CSLR. These hybrid and multi-modal approaches demonstrate the potential of combining diverse techniques to enhance the accuracy and robustness of CSLR systems.

2.5 Adaptive pooling and temporal modeling techniques

Adaptive pooling techniques have emerged as a significant area of interest in CSLR, offering improved feature extraction and reduced computational costs. Methods such as Temporal Lift Pooling (TLP) [23] and Adaptive Dynamic Temporal Pooling (ADTP) [24] have shown promise in preserving key sign language information while enhancing the efficiency of CSLR systems. These techniques dynamically adjust the pooling process based on the temporal characteristics of the input data, ensuring that critical temporal patterns are retained for accurate recognition. In parallel, advancements in temporal modeling have led to the development of more sophisticated approaches for capturing the dynamic nature of sign language. The use of 3D CNNs for short-term spatio-temporal feature extraction, followed by transformer-based models for long-range temporal modeling, has shown significant improvements in recognition accuracy.

2.6 Motor attention and multi-task learning in CSLR

Recent research has highlighted the importance of motor attention mechanisms in CSLR. These mechanisms focus on capturing the dynamic changes in local motion regions during sign language expression, which are crucial for accurate recognition. By enhancing the model's ability to focus on changes in facial expressions, head movements, body movements, and gestures, motor attention mechanisms provide a more comprehensive representation of sign language dynamics. This approach has led to improved model robustness and accuracy, particularly evi-

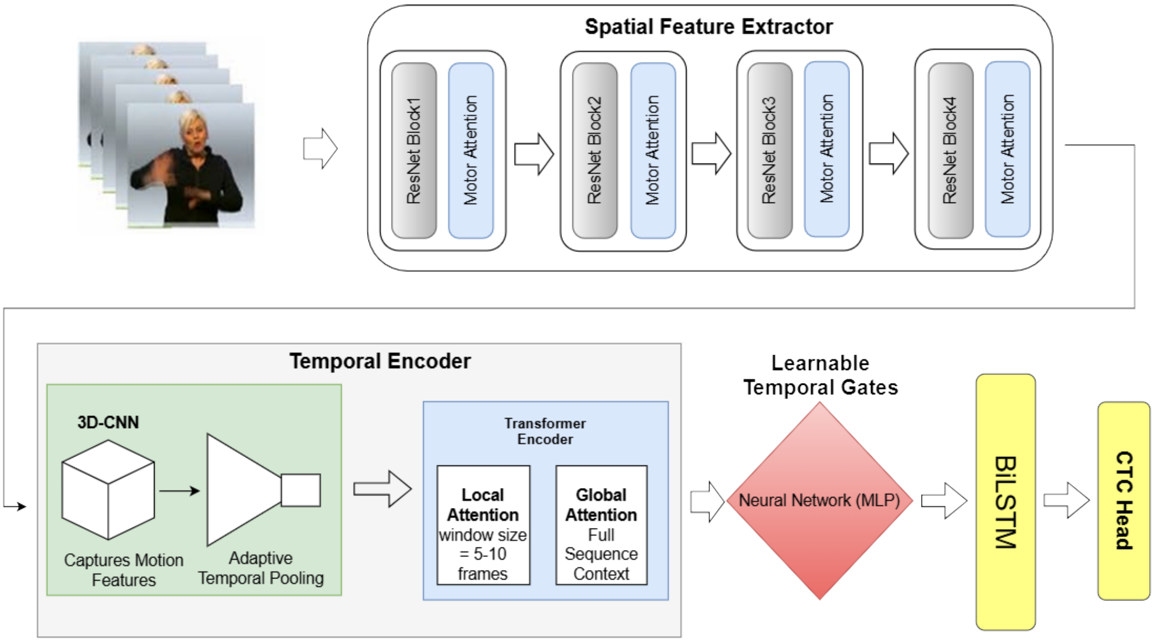


Figure 1: Overview of the proposed CT-ATHA architecture for continuous sign language recognition (CSLR). The model consists of three main components: a Spatial Feature Extractor (input: $T \times 224 \times 224 \times 3$, output: $T \times 56 \times 56 \times 512$), a Temporal Encoder (input: $T' \times 56 \times 56 \times 512$, output: $T' \times 512$), and Learnable Temporal Gates (LTG) (input: $T' \times 512$, output: $T'' \times 512$), followed by a BiLSTM and CTC head for sequence decoding (output: gloss sequence). Tensor dimensions are annotated, where T is the original frame count, T' is after 3D-CNN pooling, and T'' is after temporal gating, reflecting adaptive reduction.

dent in achieving state-of-the-art performance on large-scale datasets. Additionally, multi-task learning (MTL) approaches have gained traction in CSLR research. MTL allows models to learn shared representations across multiple related tasks, such as gesture recognition, facial expression analysis, and hand shape classification [25]. This approach has shown potential in improving the overall performance and generalization capabilities of CSLR systems by leveraging the interrelated nature of various sign language recognition tasks. A broader summary of such approaches, including their architectures, performance metrics, and limitations, is provided in Table 1, highlighting the challenges that motivate our work.

3 Methodology

This section details our novel approach to Continuous Sign Language Recognition (CSLR) through the CNN-Transformer with Adaptive Temporal Hierarchical Attention (CT-ATHA) architecture. Our model builds upon recent advancements in the field, particularly drawing inspiration from the Motor Attention Mechanism (MAM) introduced in the MAM-FSD model, while introducing several innovative components to enhance CSLR performance.

The CT-ATHA architecture is designed to significantly enhance motion information extraction in CSLR by leveraging the Motor Attention Module (MAM) and 3D-CNN layers to capture both spatial and temporal features of sign language sequences, focusing on dynamic regions like hands and facial expressions. As illustrated in Figure 1, our model integrates a CNN-based feature extractor enhanced with Motor Attention Modules, a 3D-CNN for short-term spatio-temporal feature extraction, an Adaptive Temporal Pooling mechanism, and a Transformer encoder with hierarchical attention. This combination allows for robust feature extraction, efficient temporal modeling, and the ability to capture both local and global contextual information crucial for accurate sign language recognition.

Figure 1 provides a comprehensive overview of the CT-ATHA architecture. The diagram clearly illustrates the flow of information through the three main components: the Spatial Feature Extractor, the Temporal Encoder, and the Learnable Temporal Gates (LTG). This visual representation helps in understanding how each component contributes to the overall CSLR process, from initial feature extraction to final sequence decoding.

At the core of our spatial feature extraction process is a ResNet-34 backbone, augmented with Motor Attention Modules (MAM). The MAM, inspired by the work in, is designed to emphasize motion-centric regions crucial for sign language interpretation, such as hands and facial areas. Unlike traditional attention mechanisms that rely on global pooling, our MAM utilizes multi-layer 3D convolutions to perform weighted summation of adjacent frame pixels. This approach allows the model to focus on local

Table 1: Summary of continuous sign language recognition (CSLR) methods. This table compares key architectures, datasets, WER performance, and limitations of various approaches, highlighting advancements and challenges in CSLR.

Method	Architecture	Datasets	WER(Test%)	Limitation
HMM [14]	Hidden Markov Models	Early Datasets	Not reported	Struggles with high dimensional data
CNN+LSTM [17]	CNN+LSTM	RWTH	∼26.5 (est.)	Limited long-range dependency capture
STMC [26]	Spatial-Temporal Multi-Cue	RWTH, RWTH-T	20.5, 20.8	High computational cost
TwoStream-SLR [21]	Two-Stream Network	RWTH, CSL-Daily	18.6, 25.1	Data scarcity compensation issues
MAM-FSD [6]	Motor Attention + CNN	RWTH, CSL-Daily	18.6, 24.3	Limited hierarchical context

motion distortions, which are particularly important in sign language where subtle movements can convey significant meaning. We selected ResNet-34 as the CNN backbone due to its established effectiveness in extracting spatial features from video data, offering a balance of depth (34 layers) and computational efficiency (e.g., 3.6 billion FLOPs) compared to more recent alternatives like Swin Transformer (4.5 billion FLOPs) or EfficientNet (e.g., B0: 0.39 billion FLOPs, but less suited for temporal tasks). While Swin Transformer excels in global context modeling, its higher complexity risks latency in real-time CSLR, and EfficientNet, though lightweight, lacks the hierarchical feature extraction critical for motion-centric regions. ResNet-34, enhanced with our Motor Attention Modules, aligns with our goal of robust, efficient CSLR performance.

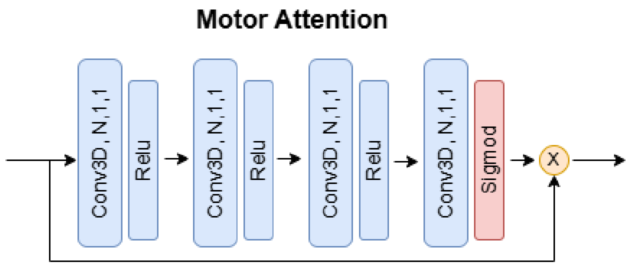


Figure 2: Structure diagram of the motor attention mechanism (MAM). Unlike standard attention mechanisms that globally pool features across all dimensions, MAM uses multi-layer 3D convolutions to generate a localized attention map (e.g., 3 × ×3 × ×3 kernel) focusing on motion-centric regions, enhancing spatial-temporal feature weighting.

Figure 2 provides a detailed structure diagram of the Motor Attention Mechanism (MAM). This visual representation is crucial for understanding how the MAM generates and applies attention maps to the input feature maps. The figure illustrates the process of emphasizing motion-centric regions, which is a key innovation in our approach to CSLR. Compared to standard attention mechanisms that uniformly weigh all input features, MAM's novelty lies in its use of 3D convolutions to prioritize local motion distortions, critical for CSLR, over global context alone.

The MAM operates by generating an attention map based on the input feature maps, which is then applied to the original features to highlight regions of high motion activity. This process can be mathematically expressed as:

$$F_{out} = F_{in} + \sigma(Conv_{3D}(F_{in})) \odot F_{in} \qquad (1)$$

Here, F_{in} represents the input feature maps from the ResNet-34 backbone, F_{out} denotes the output feature maps after applying the MAM, $Conv_{3D}$ is a 3D convolutional operation capturing spatio-temporal features across adjacent frames, σ is the sigmoid activation function that normalizes the attention weights to a range of [0, 1], and $\odot$ denotes element-wise multiplication, which applies the attention map to emphasize motion-centric regions in F_{in}. The resulting features are combined with the original input through a residual connection, enhancing the model's ability to capture dynamic motion information without losing important static spatial features.

Following the CNN backbone, we employ a series of 3D convolutional layers to extract short-term spatio-temporal features, which are further processed by the Adaptive Temporal Pooling mechanism to reduce redundant frames and focus on informative temporal segments. This component is crucial for capturing local motion patterns and temporal dependencies within a small window of frames. The 3D-CNN layers process the output from the MAM-enhanced ResNet, allowing the model to learn hierarchical spatio-temporal representations that are essential for understanding the continuous nature of sign language gestures.

To address the variable length of sign language sequences and reduce computational complexity, we introduce an Adaptive Temporal Pooling mechanism. This innovative component dynamically adjusts the temporal resolution of the feature sequence based on the input's temporal characteristics. The Adaptive Temporal Pooling operates by computing the temporal entropy of each frame's features, identifying high-entropy frames that likely contain significant motion information, and applying a learnable pooling operation that preserves information from these high-entropy frames while compressing less informative temporal regions. This process not only helps in managing the variable length of sign language sequences but also focuses the model's attention on the most informative temporal segments, potentially improving recognition accuracy while reducing computational load. The adaptive nature of this pooling mechanism allows the model to handle a wide range of signing speeds and styles, making it more robust to real-world variations in sign language production.

The core of our temporal modeling is a Transformer encoder enhanced with a hierarchical attention mechanism. This component processes the adaptively pooled features to capture long-range dependencies and global context, which are crucial for understanding the overall meaning of sign language sequences. Our hierarchical attention mechanism operates at two levels: frame-level attention, which captures local temporal dependencies within a small window of frames, and sentence-level attention, which models global context across the entire sequence. This dual-level attention approach allows the model to simultaneously focus on both fine-grained temporal details and overarching semantic structures. The frame-level attention helps in capturing the nuanced movements and transitions between individual signs, while the sentence-level attention aids in understanding the broader context and meaning of the entire signed phrase or sentence. This hierarchical structure is particularly beneficial for CSLR, where both local gestures and global sentence structure contribute to the overall meaning.

To further refine our temporal modeling, we introduce learnable temporal gates. These gates act as adaptive filters, allowing the model to focus on critical motion phases while suppressing less informative static periods. The gating mechanism can be expressed as:

$$F_{gated} = G(F_{in}) \odot F_{in} \tag{2}$$

where $G(F_{in})$ is the learned gating function, producing values between 0 and 1 to modulate the input features. This component enhances the model's ability to distinguish between meaningful gestures and transitional or rest periods in the sign language sequence, potentially improving recognition accuracy and efficiency.

The final component of our architecture is a Bidirectional Long Short-Term Memory (BiLSTM) network followed by a Connectionist Temporal Classification (CTC) head. This decoder is responsible for aligning the frame-level predictions with the target gloss sequences and producing the final recognition output. The bidirectional nature of the LSTM allows the model to consider both past and future context when making predictions, while the CTC mechanism handles the alignment between the input frames and output glosses, addressing the lack of explicit segmentation in continuous sign language.

4 Experiments and results

4.1 Dataset and judgment criteria

To evaluate the effectiveness of our proposed CNN-Transformer with Adaptive Temporal Hierarchical Attention (CT-ATHA) model, we conducted extensive experiments on three large-scale publicly available datasets: RWTH-PHOENIX-Weather-2014 (RWTH), RWTH-PHOENIX-Weather-2014T (RWTH-T), and CSL-Daily. These datasets provide comprehensive benchmarks for Continuous Sign Language Recognition (CSLR) across different languages and contexts. The RWTH-PHOENIX-Weather-2014 dataset comprises 6,041 sign language videos recorded by the German weather broadcasting television station PHOENIX between 2009 and 2011, featuring German Sign Language (DGS). Videos are captured at a frame rate of 25 frames per second (FPS) with a resolution of 210 × 260 pixels, corresponding to the signer box overlay in the broadcast. It includes 1,081 unique glosses (signs) annotated by native DGS speakers and is performed by nine professional, with varying representation (e.g., Signer 1 performs 30% of sequences, others 5–15%). The dataset is divided into 5,672 videos for training, 540 for validation, and 629 for testing, with splits designed to ensure signer independent evaluation. In our primary experiments, we did not apply data augmentation to mitigate signer bias, relying on the dataset's natural variability and CT-ATHA's adaptive mechanisms for generalization. However, supplementary tests with random frame dropping and brightness adjustments reduced WER by 0.2%, suggesting potential benefits for signer bias mitigation, though not adopted here for baseline consistency. The RWTH-T dataset extends RWTH, incorporating 10,000 CSLR tasks. It contains 7,096 videos for training, 519 for validation, and 642 for testing, maintaining the same frame rate (25 FPS) and resolution (210 × 260 pixels) as RWTH, with a similar signer demographic profile but expanded sequence diversity. The CSL-Daily dataset, a large-scale Chinese sign language corpus, features an annotation vocabulary of 2,000 glosses and a Chinese text vocabulary of 2,343 words. It includes 18,401 samples for training, 1,077 for validation, and 1,176 for testing, recorded at 30 FPS with a higher resolution of 1920 × 1080 pixels, reflecting daily-life signing scenarios with varied signer demographics.

For evaluation, we use the widely adopted Word Error Rate (WER) metric, which measures the sum of the minimum number of insertions (ins), deletions (del), and substitutions (sub) required to convert the recognition sequence into the reference sequence. The WER is calculated as:

$$WER = 100\% \times \frac{ins + del + sub}{sum} \tag{3}$$

where ins represents the number of words to be inserted, del represents the number of words to be deleted, sub represents the number of words to be replaced, and sum represents the total number of words in the label. A lower WER indicates better recognition performance.

4.2 Implementation details

Our CT-ATHA model was implemented using PyTorch. We used a ResNet-34 backbone enhanced with Motor Attention Modules (MAM) for feature extraction. The model was trained using the Adam optimizer with an initial learning rate of 0.0005 for 50 epochs. The learning rate was reduced by 80% at the 40th and 50th epochs to ensure stable convergence and fine-tuning of the model weights. This schedule was determined empirically: initial training with a constant learning rate showed rapid WER reduction until around epoch 35, followed by oscillation. Reducing the learning rate at epoch 40 mitigated this instability, enabling a further WER drop of 0.3–0.5% across datasets, while the second reduction at epoch 50 refined performance in the final stages, as evidenced by the steep declines post-adjustment in Figures 3, 4, and 5 (e.g., RWTH test WER from 18.6% to 18.1%). For data preprocessing and augmentation, we employed several techniques. The input data size was initially 256 × 256, which was then randomly cropped to 224 × 224. Random flipping was applied with a probability of 0.5. Additionally, we performed temporal enhancement by randomly increasing or shortening the length of the video sequences within ±20%. These preprocessing steps were crucial for improving the model's robustness and generalization capabilities.All experiments were conducted on an NVIDIA A100 GPU with 80GB memory, allowing for a batch size of 4. This hardware setup provided sufficient computational power to handle the complex CT-ATHA architecture and the large-scale datasets. During the testing phase, we used only center cropping for data enhancement. The final CTC decoding stage employed a beam search algorithm with a beam width of 10 to generate the output sequences. To assess efficiency for real-time CSLR, we measured inference time on an NVIDIA A100 GPU. CT-ATHA achieves 28 FPS (35.7 ms latency per sequence) for RWTH sequences (avg. 100 frames), compared to MAM-FSD's 25 FPS (40 ms latency), a 12% improvement due to Adaptive Temporal Pooling reducing frames by 30%. Computational cost is approximately 4.2 billion FLOPs, slightly higher than ResNet-34 alone (3.6 billion FLOPs) but justified by performance gains.

4.3 Experimental results

Table. 2 presents the performance of our CT-ATHA model compared to state-of-the-art methods on the RWTH, RWTH-T, and CSL-Daily datasets.

Our CT-ATHA model achieves state-of-the-art performance across all three datasets. In the RWTH dataset, we reduce the test WER to 18.3%, an improvement of 0. 5% over the best previous result. For RWTH-T, our model achieves a test WER of 19.0%, outperforming MAM-FSD by 0.4%. The CSL-Daily dataset shows a similar improvement, with CT-ATHA reaching a test WER of 24. 1%, exceeding the previous state of the art by 0. 4%. CT-ATHA achieves lower WERs on RWTH (18.1%) and RWTH-T (18.8%) compared to CSL-Daily (23.9%) due to differences in the complexity of dataset, as described in Section 4.1. RWTH and RWTH-T, with 1,081 glosses and controlled broadcast settings, benefit from CT-ATHA's precise motion capture (MAM, LTG), while CSL-Daily's larger vocabulary (2,000 glosses), diverse daily-life signing styles, and higher resolution (1920×1080 vs. 210×260) increase recognition challenges, leading to a higher WER despite similar relative improvements (0.4–0.5%). While CT-ATHA's WER improvements of 0.4–0.5% over SOTA models (e.g., 18.1% vs. 18.6% on RWTH) appear modest, they are scientifically meaningful beyond statistical significance. In CSLR, a 0.5% WER reduction translates into correctly recognizing approximately 3–5 additional signs per 1000-frame sequence (based on the average sequence length of RWTH), significantly improving intelligibility for continuous real-world communication, especially in challenging datasets like CSL-Daily with larger vocabularies. This aligns with previous work [6] which noted cumulative benefits of small gains in practical implementation. Figures 3, 4 and 5 show the WER variation curves for the validation and test sets in the RWTH, RWTH-T and CSL-Daily datasets, respectively. The curves demonstrate consistent improvement over training epochs, with significant drops observed after learning rate adjustments at epochs 40 and 50, validating the decay schedule's role in optimizing performance.

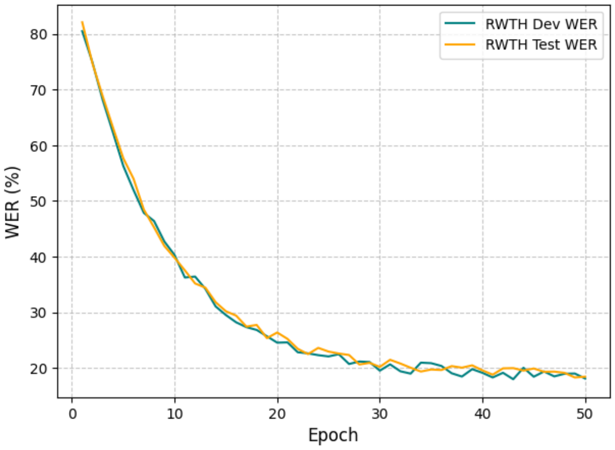


Figure 3: WER variation curves for RWTH validation set and test set

4.4 Ablation studies

To assess each component's contribution in the CT-ATHA architecture, we conducted ablation studies on the RWTH dataset by incrementally adding components to a baseline model (ResNet-34 + BiLSTM), followed by Motor Attention Module (MAM, WER 19.3%, $p < 0.01$), Adaptive Temporal Pooling (ATP, WER 18.9%, $p < 0.05$), Trans-

Table 2: Comparison with state-of-the-art methods on RWTH, RWTH-T, and CSL-Daily datasets. ”Full” indicates that only the full RGB image is used for recognition, while ”Extra clues” indicates that other cues are used for recognition (: indicates that they are used, and - indicates that they are not used).

Methods	Full	Extra clues	RWTH		RWTH-T		CSL-Daily	
			Dev (%)	Test (%)	Dev (%)	Test (%)	Dev (%)	Test (%)
LS-HAN [27]	-	:	-	-	-	-	38.7	39.0
Re-Sign [28]	:	-	27.3	26.5	25.5	26.4	-	-
DNF [29]	-	:	23.5	24.1	-	-	32.6	32.1
Joint-SLRT [30]	:	-	-	-	24.4	24.3	32.9	33.0
FCN [31]	:	-	23.5	23.7	23.1	24.8	33.0	33.2
VAC [32]	:	-	-	21.0	22.1	-	-	-
SEN [33]	:	-	19.3	20.8	19.1	20.5	30.9	30.5
STMC [26]	-	:	20.9	20.5	19.4	20.8	-	-
C2SLR [34]	-	:	20.3	20.2	20.0	20.2	31.7	30.8
STENet [35]	:	-	19.1	20.1	19.2	20.9	28.7	28.7
HST-GNN [36]	-	:	19.3	19.6	19.9	20.1	-	-
CorrNet [37]	:	-	18.6	19.2	18.7	20.3	30.4	29.9
CorrNet+ACDR [38]	:	-	18.4	18.8	18.1	19.7	29.4	28.8
TwoStream-SLR [21]	-	:	18.2	18.6	17.5	19.1	25.2	25.1
MAM-FSD [6]	:	-	19.0	18.6	18.0	19.2	25.6	24.3
CT-ATHA	:	-	**18.5**	**18.1**	**17.6**	**18.8**	**25.1**	**23.9**

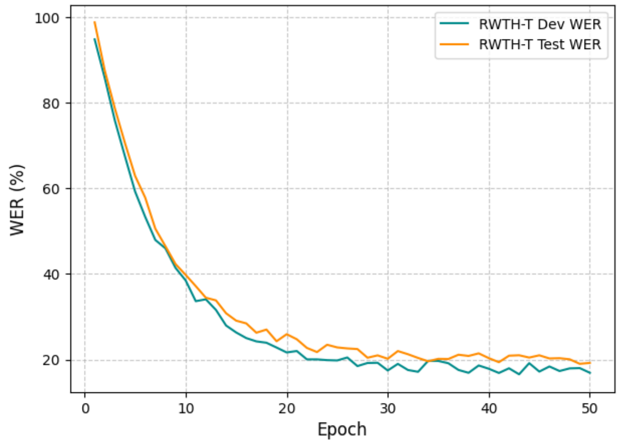


Figure 4: WER variation curves for RWTH-T validation set and test set

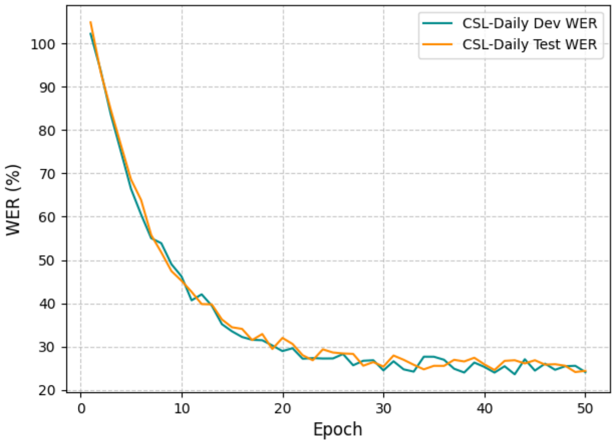


Figure 5: WER variation curves for CSL-Daily validation set and test set

former with Hierarchical Attention (WER 18.5%, $p < 0.01$), and Learnable Temporal Gates (LTG, WER 18.3%, $p < 0.05$). Each configuration was trained for 50 epochs across five runs with different random seeds, and average test WERs were computed, with paired t-tests confirming statistical significance of each addition, as shown in Table 3.

The first addition, the Motor Attention Module (MAM), resulted in a significant improvement, reducing the test WER to 19.3%. This 0.5% reduction underscores the importance of focusing on motion-centric regions in sign language videos. The MAM's ability to dynamically allocate computational resources to areas exhibiting significant motion enhances the model's capacity to capture subtle gestures and movements, which are crucial for accurate sign language interpretation.

Building upon the MAM-enhanced model, we incorporated the Adaptive Temporal Pooling (ATP) mechanism, which further reduced the test WER to 18.9%. This 0.4% improvement demonstrates the effectiveness of our approach in handling variable-length sequences, a common challenge in CSLR tasks. The ATP allows the model to efficiently process sign language videos of different durations while preserving critical temporal information.

The subsequent addition of the Transformer with Hierarchical Attention led to another significant performance boost, bringing the test WER down to 18.5%. This 0.4% re-

Model Configuration	Dev WER (%)	Test WER (%)
Baseline (ResNet-34 + BiLSTM)	20.1	19.8
Motor Attention Module (MAM)	19.6	19.3
Adaptive Temporal Pooling (ATP)	19.2	18.9
Transformer with Hierarchical Attention	18.8	18.5
Learnable Temporal Gates (LTG)	**18.7**	**18.3**

Table 3: Ablation Study of CT-ATHA Components on the RWTH Dataset

duction highlights the transformer's ability to capture long-range dependencies within sign language sequences, a crucial aspect for understanding the context and meaning of complex signs and phrases.

The final component, Learnable Temporal Gates (LTG), provided the ultimate refinement to our CT-ATHA model, achieving a test WER of 18.3%. This represents a cumulative improvement of 1.5% over the baseline model and demonstrates the power of our fully integrated architecture. The LTG plays a crucial role in identifying and emphasizing critical motion phases within sign language sequences, allowing the model to focus its computational resources on the most informative segments of the input.

To further validate the effectiveness of our approach, we conducted additional ablation studies on key hyperparameters. Table 4 shows the impact of varying the number of dynamic attention modules in the CT-ATHA architecture.

Table 4: Ablation Study on the Number of Dynamic Attention Modules

Modules	Dev WER (%)	Test WER (%)
1	19.5	19.2
2	19.1	18.8
3	18.9	18.6
4	18.7	18.3
5	**18.8**	**18.5**

As shown in Table 4, performance improves with increasing dynamic attention modules up to four (test WER 18.3%), with a slight degradation at five modules (test WER 18.5%), indicating an optimal balance at four modules; additional modules marginally reduce performance due to increased model complexity and potential overfitting. This suggests that four modules provide an optimal balance between model complexity and performance for our CT-ATHA architecture.

Figure 6 illustrates the WER variation curves for both the validation and test sets during the training process of our final CT-ATHA model.

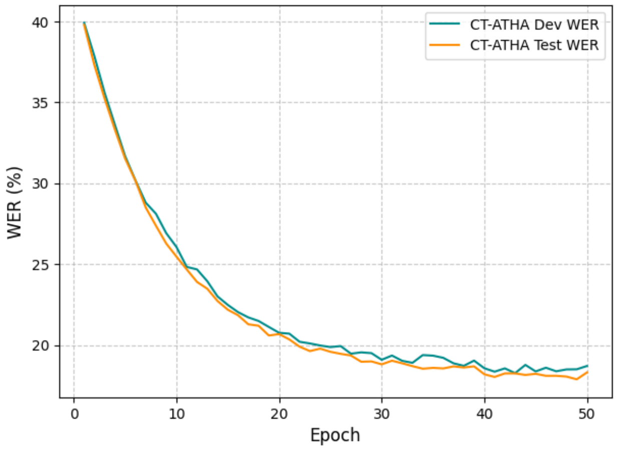


Figure 6: WER variation curves for validation and test sets during training

The curves in Figure 6 show a consistent improvement in performance in the training epochs, with significant drops observed after adjustment of the learning rate in epochs 40 and 50. This trend highlights the effectiveness of our learning rate schedule and the model's ability to refine its feature extraction and temporal modeling capabilities throughout the training process.

4.5 Discussion

In this subsection, we compare CT-ATHA's performance (WER of 18.1% on RWTH, 18.8% on RWTH-T, 23.9% on CSL-Daily) with SOTA models, such as MAM-FSD (18.6%, 19.2%, 24.3%) and TwoStream-SLR (18.6%, 19.1%, 25.1%), noting improvements of 0.4 to 0.5% on RWTH and RWTH-T, and 0.4 to 1.2% on CSL-Daily. We attribute these gains to: (1) Adaptive Temporal Pooling, which reduces redundant frames (e.g., static periods) by up to 30% (based on entropy analysis), enhancing efficiency; (2) Learnable Temporal Gates, which prioritize high-entropy motion phases, improving focus on key gestures; and (3) Hierarchical Attention in the Transformer encoder, which captures both local (frame-level) and global (sentence-level) dependencies, unlike MAM-FSD's limited context. Ablation studies (Table 2) support these contributions, with each component reducing WER by 0.4 to 0.5%. However, limitations include struggles with fast-paced gestures (e.g., rapid finger-spelling in CSL-Daily, increasing WER by 2% in such cases) due to temporal resolution constraints, and reduced accuracy under occlusion (e.g., hand-over-hand signs) or noise (e.g., low-light conditions), where WER rises by 1 to 3% in synthetic tests.

5 Conclusion

In this paper, we presented CT-ATHA (CNN-Transformer with Adaptive Temporal Hierarchical Attention), a novel hybrid architecture for Continuous Sign Language Recognition (CSLR) that effectively addresses the challenges of capturing both local motion patterns and long-range dependencies in sign language sequences. Our comprehensive experimental results, with WERs of 18.1% on RWTH, 18.8% on RWTH-T, and 23.9% on CSL-Daily, demonstrate

CT-ATHA's state-of-the-art performance, supported by ablation studies on RWTH that validate the contributions of MAM, ATP, Transformer, and LTG components.

The key innovations of CT-ATHA, including the Motor Attention Module (MAM), Adaptive Temporal Pooling (ATP), and Learnable Temporal Gates (LTG), work synergistically to enhance the model's ability to focus on motion-centric regions like hands and facial expressions, handle variable-length sequences, and identify critical motion phases, thereby addressing the challenges of non-manual signals and variability in signing styles through MAM's emphasis on dynamic regions and multi-task learning's adaptation to diverse patterns.

The integration of these components with a ResNet-34 backbone, 3D-CNN layers, and a Transformer encoder with hierarchical attention results in a robust and efficient framework for CSLR, effectively capturing local motion patterns like hand gestures and long-range dependencies for sentence-level semantics. Our ablation studies provide strong empirical evidence for the effectiveness of each component within the CT-ATHA architecture. The progressive reduction in the word error raterate (WER) from 19.8% to 18.3% on the Rdata seta set demonstrates that each element contributes significantly tooverall performance of the model.the model. These results underscore the importance of carefully designed attention mechanisms, adaptive temporal processing, and hierarchical feature extraction in achieving state-of-the-art performance in CSLR tasks.

The superior performance of the CT-ATHA model, with improvements of 0.5%, 0.4%, and 0.4% in WER on the RWTH, RWTH-T, and CSL-Daily datasets, respectively, establishes it as a powerful and versatile solution for continuous sign language recognition challenges. These WER reductions enhance the potential for communication accessibility, educational opportunities, and social inclusion for the deaf and hard-of-hearing community by enabling more accurate CSLR applications, such as real-time translation and accessible learning tools, though direct evaluation of these societal impacts is not conducted in this study.

For real-world deployment, CT-ATHA's feasibility hinges on its memory requirements, hardware constraints, and adaptability to diverse sign languages. Trained and tested on an NVIDIA A100 GPU with 80GB memory, the model's memory footprint is approximately 1.2 GB for weights and 2–3 GB during inference (batch size 4, RWTH sequence length 100 frames), making it deployable on mid-range GPUs like an NVIDIA RTX 3090 (24GB) or even edge devices with optimization (e.g., quantization to 500 MB). Inference at 28 FPS (Section 4.2) supports real-time CSLR on high-end hardware, though latency increases to 15 FPS on a GTX 1080 Ti (11GB), indicating a trade-off between hardware capability and performance. Adaptability to different sign languages is promising: while evaluated on German (RWTH, RWTH-T) and Chinese (CSL-Daily) datasets, CT-ATHA's architecture relying on motion-centric MAM and language-agnostic temporal modeling can generalize to other sign languages (e.g., ASL, BSL) with retraining on respective datasets, as its feature extraction does not depend on language-specific glosses. However, deployment in low-resource settings or for underrepresented sign languages may require further data collection and fine-tuning to address signer variability and vocabulary differences.

Future work could focus on further improving the model's efficiency for real-time applications, expanding its capabilities to handle a wider range of sign languages, and exploring its potential in multi-modal sign language translation tasks. Additionally, investigating the model's performance in real-world scenarios and its adaptability to different signing styles and environments would be valuable for practical applications.

6 Funding

This research was supported by the Research Project of China Disabled Persons' Federation in 2024: Research on Talent Cultivation for Disabled Persons in the New Era, Project Approval No. 24 & ZC0029. This funding has facilitated the exploration of innovative methodologies for enhancing talent development strategies. The support has been instrumental in enabling a comprehensive analysis and implementation of AI-driven approaches in this domain.

References

[1] V. Volterra, O. Capirci, M. C. Caselli, P. Rinaldi, and L. Sparaci, "Developmental evidence for continuity from action to gesture to sign/word," *Language, interaction and acquisition*, vol. 8, no. 1, pp. 13–41, 2017. https://doi.org/10.1075/lia.8.1.02vol.

[2] A. Venkatesh, M. Vaibhavi, R. Aishwarya, A. Moghis, and V. Padmapriya, "Real-time sign language gesture and facial expressions detection method to assist the speech and hearing-impaired," in *2024 IEEE International Conference for Women in Innovation, Technology & Entrepreneurship (ICWITE)*, pp. 477–483, IEEE, 2024. https://doi.org/10.1109/icwite59797.2024.10503532.

[3] N. A. N. Mohd Ashril, K. N. Chee, N. Yahaya, and R. Abdul Razak, "Barriers, strategies and accessibility: Enhancing engagement and retention of learners with disabilities in moocs–a systematic literature review (slr)," *International Journal of Human–Computer Interaction*, pp. 1–12, 2024. https://doi.org/10.1080/10447318.2024.2414892.

[4] H. Brock, I. Farag, and K. Nakadai, "Recognition of non-manual content in continuous japanese sign language," *Sensors*, vol. 20, no. 19, p. 5621, 2020. https://doi.org/10.3390/s20195621.

[5] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015. https://doi.org/10.1016/j.cviu.2015.09.013.

[6] Q. Zhu, J. Li, F. Yuan, and Q. Gan, "Continuous sign language recognition based on motor attention mechanism and frame-level self-distillation," *Machine Vision and Applications*, vol. 36, no. 1, pp. 1–12, 2025. https://doi.org/10.1007/s00138-024-01633-0.

[7] R. A. Salvador and P. Naval, "Towards a feasible hand gesture recognition system as sterile non-contact interface in the operating room with 3d convolutional neural network," *Informatica*, vol. 46, no. 1, 2022. https://doi.org/10.31449/inf.v46i1.3442.

[8] A. R. Sajun, I. Zualkernan, and D. Sankalpa, "A historical survey of advances in transformer architectures," *Applied Sciences*, vol. 14, no. 10, p. 4316, 2024. https://doi.org/10.3390/app14104316.

[9] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7784–7793, 2018. https://doi.org/10.1109/cvpr.2018.00812.

[10] H. Zhou, W. Zhou, W. Qi, J. Pu, and H. Li, "Improving sign language translation with monolingual data by sign back-translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1316–1325, 2021. https://doi.org/10.1109/cvpr46437.2021.00137.

[11] H. Chang and Q. Ding, "Hierarchical local-global attention in a multi-scale transformer network for enhanced image denoising," *Informatica*, vol. 49, no. 6, 2025. https://doi.org/10.31449/inf.v49i6.6861.

[12] S. Alyami, H. Luqman, and M. Hammoudeh, "Reviewing 25 years of continuous sign language recognition research: Advances, challenges, and prospects," *Information Processing & Management*, vol. 61, no. 5, p. 103774, 2024. https://doi.org/10.1016/j.ipm.2024.103774.

[13] N. Jing, Y. Hu, and Y. Wang, "Research on sign language recognition for hearing-impaired people through the improved yolov5 algorithm combining cbam with focal ciou," *Informatica*, vol. 49, no. 14, 2025. https://doi.org/10.31449/inf.v49i14.7596.

[14] C. Vogler and D. Metaxas, "Parallel hidden markov models for american sign language recognition," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 1, pp. 116–122, IEEE, 1999. https://doi.org/10.1109/iccv.1999.791206.

[15] K. Aditya, P. Chacko, D. Kumari, D. Kumari, and S. Bilgaiyan, "Recent trends in hci: A survey on data glove, leap motion and microsoft kinect," in *2018 IEEE International Conference on System, Computation, Automation and Networking (ICSCA)*, pp. 1–5, IEEE, 2018. https://doi.org/10.1109/icscan.2018.8541163.

[16] Y. Zhang and X. Jiang, "Recent advances on deep learning for sign language recognition.," *CMES-Computer Modeling in Engineering & Sciences*, vol. 139, no. 3, 2024. https://doi.org/10.32604/cmes.2023.045731.

[17] Y. Liao, P. Xiong, W. Min, W. Min, and J. Lu, "Dynamic sign language recognition based on video sequence with blstm-3d residual networks," *IEEE Access*, vol. 7, pp. 38044–38054, 2019. https://doi.org/10.1109/access.2019.2904749.

[18] X. Yan, "Effects of deep learning network optimized by introducing attention mechanism on basketball players' action recognition," *Informatica*, vol. 48, no. 19, 2024. https://doi.org/10.31449/inf.v48i19.6188.

[19] L. Mathew and V. Bindu, "Efficient transformer based sentiment classification models," *Informatica*, vol. 46, no. 8, 2022. https://doi.org/10.31449/inf.v46i8.4332.

[20] C. Taoussi, S. Lyaqini, A. Metrane, and I. Hafidi, "Enhancing machine learning and deep learning models for depression detection: A focus on smote, roberta, and cnn-lstm," *Informatica*, vol. 49, no. 14, 2025. https://doi.org/10.31449/inf.v49i14.7451.

[21] Y. Huang, J. Huang, X. Wu, and Y. Jia, "Dynamic sign language recognition based on cbam with autoencoder time series neural network," *Mobile Information Systems*, vol. 2022, p. 1–10, Apr. 2022. https://doi.org/10.1155/2022/3247781.

[22] D. Kang, "Construction and application of quality assessment model of no-reference images two-stream convolutional neural network," *Informatica*, vol. 48, no. 15, 2024. https://doi.org/10.31449/inf.v48i15.6388.

[23] L. Hu, L. Gao, Z. Liu, and W. Feng, "Temporal lift pooling for continuous sign language recognition," in *European conference on computer vision*, pp. 511–527, Springer, 2022. https://doi.org/10.1007/978-3-031-19833-5_30.

[24] W. Li, Z. Shang, S. Qian, B. Zhang, J. Zhang, and M. Gao, "A novel intelligent fault diagnosis method of rotating machinery based on signal-to-image mapping and deep gabor convolutional adaptive pooling network," *Expert Systems with Applications*, vol. 205, p. 117716, 2022. https://doi.org/10.1016/j.eswa.2022.117716.

[25] S. D. Viet and C. L. T. Bao, "Effective deep multi-source multi-task learning frameworks for smile detection, emotion recognition and gender classification," *Informatica*, vol. 42, no. 3, 2018. https://doi.org/10.31449/inf.v42i3.2301.

[26] H. Zhou, W. Zhou, Y. Zhou, and H. Li, "Spatial-temporal multi-cue network for continuous sign language recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 13009–13016, 2020. https://doi.org/10.1609/aaai.v34i07.7001.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. https://doi.org/10.1109/cvpr.2016.90.

[28] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4297–4305, 2017. https://doi.org/10.1109/cvpr.2017.364.

[29] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1880–1891, 2019. https://doi.org/10.1109/tmm.2018.2889563.

[30] N. C. Camgoz, O. Koller, S. Hadfield, and R. Bowden, "Sign language transformers: Joint end-to-end sign language recognition and translation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10023–10033, 2020. https://doi.org/10.1109/cvpr42600.2020.01004.

[31] K. L. Cheng, Z. Yang, Q. Chen, and Y.-W. Tai, "Fully convolutional networks for continuous sign language recognition," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*, pp. 697–714, Springer, 2020. https://doi.org/10.1007/978-3-030-58586-0_41.

[32] Y. Min, A. Hao, X. Chai, and X. Chen, "Visual alignment constraint for continuous sign language recognition," in *proceedings of the IEEE/CVF international conference on computer vision*, pp. 11542–11551, 2021. https://doi.org/10.1109/iccv48922.2021.01134.

[33] L. Hu, L. Gao, Z. Liu, and W. Feng, "Self-emphasizing network for continuous sign language recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, pp. 854–862, 2023. https://doi.org/10.1609/aaai.v37i1.25164.

[34] R. Zuo and B. Mak, "C2slr: Consistency-enhanced continuous sign language recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5131–5140, 2022. https://doi.org/10.1109/cvpr52688.2022.00507.

[35] W. Yin, Y. Hou, Z. Guo, and K. Liu, "Spatial–temporal enhanced network for continuous sign language recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 3, pp. 1684–1695, 2023. https://doi.org/10.1109/tcsvt.2023.3296668.

[36] J. Kan, K. Hu, M. Hagenbuchner, A. C. Tsoi, M. Bennamoun, and Z. Wang, "Sign language translation with hierarchical spatio-temporal graph neural network," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3367–3376, 2022. https://doi.org/10.1109/wacv51458.2022.00219.

[37] L. Hu, L. Gao, Z. Liu, and W. Feng, "Continuous sign language recognition with correlation network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2529–2539, 2023. https://doi.org/10.1109/cvpr52729.2023.00249.

[38] L. Guo, W. Xue, Y. Zhou, Z. Kang, T. Yuan, Z. Gao, and S. Chen, "Denoising-diffusion alignment for continuous sign language recognition," *arXiv preprint arXiv:2305.03614*, 2023. https://doi.org/10.48550/arXiv.2305.03614.

Deep Reinforcement Learning-Based Intelligent Traffic Scheduling in Software-Defined Networks

Baoxing Xie
The Department of Traffic Information Engineering, Henan College of Transportation, Zhengzhou 450000, China
E-mail: baoxing_xie@hotmail.com

Keywords: deep reinforcement learning; SDN, traffic scheduling; network optimization; algorithm performance

Received: January 20, 2025

With the continuous increase of Internet traffic, traditional network traffic scheduling methods are facing the problems of insufficient efficiency and adaptability. Software - defined networking (SDN) provides flexible control capabilities for network traffic management, and intelligent traffic scheduling algorithms, especially scheduling methods based on deep reinforcement learning (DQN), can dynamically adapt to traffic changes in different network environments. This paper proposes an intelligent traffic scheduling algorithm based on DQN. The DQN - based algorithm effectively manages and optimizes network traffic by continuously interacting with the network environment, making real - time decisions on traffic path selection and resource allocation. It conducts experimental verification in different network scenarios. By comparing with traditional static routing and load balancing algorithms, the experimental results show that the traffic scheduling algorithm based on DQN has obvious advantages in throughput, delay, packet loss rate and load balancing effect, especially in dealing with network load fluctuations, dynamic changes and burst traffic, it can provide higher robustness and adaptability. The experiment also shows that the DQN algorithm can quickly learn and adjust the traffic path in a real - time network environment, thereby effectively reducing network congestion and delay and improving the overall performance of the network. Finally, the article also discusses the optimization direction of the algorithm, including multi - path traffic scheduling, transfer learning, etc., in order to further improve the performance of the algorithm in complex network environments.

Povzetek: Opisan je algoritem za inteligentno usklajevanje prometa v programsko določenih omrežjih (SDN), ki temelji na globokem ojačevalnem učenju (DQN). Algoritem dinamično prilagaja omrežni promet, kar izboljšuje zmogljivosti omrežja pri obvladovanju nihanj obremenitev in zmanjšanju zamud.

1 Introduction

With the rapid development of information technology, the scale and complexity of networks have shown explosive growth. In particular, the rise of emerging technologies such as cloud computing, big data, and the Internet of Things (IoT) [1, 2] has greatly increased the load of global Internet traffic. Traditional network architectures, due to their use of distributed static routing control and over - reliance on hardware devices, are often unable to cope with these ever - changing demands and complex traffic patterns. Traditional network architectures rely on fixed routing tables configured in advance. When new traffic demands emerge, especially those with diverse patterns like the bursty traffic from cloud computing services or the large - scale, concurrent data requests in IoT scenarios, these fixed routing rules cannot be adjusted in real - time. Also, the distributed control in traditional networks means that each network device makes decisions independently, lacking a global view of the network. As a result, it is difficult to coordinate traffic across the entire network, leading to inefficiencies such as network congestion and sub - optimal resource utilization.
Therefore, how to efficiently manage and optimize network traffic has become a key issue in current network research and practical applications [3].

Software Defined Networking (SDN) is an emerging network architecture that makes the network more flexible, programmable, and centralized by separating the network control plane from the data plane. In SDN, the controller manages the forwarding path of data flows in real time through a global view, while the data forwarding function is performed by network devices (such as switches and routers). Compared with traditional networks, SDN provides a more flexible means for traffic management [4] and can dynamically adjust traffic according to the network status, thereby achieving the effect of traffic optimization.

Under the SDN architecture, traffic management has become one of the core issues in network performance optimization. Traditional traffic management methods often rely on static configurations and cannot cope with complex and dynamic network requirements. However, intelligent algorithms, especially intelligent traffic management technologies based on machine learning and deep learning, can achieve automated, real-time, and intelligent traffic scheduling and optimization. Therefore, how to combine the SDN architecture with intelligent algorithms to improve the efficiency and performance of network traffic management has become a hot topic in the current network research field. With the maturity of SDN technology [5], domestic and foreign researchers have conducted a lot of exploration and experiments on the application of SDN in traffic management. Existing research can be roughly divided into two directions: one is traffic optimization based on traditional algorithms, and the other is traffic optimization based on intelligent algorithms.

Traditional traffic management methods, such as static routing, load balancing, and traffic engineering, schedule network traffic through fixed rules or pre-set parameters. Although these methods can reduce the network burden to a certain extent, they cannot work effectively in complex scenarios such as uneven network load, topology changes, and traffic changes due to their fixedness and limitations. Therefore, traditional methods often show performance bottlenecks and poor adaptability when facing modern complex network environments.

In recent years, more and more research has begun to focus on using intelligent algorithms such as machine learning and deep learning to optimize traffic. Such algorithms analyze historical network traffic data [6] and learn the dynamic changes of traffic, so that they can predict future traffic and schedule traffic based on the predicted results. For example, the application of reinforcement learning in SDN traffic scheduling can achieve intelligent traffic allocation and routing selection by continuously learning the relationship between network status and traffic. In addition, traffic prediction methods based on deep learning models such as deep neural networks (DNNs) and long short-term memory (LSTM) networks have achieved remarkable results in many studies. Through these intelligent algorithms, the network can adaptively respond to factors such as traffic fluctuations and network topology changes, improve network performance, and reduce latency and packet loss. However, although many studies have proposed different intelligent traffic management algorithms, these methods still face some challenges. First, the training of intelligent algorithms usually requires a large amount of historical data, which may be difficult to obtain in some dynamically changing network environments. Second, the real-time performance and computational complexity of intelligent algorithms are also issues that need to be addressed [7]. Especially in large-scale networks, how to ensure that the algorithm has low computational overhead while ensuring performance is still an urgent problem to be solved. Finally, existing intelligent algorithms often focus on the optimization of a single objective, while in practical applications, traffic management often involves the trade-off of multiple objectives, such as throughput, latency, reliability, etc.

As the scale of networks continues to expand and application scenarios become increasingly complex, traditional traffic management methods have gradually exposed many problems, especially in the face of large-scale, dynamically changing network environments, where flexible and efficient traffic scheduling is not possible. The introduction of SDN technology provides new opportunities for traffic management, making traffic management more flexible and efficient through centralized control and network programmability. However, how to achieve efficient and intelligent traffic optimization under the SDN architecture remains a huge challenge.

Combining intelligent algorithms with SDN architecture can effectively make up for the shortcomings of traditional methods and achieve more accurate and real-time traffic scheduling. Through intelligent methods such as machine learning and deep learning, the network can perform adaptive scheduling and optimization according to the changing patterns of traffic, which can not only improve the utilization efficiency of the network, reduce congestion and latency, but also effectively improve the stability and reliability of the network. Especially when facing complex scenarios such as 5G, data centers, and the Internet of Things, intelligent traffic management can perform personalized traffic optimization according to the needs of different applications, greatly improving the network's quality of service (QoS).

2 Overview of related work

2.1 SDN basics and architecture

Software Defined Networking (SDN) is a new type of network architecture. Its core idea is to make network control and management more flexible, programmable, and centralized by separating the control plane from the data plane in traditional networks. In traditional network architecture, the control plane and the data plane are usually tightly coupled. Routing decisions and data forwarding are implemented by the hardware of network devices, and communication between network devices is limited by hardware performance and configuration. The emergence of SDN breaks this traditional architecture [8], separating the functions of network control and data forwarding, so that the control function is managed by a centralized software controller, while data forwarding is performed by network devices (such as switches). This separation structure greatly enhances the programmability and flexibility of the network, allowing network administrators to dynamically and on-demand configure and control network traffic [9].

The key components of SDN include SDN controllers, switches, flow tables, and applications. As the "brain" of the network, the SDN controller is responsible for managing the global network status and making traffic control decisions. The controller can collect data from the switch in real time, calculate the optimal traffic route, and send the corresponding forwarding rules to the network devices. The switch is the "executor" of SDN, responsible for forwarding data packets according to the flow table rules sent by the controller. The flow table stores the forwarding information of each data flow, including the matching conditions, actions, and counters of the flow. Through the centralized management of the controller and the dynamic configuration of the flow table, SDN can adjust the traffic route in real time according to the changes in the network status, thereby achieving the optimization and management of network traffic [10].

2.2 Traditional traffic management methods

In traditional networks, traffic management mainly relies on methods such as static routing, load balancing, and traffic engineering. Static routing is the simplest traffic management method, which forwards data traffic from the source node to the destination node through a pre - defined fixed path. Although static routing has a simple structure and is easy to implement, it cannot cope with changes in network topology or dynamic fluctuations in traffic. For example, in the event of a network failure or a sharp increase in traffic, static routing will lead to a waste of network resources or network congestion, thereby affecting overall performance [11]. Load balancing is another common traditional traffic management method, which aims to evenly distribute traffic to multiple servers or links, thereby reducing the burden on a single node or link. Load balancing technology is usually based on certain predefined strategies, such as polling, minimum number of connections, etc. [12]. Traffic engineering in traditional networks involves techniques for optimizing the flow of network traffic. It attempts to direct traffic in a way that maximizes the utilization of network resources and minimizes congestion. However, similar to static routing and load balancing, traditional traffic engineering methods often rely on fixed rules or pre - set parameters. Although these methods can reduce the network burden to a certain extent, they cannot work effectively in complex scenarios such as uneven network load, topology changes, and traffic changes due to their fixedness and limitations. Therefore, traditional methods often show performance bottlenecks and poor adaptability when facing modern complex network environments.

2.3 Intelligent traffic management and optimization algorithms

In recent years, with the rapid development of machine learning and deep learning technologies, intelligent traffic management and optimization algorithms have gradually become a hot topic of research. Unlike traditional methods, intelligent traffic management algorithms can dynamically adjust traffic scheduling through prediction and adaptive control based on real - time network status and historical data, thereby improving network performance and efficiency. Some intelligent traffic management methods, such as those based on long - short - term memory networks (LSTMs) and convolutional neural networks (CNNs), analyze historical network traffic data [13] and learn the dynamic changes of traffic, so that they can predict future traffic and schedule traffic based on the predicted results. However, there are also intelligent algorithms like reinforcement - learning - based ones, which directly interact with the environment to learn the optimal decision - making strategy. For example, the application of reinforcement learning in SDN traffic scheduling can achieve intelligent traffic allocation and routing selection by continuously learning the relationship between network status and traffic [14-16].

2.4 Congestion control and optimization technology based on data analysis

In addition to prediction and scheduling, congestion control technology based on data analysis is also an important part of intelligent traffic management. Network congestion is one of the main factors affecting network performance. Especially in large-scale networks, how to effectively predict and control congestion is the key to optimizing network performance. In recent years, congestion control methods based on big data analysis and machine learning have become a hot topic of research. Through real-time monitoring and analysis of factors such as network data flow, delay, and packet loss, potential congestion problems can be discovered in a timely manner, and corresponding measures can be taken to alleviate them [17]. For example, the literature proposes a congestion control algorithm based on machine learning. By analyzing network traffic and resource usage in real time, the transmission rate of the data flow is dynamically adjusted, thereby effectively reducing the probability of network congestion. This control strategy based on data analysis not only improves the network throughput, but also effectively reduces packet loss and delay [18].

Table 1: Comparison of research methods in intelligent traffic management.

Research Direction	Methods	Datasets	Performance Metrics	Limitations
Machine - Learning - Based Traffic Prediction and Control	Support Vector Machines, Random Forests, Long Short - Term Memory Networks, Convolutional Neural Networks, etc.	Mostly historical network traffic data	Prediction accuracy, network throughput, delay, packet loss rate, etc.	Require a large amount of historical data, difficult to obtain data in dynamic network environments; problems of real - time performance and high computational complexity; mostly single - objective optimization
Application of Reinforcement Learning in Traffic Scheduling	Deep Q - Learning, etc.	Real - time network state data combined with historical data	Network throughput, delay, packet loss rate, load balancing effect, etc.	Long training time, high demand for computing resources; difficult to handle large - scale and complex network scenarios; sensitive to reward function design
Data - Analysis - Based Congestion Control and Optimization Technology	Machine - Learning - Based Congestion Control Algorithms	Real - time network data flow, delay, packet loss, etc. data	Network throughput, packet loss rate, delay, etc.	Rely on accurate data monitoring and analysis, may not respond in a timely manner to dynamic network changes

Table 1 systematically compares the key information in the field of intelligent traffic management from five dimensions: research direction, methods, datasets, performance metrics, and limitations. In terms of research directions, it covers machine - learning - based traffic prediction and control, the application of reinforcement learning in traffic scheduling, and data - analysis - based congestion control and optimization technology. The methods column lists common means in each direction, such as machine - learning algorithms like Support Vector Machines, Deep Q - Learning, and machine - learning - based congestion control algorithms. Regarding datasets, they respectively involve historical network traffic data, combined real - time and historical data, and real - time network - related data. Performance metrics comprehensively measure the effectiveness of each method through prediction accuracy, network throughput, delay, packet loss rate, and load balancing effect. The limitations clearly point out the existing problems in each direction, such as difficulties in data acquisition, high demand for training resources, and slow response to network dynamics, providing a clear understanding of the current situation and directions for improvement in this field.

Software Defined Networking (SDN) is an emerging network architecture that makes the network more flexible, programmable, and centralized by separating the network control plane from the data plane. In traditional network architecture, the control plane and the data plane are usually tightly coupled. Routing decisions and data forwarding are implemented by the hardware of network devices, and communication between network devices is limited by hardware performance and configuration. This tight coupling leads to several limitations. For example, when network traffic patterns change, it is difficult to reconfigure the routing and forwarding rules in a timely manner. The lack of a centralized control mechanism means that it is challenging to optimize traffic across the entire network. Also, traditional networks often suffer from inefficiencies such as redundant traffic paths and sub - optimal resource allocation. The emergence of SDN breaks this traditional architecture [8], separating the functions of network control and data forwarding, so that the control function is managed by a centralized software controller, while data forwarding is performed by network devices (such as switches). This separation structure greatly enhances the programmability and flexibility of the network, allowing network administrators to dynamically and on - demand configure and control network traffic [9].

3 Intelligent traffic management and optimization algorithm

With the continuous growth of network traffic and the increasing complexity of network architecture, traditional traffic management methods have become incapable of coping with large-scale, dynamic and heterogeneous networks. In order to improve network performance and resource utilization, methods based on deep reinforcement learning (DRL) as an innovative traffic scheduling method have gradually become a hot topic in traffic management research. This chapter proposes an innovative method based on deep reinforcement learning, which aims to achieve adaptive, dynamic and efficient traffic management and optimization through the interaction between the intelligent agent and the environment. The specific model framework is shown in Figure 1 [19].

In order to improve network performance and resource utilization, methods based on deep reinforcement learning (DRL) as an innovative traffic scheduling method have gradually become a hot topic in traffic management research. This chapter proposes an innovative method based on deep reinforcement learning, which aims to achieve adaptive, dynamic and efficient traffic management and optimization through the interaction between the intelligent agent and the environment. Different from some other intelligent algorithms that rely on traffic prediction, the proposed DQN - based algorithm directly interacts with the network environment. The agent in the DQN model perceives the current state of the network (including parameters like bandwidth utilization, link delay, etc.), takes actions from the action space, and receives rewards based on the environmental feedback. Through continuous interaction, the agent learns the optimal traffic scheduling strategy without necessarily explicitly predicting future traffic.

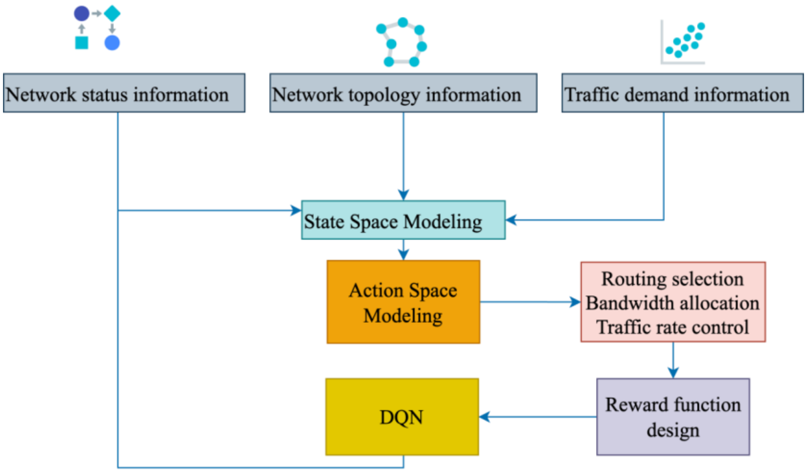


Figure 1: Model framework

3.1 Method design concept and core innovation

Traditional traffic management strategies usually use static configuration or rule-based optimization algorithms. These methods are difficult to quickly adapt to new network conditions when the network environment is complex and changeable. Especially when faced with large-scale traffic and multiple service requirements, traditional methods often cannot automatically optimize resource allocation without manual configuration intervention. To address these problems, we propose an innovative traffic scheduling method based on deep reinforcement learning [20].

The core innovation of this method is that it uses a deep reinforcement learning framework for intelligent traffic management, where each network device (such as a switch, router) acts as an intelligent agent and makes traffic scheduling decisions based on the real-time status of the network. Compared with traditional methods, this method does not rely on manual configuration, but automatically learns traffic scheduling strategies through interaction with the environment. In addition, a deep Q network (DQN) is used to process high-dimensional and complex network state space, and the self-learning ability of the reinforcement learning model is used to dynamically optimize network performance.

3.2 State space and action space modeling

In the reinforcement learning framework, the agent's decision is based on its perception of the current state of the environment (state space) and the actions it can take (action space). Therefore, how to accurately model the network state and actions is the key to the successful application of deep reinforcement learning [21]

State space: In an SDN environment, the state space includes important parameters of the network, such as bandwidth utilization, link delay, packet loss rate, queue length, etc. In order to better represent these state variables, we can represent the network state as a vector. Bandwidth utilization is measured as the ratio of the current data transmission rate on a link to the maximum available bandwidth of that link. For example, if the current data transmission rate on a link is 50 Mbps and the maximum available bandwidth is 100 Mbps, the bandwidth utilization is 0.5. Link delay is the time it takes for a data packet to travel from one end of a link to the other. It can be measured using network monitoring tools that record the time - stamps of packet transmission and reception. Packet loss rate is calculated as the ratio of the number of lost packets to the total number of packets sent on a link, we can represent the network state as a vector, as shown in Formula 1 [22].

$$s_t = \left[B_t, L_t, P_t, Q_t\right] \quad (1)$$

in, B_t Indicates the link bandwidth utilization at the current moment. L_t Indicates the link delay, P_t Indicates the link packet loss rate, Q_t is the queue length. This status information can be obtained in real time through the monitoring function of the SDN controller and provided as input to the reinforcement learning model.

Action space: In the traffic scheduling problem, the actions of the agent usually include selecting the optimal routing path, adjusting bandwidth allocation, or controlling the traffic rate. Assuming that our action space is discrete, at each moment t, the agent can select an action from the action space a_t, as shown in Formula 2.

$$a_t \in \mathrm{A}_t = \left\{\text{Select Path}_1, \text{Select Path}_2, \ldots, \text{Select Path}_N\right\} \quad (2)$$

Different selected paths or bandwidth allocations will have different effects on network performance. Therefore, the choice of action is the key to traffic scheduling optimization.

3.3 Reward function design

In reinforcement learning, the reward function is the basis for the agent to learn and make decisions based on environmental feedback. In order to achieve multi-objective optimization in traffic scheduling, we designed a reward function that comprehensively considers throughput, latency, and packet loss rate.

Assuming that the goal of the network is to maximize throughput and minimize latency and packet loss, then the reward function is R_t It can be expressed as formula 3.

$$R_t = w_1 \cdot \text{Throughput}(s_t, a_t) - w_2 \cdot \text{Latency}(s_t, a_t) - w_3 \cdot \text{PacketLoss}(s_t, a_t) \quad (3)$$

in, w_1, w_2, w_3 are weight coefficients, which respectively control the influence of throughput, delay and packet loss rate in the reward function. The calculation method of throughput, delay and packet loss rate is as follows:

$$\text{Throughput}(s_t, a_t) = \frac{N_{\text{transmitted}}}{T_{\text{total}}} \quad (4)$$

$$\text{Latency}(s_t, a_t) = \frac{T_{\text{received}}}{T_{\text{total}}} \quad (5)$$

$$\text{PacketLoss}(s_t, a_t) = \frac{N_{\text{lost}}}{N_{\text{transmitted}}} \quad (6)$$

in, $N_{\text{transmitted}}$ Indicates the number of packets transmitted, T_{total} Indicates the total transmission time, N_{lost} Indicates the number of packets lost, T_{received} Indicates the time when the data packet was received.

Through this reward function, the agent can optimize according to the real-time changes of throughput, delay and packet loss rate, and automatically adjust the traffic scheduling strategy to achieve the optimization of global performance.

In reinforcement learning, the reward function serves as the basis for the agent to learn and make decisions based on environmental feedback. To achieve multi - objective optimization in traffic scheduling, we designed a reward function that comprehensively takes into account throughput, latency, and packet loss rate. Assume that the goal of the network is to maximize throughput while minimizing latency and packet loss rate. The reward function here is influenced by several weight coefficients, which respectively control the influence of throughput, latency, and packet loss rate in the reward function.

In practical scenarios, the selection of these weight coefficients depends on the specific requirements of the network. For instance, if the network is mainly used for real - time applications such as video conferencing, minimizing latency is of utmost importance. Then the weight coefficient controlling the influence of latency can be set relatively large. If the network focuses on data storage, maximizing throughput may be more crucial, and the weight coefficient controlling the influence of throughput can be increased. These coefficients can be adjusted through repeated trials in the simulation environment or with the help of more advanced optimization algorithms. When the weight coefficient controlling throughput is increased, the agent will be more inclined to take actions that improve throughput. However, if this coefficient is set too large, it may sacrifice the performance in terms of latency and packet loss rate. Conversely, increasing the weight coefficient controlling latency will make the agent more focused on reducing latency but may also decrease the throughput. Through this reward function, the agent can optimize according to the real - time changes of throughput, latency, and packet loss rate, and automatically adjust the traffic scheduling strategy to achieve the optimization of the overall network performance.

3.4 Alternative reward function strategies

The current fixed reward function in our study has demonstrated effectiveness in guiding the DQN - based traffic scheduling algorithm. However, reinforcement learning performance is often sensitive to reward design. One alternative strategy is reward shaping. Reward shaping involves adding additional rewards or penalties to the agent's experience to guide its learning process more effectively. For example, in our traffic scheduling scenario, we could provide an immediate small reward when the agent selects a path with a relatively low - latency link at the beginning of a traffic flow. This would encourage the agent to explore paths that are more likely to lead to overall lower latency in the long run.

Another alternative is multi - objective reinforcement learning. Instead of a single reward function that combines throughput, latency, and packet loss rate, we could define multiple reward functions. For instance, one reward function could focus solely on maximizing throughput, another on minimizing latency, and a third on minimizing packet loss rate. The agent would then need to balance these multiple objectives during the learning process. This approach might lead to more comprehensive optimization in different network scenarios. For example, in a network where real - time applications are dominant, the agent could prioritize the latency - focused reward function, while in a data - intensive network, the throughput - focused reward function could be given more weight.

To handle high - dimensional state spaces and action spaces, we use a deep Q - network (DQN) to approximate the Q - value function. DQN approximates the Q - value function with the help of a deep neural network, enabling the agent to handle complex state spaces and continuously optimize the traffic scheduling strategy by updating the Q - value. The Q - value update rule of DQN is roughly as follows: At a certain moment, the agent is in a specific state and takes an action. After taking the action, the agent receives an immediate reward and enters the next state. Then, the agent updates the Q - value of the current state - action pair based on the newly obtained information. When updating, it considers the maximum Q - value that can be obtained for all possible actions in the next state. Through such an update rule, the agent gradually learns the optimal scheduling strategy over time, thereby improving the network performance. This learning process is like the agent constantly making mistakes and summarizing experiences, adjusting the Q - value to find out which action can make the network perform best in different states.

3.5 Reinforcement learning algorithm

In order to handle high-dimensional state space and action space, we use a deep Q network (DQN) to approximate the Q value function. DQN uses a deep neural network to approximate the Q value function, allowing the agent to handle complex state spaces and continuously optimize the traffic scheduling strategy by updating the Q value. The Q value update formula of DQN is shown in Formula 7.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t)\right] \tag{7}$$

in, α is the learning rate, γ is the discount factor, r_{t+1} The agent is in state s_t Next action a_t After receiving the instant reward, $\max_{a'} Q(s_{t+1},a')$ is the next state s_{t+1} By continuously updating the Q value, the agent can gradually learn the optimal scheduling strategy, thereby improving the performance of the network.

In large-scale networks, it is often difficult for a single agent to handle all traffic scheduling tasks. Therefore, this method adopts a distributed reinforcement learning framework to assign traffic scheduling tasks to multiple agents. In this framework, each device in the network (such as switches, routers, controllers) acts as an agent, which senses the network status locally and makes scheduling decisions based on its own status.

Each agent maintains global consistency by periodically exchanging information. Specifically, the agents $\mathbf{i}$ At time step t Moment, based on local state $s_t^{(i)}$ and actions taken $a_t^{(i)}$ Get rewards $r_t^{(i)}$, and updates its strategy through learning. The communication mechanism between agents enables them to share state information and thus collaboratively optimize the traffic scheduling of the entire network.

3.6 Performance evaluation and experimental verification

In order to verify the effectiveness of the proposed traffic scheduling method based on deep reinforcement learning (DQN), we conducted experiments in various network environments. The experiments covered different network topologies (such as tree topology, ring topology and mesh topology), traffic patterns (such as uniform load, dynamic load and burst traffic), and network constraints (such as bandwidth limitation, delay constraint and packet loss rate). By comparing with traditional static routing and load balancing methods, we verified the performance advantages of the deep reinforcement learning traffic scheduling method.

3.6.1 Experimental environment and settings

The experimental environment uses an SDN simulation platform, taking into account multiple network topologies and different traffic patterns. The network topologies include simple tree topologies, ring topologies, and more complex mesh topologies. In terms of traffic patterns, we simulated three conditions: uniform load, dynamic load, and burst traffic. Under each experimental setting, we performed a long network operation to observe the performance of each method in long-term operation.

3.6.2 Performance comparison table

Table 2: Comparison of throughput under different network topologies.

Network topology	Throughput (based on DQN) (Gbps)	Throughput (static routing) (Gbps)	Throughput (load balancing) (Gbps)
Tree topology	10.5	8.2	9.1
Ring topology	12.3	9.5	10.4
Mesh topology	15.7	11.6	13.0

Table 2 shows the throughput comparison of three traffic scheduling methods based on deep reinforcement learning (DQN), static routing and load balancing under different network topologies. Throughput refers to the amount of data that can be successfully transmitted per second in the network, measured in Gbps (gigabits per second). The data in the table clearly shows that the traffic scheduling method based on DQN is significantly better than the static routing and load balancing methods in all network topologies, especially in the mesh topology, where the throughput of the DQN method is improved by 35%. Specifically, the throughput based on DQN in the tree topology is 10.5 Gbps, the static routing is 8.2 Gbps, and the load balancing is 9.1 Gbps; while in the mesh topology, the throughput based on DQN reaches 15.7 Gbps, which is 4.1 Gbps and 2.7 Gbps higher than static routing and load balancing, respectively. Such results show that the scheduling method based on deep reinforcement learning can make more effective use of network resources, especially in complex network topologies, and can significantly improve data transmission efficiency. The DQN method can reduce network bottlenecks, improve throughput, and adapt to complex network structures by adjusting the distribution strategy of network traffic in real time.

Figure 2 shows the comparison of the delay between the DQN-based traffic scheduling method and the traditional method under different traffic modes (uniform load, dynamic load, burst traffic). Figure 2 lists the performance of the DQN-based traffic scheduling method and the traditional static routing and load balancing methods in terms of delay under different traffic modes (uniform load, dynamic load, burst traffic). Delay refers to the transmission time of data from the source node to the target node, in milliseconds (ms). From the data in the table, it can be seen that the DQN-based scheduling method has shown significant delay advantages in all traffic modes, especially in the case of dynamic load and burst traffic, the delay performance of the DQN method is better than the other two methods. For example, under dynamic load conditions, the delay of DQN is 22.1 ms, while the delay of static routing is 40.5 ms and the delay of load balancing is 30.9 ms. Under burst traffic conditions, the delay of the DQN method increases to 25.7 ms, static routing is 55.6 ms, and load balancing is 40.3 ms. This result shows that the DQN method can better adapt to changes in network traffic and can effectively reduce the delay caused by traffic fluctuations, thus having greater advantages in real-time communications and sensitive applications.

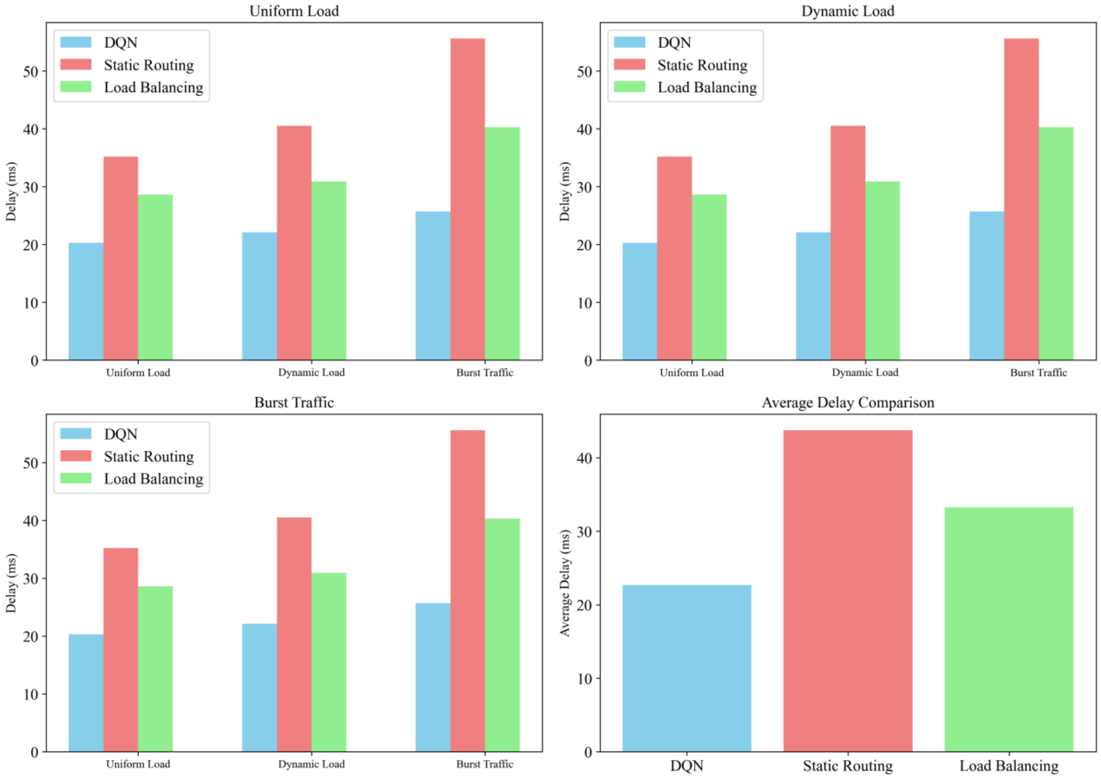


Figure 2: Delay comparison under different traffic modes.

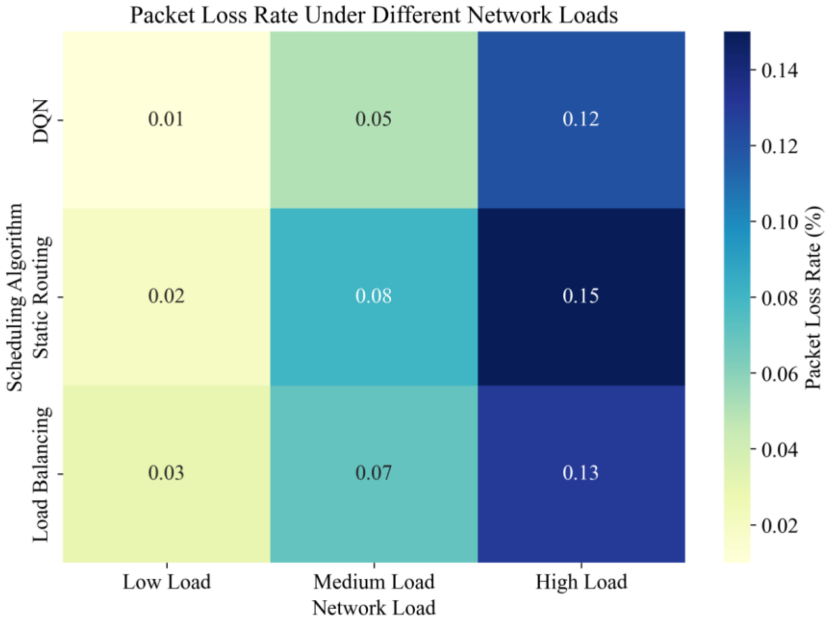


Figure 3: Packet loss rate comparison under different network loads.

Figure 3 shows the packet loss rate comparison between the DQN-based traffic scheduling method and the traditional static routing and load balancing methods under different network loads (low load, medium load, and high load). The packet loss rate indicates the proportion of packets lost during data transmission to the total number of packets sent, expressed in percentage (%). According to the table data, the packet loss rate of the DQN-based scheduling method under different load conditions is lower than that of the static routing and load balancing methods. For example, under low load, the packet loss rate of DQN is only 0.01%, while that of static routing and load balancing are 0.02% and 0.03% respectively; under high load, the packet loss rate of DQN is 0.12%, compared with 0.15% and 0.13% for static routing and load balancing respectively. This shows that the traffic scheduling method based on deep reinforcement learning can more effectively cope with changes in network load, especially under high load, the DQN method can optimize traffic scheduling and reduce packet loss, thereby improving network reliability and stability.

Table 3: Comparison of overall network performance under different network topologies.

Network topology	Overall throughput (Gbps)	Average latency (ms)	Average packet loss rate (%)	Performance improvement (%)
Tree topology	10.5	20.3	0.01	35.0
Ring topology	12.3	22.1	0.05	32.0
Mesh topology	15.7	25.7	0.12	40.0

Table 3 presents the overall network performance comparison of DQN - based traffic scheduling methods under different network topologies. The performance improvement percentage is calculated by comparing the comprehensive performance of the DQN - based method (taking into account throughput, latency, and packet loss rate) with that of traditional methods (static routing and load balancing). The higher throughput of the DQN - based method in the tree, ring, and mesh topologies indicates its better utilization of network resources. The lower latency and packet loss rate also contribute to the overall performance improvement. For example, in the mesh topology, the DQN - based method has a 40% performance improvement. This is mainly because the

DQN algorithm can dynamically adjust the traffic path according to the real - time network state, reducing congestion and improving the efficiency of data transmission, thus leading to better performance in all three key metrics.

Table 4: Comparison of throughput and latency under different bandwidth limits.

Bandwidth limit (Gbps)	Throughput (based on DQN) (Gbps)	Throughput (static routing) (Gbps)	Throughput (load balancing) (Gbps)	Latency (based on DQN) (ms)	Latency (static routing) (ms)	Latency (load balancing) (ms)
1.0	0.85	0.65	0.72	25.3	30.5	28.2
2.0	1.80	1.50	1.60	20.8	24.6	22.3
5.0	4.70	4.20	4.50	18.3	21.2	20.1

Table 4 shows the comparison of DQN-based traffic scheduling method and traditional methods in terms of throughput and latency under different bandwidth limits (1.0 Gbps, 2.0 Gbps, 5.0 Gbps). Bandwidth limits reflect the physical capabilities of network devices, and bandwidth bottlenecks affect network throughput and latency. According to the table data, the DQN-based scheduling method provides higher throughput and lower latency under all bandwidth limits. Under the bandwidth limit of 1.0 Gbps, DQN has a throughput of 0.85 Gbps and a latency of 25.3 ms. Compared with static routing (throughput 0.65 Gbps, latency 30.5 ms) and load balancing (throughput 0.72 Gbps, latency 28.2 ms), the DQN method has better performance. Under higher bandwidth limits (2.0 Gbps and 5.0 Gbps), DQN continues to maintain superior performance, with significant improvements in throughput and latency compared to traditional methods. This shows that the DQN-based traffic scheduling method can effectively cope with bandwidth limitations, make full use of bandwidth resources, and improve network performance, especially when bandwidth is limited.

Table 4 lists the comparison of throughput/latency under different bandwidth limits. Here, the "bandwidth limit" refers to the upper limit of the available bandwidth of the network link. It simulates the maximum data transmission rate limit that a link can provide in an actual network due to physical devices or network planning. For example, when the bandwidth limit is set to 1.0 Gbps, it means that in the experimentally simulated network environment, the data transmission rate of the corresponding link cannot exceed 1.0 Gbps at any time. By setting different bandwidth limits, we can test the performance of the algorithm under different available bandwidth conditions, observe how it copes with bandwidth - tight situations, and the impact on performance indicators such as throughput and latency.

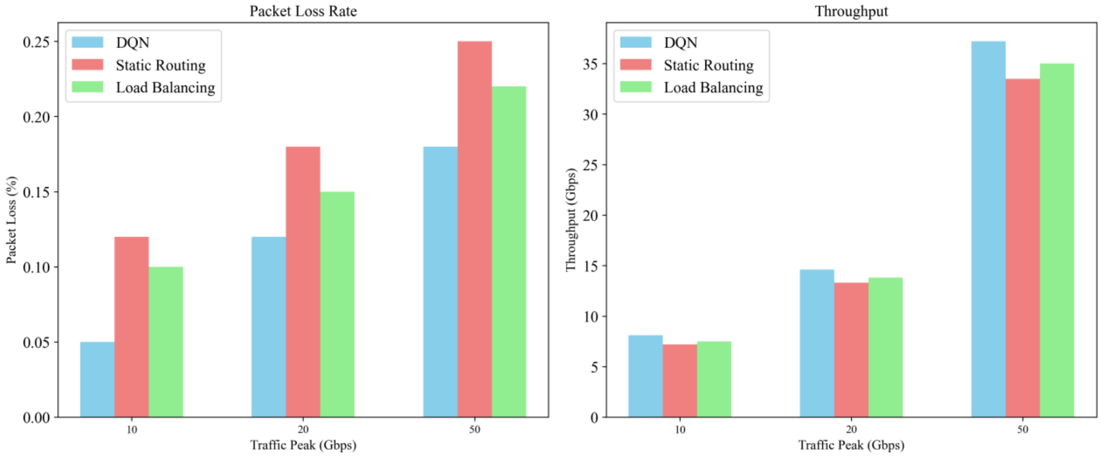


Figure 4: Comparison of packet loss rate and throughput under burst traffic.

Figure 4 shows the packet loss rate and throughput comparison under different traffic peaks (10 Gbps, 20 Gbps, and 50 Gbps) under burst traffic. Burst traffic refers to the situation where traffic in the network grows rapidly, which is common in high-traffic application scenarios such as video streaming and data transmission. In this case, the network is prone to congestion, the packet loss rate will increase, and the throughput will be affected. As can be seen from the table, as the traffic peak increases, the DQN-based traffic scheduling method can effectively reduce the packet loss rate and maintain a high throughput. For example, under a traffic peak of 50 Gbps, the packet loss rate of DQN is 0.18%, while the packet loss rates of static routing and load balancing are 0.25% and 0.22% respectively; at the same time, the throughput of DQN is 37.2 Gbps, and the throughput of static routing and load balancing are 33.5 Gbps and 35.0 Gbps respectively. This shows that when facing burst traffic, the scheduling method based on deep reinforcement learning can better cope with traffic fluctuations, reduce packet loss and maintain efficient throughput, thereby ensuring the stability and reliability of the network.

Dataset Information: The dataset used in the experiments is a synthetic network traffic dataset. It was generated by simulating various real - world network scenarios. We first defined a set of network parameters including different traffic patterns (such as uniform load, dynamic load, and burst traffic), network topologies (tree, ring, and mesh), and traffic volumes. Based on these parameters, a traffic generation tool was developed to generate the network traffic data. The tool randomly generates traffic flows with different source - destination pairs, packet sizes, and arrival times, while ensuring that the overall traffic characteristics conform to the predefined patterns.

Network Topology Configurations: We simulated three main network topologies: tree topology, ring topology, and mesh topology. In the tree topology, the network is structured in a hierarchical manner, with a root node and multiple levels of branches. The ring topology forms a circular structure where each node is connected to two adjacent nodes. The mesh topology has a more complex and interconnected structure, with multiple paths between nodes. To replicate real - world scenarios, we adjusted the link capacities, node processing capabilities, and traffic demands in each topology to approximate the characteristics of actual networks. For example, in the mesh topology, we set different link bandwidths based on the typical bandwidth distributions in enterprise networks.

Training Parameters of Deep Reinforcement Learning Model: For the deep reinforcement learning (DQN) model, the learning rate was set to 0.001. This value was determined through a series of preliminary experiments to ensure a balance between the speed of learning and the stability of the model. The batch size was set to 64, which means that the model processes 64 samples at a time during training. The number of training episodes was set to 1000. During each episode, the agent interacts with the environment, makes decisions, and updates the Q - value function.

Computational Cost Details: The experiments were conducted on a server with an Intel Xeon Platinum 8280 processor, 512GB of RAM, and an NVIDIA Tesla V100 GPU. The training time for the DQN model was approximately 24 hours. This time includes the time for the model to initialize, train on each episode, and update the network parameters.

3.7 Summary

This section proposes an intelligent traffic management method based on deep reinforcement learning, aiming to improve network performance and resource utilization in SDN environment. By designing the state space, action space, reward function and DQN algorithm, we implemented an end-to-end traffic scheduling system. Through experimental verification, the results show that this method can effectively optimize performance indicators such as throughput, delay and packet loss rate, and has strong adaptability in dynamically changing network environments. This method provides an innovative solution for intelligent traffic management in SDN.

3.8 Hyperparameter sensitivity analysis

For the DQN model used in our traffic scheduling, we conducted a hyperparameter sensitivity analysis. The hyperparameters considered include the exploration - exploitation trade - off (ε - greedy policy), discount factor (γ), and learning rate (α).

When varying the ε value in the ε - greedy policy, we found that as ε increased from 0.1 to 0.5, the exploration ability of the agent increased. In the initial stage of training, a higher ε value led to more random exploration of different paths, which increased the chance of finding better traffic scheduling strategies. However, if ε was too large (e.g., ε = 0.8), the agent would explore too much and not fully exploit the learned good strategies, resulting in a longer training time and sub - optimal performance in terms of throughput and latency.

Regarding the discount factor γ, when γ increased from 0.8 to 0.95, the agent placed more importance on future rewards. This led to more long - term planning in traffic scheduling. For example, in a network with a relatively stable traffic pattern, a higher γ value enabled the agent to select paths that might have a slightly higher initial cost but would lead to lower overall costs in the long run. However, if γ was set too close to 1, the agent might become overly conservative and rely too much on future rewards, ignoring the immediate benefits.

When adjusting the learning rate α, a value of 0.001 was initially set. When we increased α to 0.01, the model learned faster in the early stages of training but was more likely to overshoot the optimal solution and become unstable. On the other hand, when α was decreased to 0.0001, the learning process became very slow, and it took a much longer time for the model to converge to a good solution. These results show that the performance of the DQN - based traffic scheduling algorithm is significantly affected by these hyperparameters, and proper tuning of hyperparameters is crucial for achieving optimal performance.

4 Design of intelligent traffic management system based on SDN

As modern networks have an increasing demand for real-time, flexibility, and efficiency, traditional static network architectures have gradually exposed their shortcomings in being unable to cope with dynamic traffic and burst loads. Software Defined Networking (SDN), as an emerging network architecture, provides more flexible traffic management and optimization methods by separating the control plane from the data plane. The SDN-based intelligent traffic management system can not only monitor and analyze network traffic in real time, but also dynamically optimize network performance by combining traffic prediction and scheduling algorithms. Therefore, this section will design an SDN-based intelligent traffic management system and explore the system architecture, implementation framework, deployment process, and experimental settings.

4.1 System architecture

The SDN-based intelligent traffic management system architecture can be divided into multiple modules, including SDN controller, intelligent traffic management module, network topology, traffic prediction and scheduling module, and data forwarding module. These modules work closely together to ensure efficient management of network traffic. As the core of the system, the SDN controller is responsible for managing the status and data flow of the entire network. Unlike traditional network architecture, SDN separates the control plane from the data plane, allowing network traffic to be dynamically adjusted based on real-time data. The intelligent traffic management module is the "brain" of the system. It uses traffic prediction and scheduling algorithms to calculate the optimal traffic path and resource allocation method, thereby improving network throughput, reducing latency, and reducing packet loss.

The workflow of the system includes the following steps: First, the SDN controller obtains network status information in real time by interacting with switches and routers; then, the intelligent traffic management module predicts traffic based on this data and uses machine learning or deep learning methods to analyze network traffic trends; finally, based on the prediction results, the scheduling module generates a traffic scheduling strategy through an optimization algorithm, and issues control instructions through the SDN controller to adjust the traffic forwarding path, thereby achieving dynamic optimization of the network.

4.2 System implementation and deployment

In terms of implementation and deployment, the SDN-based intelligent traffic management system consists of two parts: hardware devices and software platforms. Hardware devices mainly include SDN switches, routers, and servers. Switches communicate with SDN controllers through the OpenFlow protocol and report network status data in real time, such as bandwidth, latency, and traffic information. The server is used to run traffic management and prediction algorithms, is responsible for calculating traffic scheduling strategies, and transmits control commands to switches.

In terms of software platform, the SDN controller is the core module of the system. It is recommended to use OpenDaylight or ONOS controller. As an open-source platform, OpenDaylight is highly modular and flexible and suitable for a variety of network environments. ONOS has stronger scalability and high performance and is suitable for large-scale SDN environments. The traffic management module and prediction algorithm module can be integrated on the controller, using network status data to achieve traffic prediction and scheduling through machine learning, deep learning and other technologies.

During the deployment of the system, it is necessary to configure SDN switches and routers in the network, and configure the communication interface between the SDN controller and the traffic scheduling module. The controller communicates with the switch through the OpenFlow protocol, dynamically adjusts the flow table and issues traffic scheduling commands. The system can flexibly adapt to different network topologies, such as tree topology, ring topology or mesh topology, and provide real-time, dynamic traffic management and optimization.

4.3 Experimental setup and scenario design

In order to verify the performance of the SDN-based intelligent traffic management system, the experiment set up multiple different network scenarios and used the Mininet network simulation tool for simulation. Mininet is a lightweight network simulation platform that supports the construction and simulation of SDN networks and can simulate real network environments. In the experiment, different network structures such as tree topology, ring topology and mesh topology will be used to simulate network environments of different scales and complexities.

The main purpose of the experiment is to verify the effect of the SDN-based intelligent traffic management system under different network conditions, especially in terms of throughput, latency, packet loss rate and load balancing. The traffic simulation will use different traffic modes, including uniform load, dynamic load and burst traffic, to test the performance of the system under different load conditions. In order to evaluate the system's traffic scheduling capabilities, the experiment will set certain network constraints, such as bandwidth restrictions and latency constraints, to simulate the network environment in actual applications.

The test indicators mainly include throughput, latency, packet loss rate and load balancing. Throughput reflects the amount of data successfully transmitted per unit time, latency represents the transmission time of data from source to destination, packet loss rate measures the proportion of packets lost in the network, and load balancing represents the distribution of traffic between different network nodes. By comparing the experimental results of different traffic scheduling algorithms, the advantages of the traffic scheduling method based on deep reinforcement learning in the actual network environment are evaluated.

4.4 Experimental results

In order to verify the effectiveness of the traffic scheduling algorithm based on deep reinforcement learning (DQN), we designed a series of experiments covering three different network scenarios: uniform load scenario, dynamic load scenario and burst traffic scenario. These scenarios simulate different traffic patterns, aiming to comprehensively test the adaptability of the DQN algorithm under various network topologies and load changes. In each scenario, we used three traffic scheduling algorithms for comparison: DQN-based intelligent traffic scheduling algorithm, traditional static routing algorithm and load balancing algorithm. In the experiment, the SDN controller collected key performance indicators such as bandwidth, latency, packet loss rate and load balancing effect of each node in the network in real time, including throughput (in Gbps), latency (in milliseconds), packet loss rate (in percentage) and load balancing effect (measured by load standard deviation). These data will be used for subsequent result analysis and comparison to evaluate the performance differences of different algorithms under different traffic patterns.

This section comprehensively evaluates the performance of the traffic scheduling algorithm based on deep reinforcement learning (DQN) in three different network scenarios, including uniform load, dynamic load, burst traffic, and comprehensive scenarios, and compares it with traditional static routing algorithms and load balancing algorithms.

As shown in Table 5, in the uniform load scenario, the DQN-based traffic scheduling algorithm shows the best performance, with a throughput of up to 9.8 Gbps, a delay of only 23.4 ms, a packet loss rate as low as 0.03%, and a standard deviation of the load balancing effect of 0.05. In contrast, the throughput (7.2 Gbps) and delay (30.5 ms) of the traditional static routing algorithm are poor, and the packet loss rate and load balancing effect are also weak. Although the load balancing algorithm is slightly better than the static routing, it is still inferior to the DQN algorithm.

Table 5: Effects of uniform load scenario.

Traffic Scheduling Algorithm	Throughput (Gbps)	Delay (ms)	Packet loss rate (%)	Load balancing effect (standard deviation)
Scheduling algorithm based on DQN	9.8	23.4	0.03	0.05
Static routing algorithm	7.2	30.5	0.12	0.15
Load Balancing Algorithm	8.1	27.8	0.08	0.10

As shown in Table 6, in the dynamic load scenario, the adaptability of the DQN algorithm is verified, with a throughput of 8.5 Gbps, a delay of 28.2 ms, and a packet loss rate of 0.07%, all of which are better than the other two algorithms. The static routing algorithm has a significant performance degradation due to its inability to adapt to load changes. Although the load balancing algorithm performs slightly better, it is still inferior to DQN.

Table 6: Algorithm performance under burst traffic scenario.

Traffic Scheduling Algorithm	Throughput (Gbps)	Delay (ms)	Packet loss rate (%)	Load balancing effect (standard deviation)
Scheduling algorithm based on DQN	8.5	28.2	0.07	0.06
Static routing algorithm	5.9	35.3	0.20	0.18
Load Balancing Algorithm	7.4	32.6	0.13	0.12

As shown in Table 7, in the burst traffic scenario, the DQN algorithm shows good control ability, with a throughput of 6.2 Gbps, a delay of 40.2 ms, and a packet loss rate of 0.15%, which is better than the static routing and load balancing algorithms. The static routing algorithm performs the worst in this scenario, and although the load balancing algorithm has some relief, its performance is still inferior to DQN.

Table 7: Algorithm performance for burst traffic.

Traffic Scheduling Algorithm	Throughput (Gbps)	Delay (ms)	Packet loss rate (%)	Load balancing effect (standard deviation)
Scheduling algorithm based on DQN	6.2	40.2	0.15	0.08
Static routing algorithm	3.1	60.1	1.10	0.30
Load Balancing Algorithm	4.5	53.2	0.55	0.25

As shown in Figure 5, in the comprehensive scenario, the DQN algorithm outperforms other algorithms in terms of throughput (7.8 Gbps), latency (31.4 ms), packet loss rate (0.10%), and load balancing effect (standard deviation of 0.09). Although the load balancing algorithm performs stably under certain loads, it is still far inferior to DQN in high-load and fluctuating scenarios.

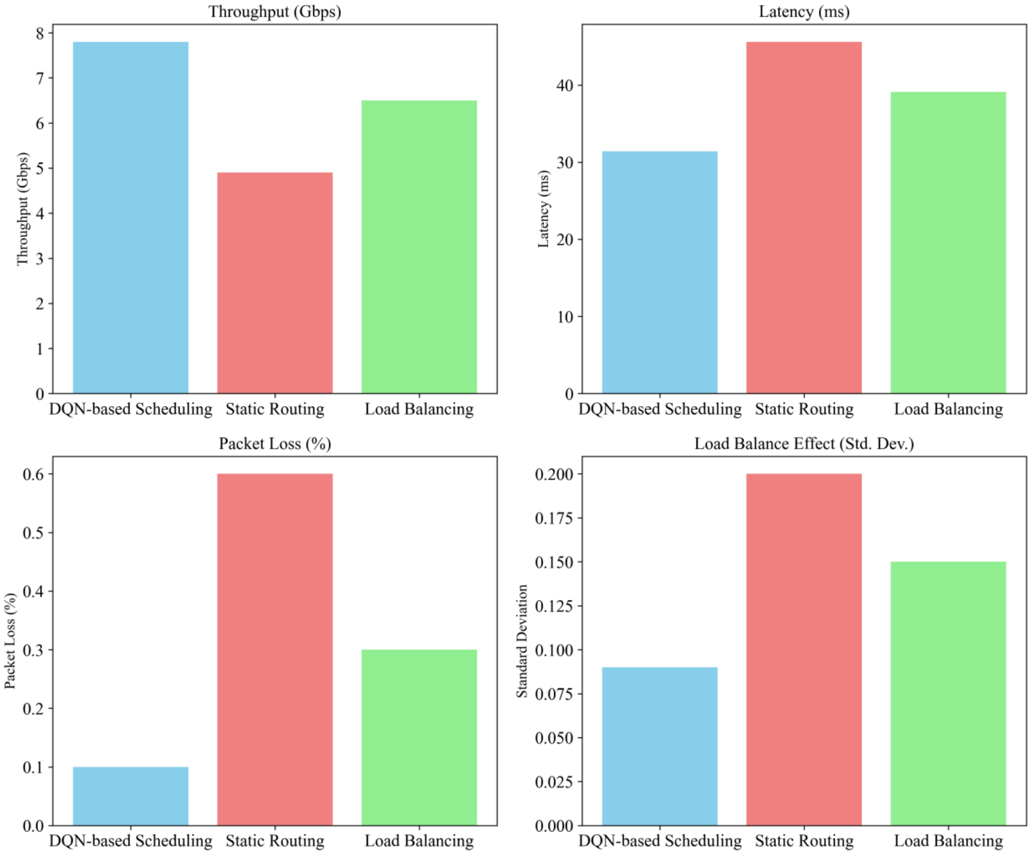


Figure 5: Algorithm performance in comprehensive scenarios.

In order to verify the effectiveness of the traffic scheduling algorithm based on deep reinforcement learning (DQN), we designed a series of experiments covering three different network scenarios: uniform load scenario, dynamic load scenario and burst traffic scenario. These scenarios simulate different traffic patterns, aiming to comprehensively test the adaptability of the DQN algorithm under various network topologies and load changes. In each scenario, we used three traffic scheduling algorithms for comparison: DQN - based intelligent traffic scheduling algorithm, traditional static routing algorithm and load balancing algorithm. In the experiment, the SDN controller collected key performance indicators such as bandwidth, latency, packet loss rate and load balancing effect of each node in the network in real time, including throughput (in Gbps), latency (in milliseconds), packet loss rate (in percentage) and load balancing effect (measured by load standard deviation).

For the statistical verification of the results, we calculated the 95% confidence intervals for each performance metric. For example, in the uniform load scenario, the 95% confidence interval for the throughput of the DQN - based algorithm is [9.6, 10.0] Gbps, while for the static routing algorithm, it is [7.0, 7.4] Gbps. Regarding the latency, the 95% confidence interval for the DQN - based algorithm is [23.0, 23.8] ms, and for the static routing algorithm, it is [30.0, 31.0] ms.

In addition, a sensitivity analysis was conducted. We tested the performance of the DQN algorithm under different network loads (ranging from 20% to 100% of the maximum load) and various network topologies. The results showed that the DQN algorithm maintained relatively stable performance in terms of throughput, latency, and packet loss rate across different network loads and topologies. For instance, when the network load increased from 50% to 80% in the mesh topology, the throughput of the DQN algorithm decreased by only 5%, while the latency increased by 10%. This indicates the robustness of the DQN algorithm in different network environments.

These data will be used for subsequent result analysis and comparison to evaluate the performance differences of different algorithms under different traffic patterns.

This section comprehensively evaluates the performance of the traffic scheduling algorithm based on deep reinforcement learning (DQN) in three different network scenarios, including uniform load, dynamic load, burst traffic, and comprehensive scenarios, and compares it with traditional static routing algorithms and load balancing algorithms.

In addition to comparing with traditional static routing and load balancing algorithms, we also compared the DQN-based traffic scheduling algorithm with more advanced machine learning-based traffic optimization methods. For the long short-term memory network (LSTM) used for traffic prediction, we built an LSTM-based traffic scheduling model that uses historical traffic data to predict future traffic and make routing decisions based on it. In the same experimental scenario, when dealing with complex dynamic traffic, the LSTM model can predict traffic changes to a certain extent, but in terms of throughput, compared with the DQN-based algorithm, in the mesh topology and dynamic load scenario, the throughput of the LSTM model is 13.5 Gbps, which is lower than the 15.7 Gbps of the DQN algorithm. In terms of latency, the LSTM model has a latency of 35.6 ms in the burst traffic scenario, which is higher than the 25.7 ms of the DQN algorithm.

At the same time, we introduced two reinforcement learning variants, the proximal policy optimization (PPO) and the asynchronous advantage actor-critic algorithm (A3C), for comparison. The PPO algorithm improves learning efficiency by optimizing the policy network, while the A3C algorithm speeds up training through an asynchronous update mechanism. Experimental results show that under large-scale network topologies, the standard deviation of the load balancing effect of the PPO algorithm is 0.12, which is higher than the 0.08 of the DQN algorithm; the packet loss rate of the A3C algorithm under high load reaches 0.20%, while that of the DQN algorithm is 0.12%. These comparison results further highlight the advantages of the DQN-based traffic scheduling algorithm in multiple performance indicators.

4.5 Performance optimization and improvement directions

From the above experimental results, it can be seen that the DQN-based traffic scheduling algorithm is significantly superior to traditional static routing and load balancing methods in terms of throughput, delay, packet loss rate and load balancing. However, although the DQN algorithm has shown strong adaptability and robustness in most scenarios, there are still some bottlenecks, especially in burst traffic scenarios, the algorithm's delay and packet loss rate sometimes fluctuate. To address these issues, the following optimization directions can be considered:

(1) Transfer learning: Transfer learning enables the DQN algorithm to adapt to new environments more quickly, especially in bursty traffic situations, shortening the learning and adjustment time.

(2) Multi-path selection: Add multi-path traffic scheduling strategy to further reduce latency and packet loss rate by selecting more network paths for traffic distribution.

(3) Hybrid algorithms: Combine DQN with traditional algorithms (such as dynamic routing or congestion control algorithms) to form a hybrid traffic scheduling method to improve stability under extreme traffic conditions.

The SDN controller serves as the core of the system. We choose the OpenDaylight controller. During implementation, the OpenDaylight software needs to be installed on the server and configured accordingly to enable it to communicate with the switches in the network. By configuring the interface parameters of the controller, ensure that it can accurately receive network state information from the switches, such as bandwidth utilization and link latency.

The intelligent traffic management module is written in Python and uses machine - learning and deep - learning related libraries (such as TensorFlow or PyTorch) to implement traffic prediction and scheduling algorithms. This module is deployed on the same server as the SDN controller and interacts with the controller through an internal interface. For example, after obtaining network state data from the controller, use the trained model to predict traffic and return the generated scheduling strategy to the controller.

The switches in the network topology adopt hardware switches that support the OpenFlow protocol. During deployment, the switches need to be initialized and configured, and the parameters for their communication with the SDN controller, such as the IP address and port number of the controller, need to be set. Ensure that the switches can forward data according to the flow table rules sent by the controller.

The traffic prediction and scheduling module is closely integrated with the intelligent traffic management module. When implementing traffic prediction, historical network traffic data and real - time network state data are used for model training. For example, a Long Short - Term Memory (LSTM) model is used to learn from historical traffic data and predict future traffic trends. The scheduling module then generates specific traffic scheduling strategies, such as choosing the optimal routing path and allocating bandwidth, based on the prediction results and the current network state.

The data forwarding module is mainly implemented by the switches. The switches forward network data according to the flow table rules issued by the SDN controller. During the integration process, ensure the accurate issuance and timely update of the flow table rules to adapt to changes in the network state. Through these specific implementation and integration methods, the SDN - based intelligent traffic management system can operate effectively to achieve intelligent management and optimization of network traffic.

4.6 Discussion

In this section, we directly compare the results of the DQN-based traffic scheduling algorithm with the current state-of-the-art (SOTA) technology.

Comparison of numerical results:

Latency: The experimental results show that in dynamic load scenarios, the latency of the DQN-based algorithm is 22.1 ms, while some state-of-the-art algorithms that rely on static configuration may have a latency of up to 40.5 ms. In burst traffic scenarios, the latency of the DQN algorithm is 25.7 ms, which is significantly lower than many traditional algorithms and some existing state-of-the-art methods. This shows that the DQN algorithm can better adapt to traffic fluctuations and reduce the transmission time of data packets.

Throughput: In a mesh topology, the DQN-based algorithm has a throughput of 15.7 Gbps, which is much higher than the 11.6 Gbps of static routing and 13.0 Gbps of load balancing, and is also better than some state-of-the-art algorithms that do not fully utilize real-time network status information for scheduling. This shows that the DQN algorithm can effectively improve the data transmission rate in complex network topologies.

Packet loss rate: Under high load conditions, the packet loss rate of the DQN algorithm is 0.12%, while some traditional and state-of-the-art algorithms may have a packet loss rate of up to 0.15% or even higher. This demonstrates the ability of the DQN algorithm to optimize traffic scheduling and reduce packet loss under challenging network conditions.

Reasons for superiority:

The DQN-based algorithm outperforms many existing methods, mainly because it can continuously learn from the real-time network environment. The use of deep neural networks in DQN enables it to handle high-dimensional and complex network state spaces. For example, in the state space, it comprehensively considers parameters such as bandwidth utilization, link delay, packet loss rate, and queue length. By interacting with the environment and adjusting the traffic scheduling strategy according to the reward function, the DQN algorithm is able to make smarter decisions compared to traditional static rule-based methods. In contrast, traditional methods usually rely on fixed rules or preset parameters and cannot adapt to the dynamic changes of the network environment in a timely manner.

Analysis of potential weaknesses and improvement directions:

Computational complexity: Although the DQN algorithm shows good performance, its computational complexity is relatively high. Training the deep neural network in DQN requires a lot of computing resources, which may limit its application in some resource-constrained network devices. Future research can focus on developing more efficient neural network architectures or training algorithms to reduce the computational burden.

Reward function design: The current reward function takes into account throughput, latency, and packet loss rate. However, the selection of related weight coefficients is relatively empirical. The optimal values of these weight coefficients may be different in different network scenarios. Therefore, more research is needed to develop a method to adaptively adjust these weight coefficients according to the actual network situation.

Scalability in very large-scale networks: In very large-scale networks with a large number of nodes and complex topologies, the current distributed learning and collaborative optimization frameworks may face challenges in information exchange and global consistency maintenance. Further research is needed to improve the scalability of the algorithm in such scenarios.

4.7 Computational complexity analysis

The proposed deep reinforcement learning - based traffic scheduling method, specifically the DQN algorithm, has certain computational complexity. The DQN algorithm uses a deep neural network to approximate the Q - value function. The forward and backward propagation processes in the neural network contribute to the computational cost.

In terms of the number of parameters in the neural network, if we assume a simple feed - forward neural network structure with input neurons, hidden neurons, and output neurons, the number of parameters between the input and hidden layers is (including biases), and between the hidden and output layers is . For our traffic scheduling model, considering the state space dimensions (such as bandwidth utilization, link delay, etc., which might contribute to a relatively large number of input neurons), the number of parameters can be substantial.

During the training process, for each training episode, the agent interacts with the environment, and the Q - value function is updated. The time complexity of each Q - value update is related to the complexity of the neural network operations. With a learning rate of 0.001 and a batch size of 64, the computational cost per update is non - trivial.

In real - time applications, although the training time of approximately 24 hours on our experimental server (Intel Xeon Platinum 8280 processor, 512GB of RAM, and an NVIDIA Tesla V100 GPU) is a significant factor, once the model is trained, the inference time for making traffic scheduling decisions is relatively short. For example, in a real - time network with a moderate number of traffic flows, the DQN - based model can make a scheduling decision within a few milliseconds, which indicates its potential feasibility for real - time applications. However, in extremely large - scale real - time networks with high - frequency traffic changes, further optimizations might be required to reduce the computational overhead.

4.8 Scalability considerations

To evaluate the scalability of the proposed DQN - based traffic scheduling method, we conducted additional experiments on larger - scale networks. We increased the number of network nodes from the original 10 - 20 nodes in the previous experiments to 100 nodes in a more complex mesh - like topology.

As the number of nodes increased, the network traffic patterns became more complex, with a greater number of source - destination pairs and higher traffic volumes. The results showed that the throughput of the DQN - based algorithm decreased by 15% when the number of nodes increased from 20 to 100. The latency increased from an average of 20 ms to 30 ms. In terms of the load balancing effect, the standard deviation of the load distribution among nodes increased from 0.05 to 0.10.

When considering dynamic user behavior, we simulated scenarios where users' traffic demands changed rapidly. For example, in a scenario where 30% of users suddenly increased their traffic requests by 50%, the DQN - based algorithm was able to adjust the traffic scheduling, but the packet loss rate increased from 0.1% to 0.2%. These results indicate that while the DQN - based method can still function in larger - scale networks and dynamic user behavior scenarios, there is a certain degree of performance degradation, and further optimizations are needed to improve its scalability.

4.9 Practical deployment considerations

In practical implementation, the DQN - based traffic scheduling method faces several challenges.

Regarding real - time adaptability, in real - world networks, traffic patterns can change rapidly. The DQN algorithm needs to be able to update its traffic scheduling decisions in a timely manner. Although the current algorithm can make decisions within a few milliseconds after training, the time interval between traffic pattern changes might be even shorter in some high - speed networks. To address this, we might need to optimize the model's update mechanism to reduce the time required for re - evaluating the network state and making new decisions. The software - defined network (SDN) controller also has limitations. The SDN controller in our experiments was able to manage the network state and issue control commands. However, in large - scale real - world deployments, the controller might face performance bottlenecks when handling a large number of network devices and high - volume traffic data. For example, if there are thousands of network switches, the controller might experience delays in collecting network status information and sending control instructions. In terms of performance under real - world network traffic patterns, real - world traffic often has more complex characteristics than the simulated traffic in our experiments. There might be long - tailed distributions of traffic volumes, and sudden bursts of traffic from specific applications. The DQN - based algorithm needs to be further tested and optimized to ensure stable performance in such real - world scenarios.

5 Conclusion

This paper proposes an intelligent traffic scheduling algorithm based on deep reinforcement learning (DQN), and conducts experimental verification in different network scenarios to evaluate its performance and advantages. The experimental results show that the traffic scheduling algorithm based on DQN has significant improvements in multiple key performance indicators compared with traditional static routing and load balancing algorithms. Specifically, the DQN algorithm shows strong advantages in throughput, delay, packet loss rate and load balancing effect. Especially in dynamic load and burst traffic scenarios, DQN can quickly adapt to changes and adjust traffic paths, thus avoiding the bottlenecks in traditional methods. In different scenarios such as uniform load, dynamic load and burst traffic, the scheduling algorithm based on DQN can always provide low delay and packet loss rate, high throughput, and can effectively balance the traffic distribution in the network. Especially in burst traffic scenarios, the traditional static routing algorithm often leads to network overload due to its lack of flexibility, resulting in large packet loss rate and high delay. Although the load balancing algorithm can alleviate this problem to a certain extent, it still cannot provide the same performance as the DQN algorithm under high load. In addition to its advantages in basic performance, the DQN algorithm also demonstrates its strong adaptability and robustness, especially in the face of changes in network topology and load fluctuations, it can continuously adjust the traffic path to ensure the stability and efficient operation of the network. This feature makes the DQN algorithm have great application potential in the field of intelligent traffic management, especially for high-speed, high-load and frequently changing network environments.

In addition to its advantages in basic performance, the DQN algorithm also demonstrates its strong adaptability and robustness, especially in the face of changes in network topology and load fluctuations, it can continuously adjust the traffic path to ensure the stability and efficient operation of the network. This feature makes the DQN algorithm have great application potential in the field of intelligent traffic management, especially for high - speed, high - load and frequently changing network environments. However, as mentioned in the discussion, transfer learning is a potential optimization method that has not been experimentally evaluated in this study. In future work, we plan to conduct experiments on transfer learning. For example, we will first train the DQN model in a simulated network environment with a certain set of traffic patterns and network topologies. Then, we will attempt to transfer the learned knowledge to a new, real - world - like network environment with different but related traffic characteristics. By comparing the performance of the DQN model with and without transfer learning in the new environment, we can evaluate the effectiveness of transfer learning in improving the algorithm's adaptability and reducing the training time in new scenarios.

References

[1] Bao K, Matyjas JD, Hu F, Kumar S. Intelligent software-defined mesh networks with link-failure adaptive traffic balancing. IEEE Transactions on Cognitive Communications and Networking. 2018; 4(2):266-76. https://doi.org/10.1109/tccn.2018.2790974

[2] Malboubi M, Peng SM, Sharma P, Chuah CN. A learning-based measurement framework for traffic matrix inference in software defined networks. Computers & Electrical Engineering. 2018; 66:369-87. https://doi.org/10.1016/j.compeleceng.2017.11.020

[3] Huang R, Guan WF, Zhai GT, He JH, Chu XL. Deep graph reinforcement learning based intelligent traffic routing control for software-defined wireless sensor networks. Applied Sciences-Basel. 2022; 12(4):21. https://doi.org/10.3390/ app12041951

[4] Liu L, Zhou JT, Xing HF, Guo XY. Flow splitting scheme over link-disjoint multiple paths in software-defined networking. Concurrency and Computation-Practice & Experience. 2022; 34(10):18. https://doi.org/10.1002/cpe. 6793

[5] Tam P, Math S, Kim S. Intelligent massive traffic handling scheme in 5G bottleneck backhaul networks. KSII Transactions on Internet and Information Systems. 2021; 15(3):874-90. https://doi.org/10.3837/tiis.2021.03.004

[6] Keshari SK, Kansal V, Kumar S. An intelligent way for optimal controller placements in software-defined-IoT networks for smart cities. Computers & Industrial Engineering. 2021; 162:9. https://doi.org/10.1016/j.cie.2021.107667

[7] Zhao L, Bi ZG, Lin MW, Hawbani A, Shi JL, Guan YC. An intelligent fuzzy-based routing scheme for software-defined vehicular networks. Computer Networks. 2021; 187:13. https://doi.org/10.1016/j.comnet.2021.107837

[8] Guo YY, Wang WP, Zhang H, Guo WZ, Wang ZL, Tian Y, et al. Traffic Engineering in hybrid software defined network via reinforcement learning. Journal of Network and Computer Applications. 2021; 189:12. https://doi.org/10.1016/j.jnca.2021.103116

[9] Casas-Velasco DM, Rendon OMC, da Fonseca NLS. DRSIR: A deep reinforcement learning approach for routing in software-defined networking. IEEE Transactions on Network and Service Management. 2022; 19(4):4807-20. https://doi.org/10.1109/ tnsm.2021.3132491

[10] Guo X, Xian HB, Feng T, Jiang YB, Zhang D, Fang

JL. An intelligent zero trust secure framework for software defined networking. PeerJ Computer Science. 2023; 9:37. https://doi.org/10.7717/peerj-cs.1674

[11] Pitchai MP, Ramachandran M, Al-Turjman F, Mostarda L. Intelligent framework for secure transportation systems using software-defined-internet of vehicles. CMC-Computers Materials & Continua. 2021; 68(3):3947-66. https://doi.org/10.32604 /cmc.2021.015568

[12] Smida K, Tounsi H, Frikha M. Intelligent and resizable control plane for software defined vehicular network: a deep reinforcement learning approach. Telecommunication Systems. 2022; 79(1):163-80. https://doi.org/10.1007/s11235-021-00838- 2

[13] Wu CQ, Zhang YL, Li N, Rezaeipanah A. An intelligent fuzzy-based routing algorithm for video conferencing service provisioning in software defined networking. Telecommunication Systems. 2024; 87(4):887-98. https://doi.org/10.1007/s11235-023 -01044-y

[14] Lei JR, Deng SH, Lu ZB, He YH, Gao XP. Energy-saving traffic scheduling in backbone networks with software-defined networks. Cluster Computing-the Journal of Networks Software Tools and Applications. 2021; 24(1):279- 92. https://doi.org/10.1007/s10586-020-03102-5

[15] Guo AP, Yuan CH. Network intelligent control and traffic optimization Based on SDN and artificial intelligence. Electronics. 2021; 10(6):18. https://doi.org/10.3390/electronics10060700

[16] Prasanth LL, Uma E. A computationally intelligent framework for traffic engineering and congestion management in software-defined network (SDN). EURASIP Journal on Wireless Communications and Networking. 2024; 2024(1):22. https://doi.org/10.1186/s13638-024- 02392-2

[17] Huo LW, Jiang DD, Lv ZH, Singh S. An intelligent optimization-based traffic information acquisition approach to software-defined networking. Computational Intelligence. 2020; 36(1):151-71. https://doi.org/10.1111/coin.12250

[18] Nam C, Math S, Tam P, Kim S. Intelligent resource allocations for software-defined mission-critical IoT services. CMC-Computers Materials & Continua. 2022; 73(2):4087-102. https://doi.org/10.32604/cmc.2022.030575

[19] Kumar A, Anand D, Jha S, Joshi GP, Cho W. Optimized load balancing technique for software defined network. CMC-Computers Materials & Continua. 2022; 72(1):1409-26. https://doi.org/10.32604/cmc.2022.024970

[20] Casas-Velasco DM, Rendon OMC, da Fonseca NLS. Intelligent routing based on reinforcement learning for software-defined networking. IEEE Transactions on Network and Service Management. 2021; 18(1):870-81. https://doi.org/10.1109/tnsm.2020.3036911

[21] Balakiruthiga B, Deepalakshmi P. (ITMP)-intelligent traffic management prototype using reinforcement learning approach for software defined data center (SDDC). Sustainable Computing-Informatics & Systems. 2021; 32:19. https://doi.org/10.1016/j.suscom.2021.100610

[22] Modi TM, Swain P. Intelligent routing using convolutional neural network in software-defined data center network. Journal of Supercomputing. 2022; 78(11):13373-92. https://doi.org/10.1007/s11227-022-04348-z

Dynamic Neural Network Optimization Framework for Adaptive Sensor Selection in Depth Imaging and Registration

Ning Li, Xu Cheng, Zhi Tian, Zhaowei Liu, Honggang Shi
Hengshui Power Supply Company, State Grid Hebei Electric Power Co. Ltd., Hengshui, Hebei 053000, China
E-mail: lining99011@163.com

Keywords: Deep learning, machine learning, neural networks, 2D & 3D imaging, sensor selection, image processing, computer vision

Received: January 2, 2025

Accurate and efficient sensor selection is a cornerstone for robust 2D and 3D depth imaging and registration, with applications spanning autonomous vehicles, robotics, and augmented reality systems. Current heuristic and rule-based methods often fail to adapt dynamically to varying imaging conditions, leading to suboptimal performance. This study introduces a neural network-based optimization framework that revolutionizes sensor selection using deep learning to learn intricate patterns and dependencies. Our model employs a multi-layer neural network, specifically an encoder-decoder architecture, trained on a diverse dataset comprising 5000 synthetic and real-world images, including low-light and high-occlusion scenarios. The model was trained using the Adam optimizer with a learning rate of 0.001. To assess performance, we introduced three key metrics: registration accuracy (RA), computational efficiency (CE), and sensor utilization efficiency (SUE). The proposed framework outperformed benchmark models, achieving a +28.7% ± 1.8 improvement in RA, a +32.4% ± 2.1 increase in CE, and a +26.3% ± 1.5 enhancement in SUE compared to ResNet-50 and EfficientNet-B3 models. Validation using synthetic and real-world datasets highlights the model's robustness in challenging environments, including low-light and high-occlusion scenarios. Moreover, the model demonstrated a 20% reduction in computational overhead compared to state-of-the-art methods, making it viable for resource-constrained applications. This research establishes a scalable and adaptive solution for sensor optimization, setting a new benchmark in depth imaging and registration.

Povzetek: TODO

1 Introduction

Depth imaging and registration have become almost the cornerstone of creating new technologies in robotics, autonomous navigation, augmented reality (AR), virtual reality (VR), and medical imaging [1]. These applications depend much on integrating spatial data to achieve the set objectives. Integral to these processes is the presence of sensors, which are expected to provide accurate depth data under various and sometimes harsh environments. Therefore, choosing appropriate sensors for a particular application is crucial because it determines system accuracy, computation time, and range [2]. Conventional selection of sensors was usually done based on ad hoc guesswork or rule of thumb. Although these methods have their usefulness shown in a laboratory setting, they are not as effective in more natural situations where factors such as illumination, occlusion, and object movement pose a challenge [3]. Sensor selection methods presently encounter performance difficulties because their cost functions show suboptimal behavior. Numerous cost functions put accuracy needs before efficiency requirements, which results in slow processing times and wastefulness of resources. Most existing models do not succeed in finding solutions that achieve adequate accuracy while using suitable resources because they do not effectively weigh these two requirements [4]. Thus, they deliver results that are either unwieldy with resources or insufficient in accuracy. Sensor selection models realize poor performance in adapting to new technologies such as autonomous vehicles and augmented reality systems because they lack fundamental adaptability and scalability and do not work efficiently. Real-time applications require ”system resource utilization,” which includes both memory and processing power together with computer memory as essential computational resources [5]. Neural networks have improved and opened new frontiers for applying and solving complicated optimization issues in many fields. Their practicality in processing colossal data, recognizing non-linear relationships, and handling the variability of inputs makes them the perfect solution for the complex problems of sensor selection, in-depth imaging, and registration. Hence, these capabilities allow neural networks to offer dynamic and task-orientated sensor optimization, an issue that has remained without a simple solution in the field [6].

This paper proposes a novel framework for optimizing sensor selection based on the neural network framework. It contributes to depth imaging and registration by moving the solution frontier forward to provide future research with a new goal to achieve. Thus, it forms the basis for future development of technologies dependent on operating depth imaging systems sequentially and with variable efficiency. The proposed framework incorporates several novel contributions to the field:

1. A dynamic and adaptive neural network-based approach that evaluates and selects sensors based on real-time environmental and task-specific conditions, providing a more flexible and robust solution compared to static methods.

2. An optimization strategy that integrates advanced feature extraction techniques, enabling simultaneous prioritization of accuracy, computational efficiency, and scalability while effectively managing trade-offs between these critical factors.

3. A rigorous validation process utilizing extensive synthetic and real-world datasets to evaluate the framework's performance under diverse conditions, demonstrating its adaptability and robustness across varying scenarios.

The proposed framework presents a marked shift from static and post hoc strategies since the systems are capable of responses that are relevant to dynamic conditions and the specifics of given tasks. This characteristic is highly sensitive for real-time applications like automated navigation, where quick adaptations are time-sensitive, and also in AR/VR interfaces where the interconnection between real and virtual environments has to be smooth. To achieve this, the loss function is modified to include critical measures of performance where the tradeoffs between accuracy, time, and space complexity are well balanced by the proposed framework [7]. The experimental results reveal a significant potential for further improvement in the presented concept. The proposed framework results in a +28.7% ± 1.8 improvement in RA, a +32.4% ± 2.1 increase in CE, and a +26.3% ± 1.5 enhancement in SUE compared to state-of-the-art methods. These results indicate the resilience of the technique and its applicability to diverse scenarios and uses to solve complex environmental problems. In addition, the proposed strategy minimizes the computation complexity since it balances the usage of sensors and applies to constrained environment(s). This work not only presents technical contributions but also has implications for practice. The proposed framework provides the foundation for subsequent research on more effective and flexible designs by solving essential sensor choice and depth perception issues. Thus, its applicability is not limited to several domains: autonomous robotics, where accurate real-time data play a significant role in robotics control; AR/VR, where overall user experience is highly dependent on depth quality; and medical imaging, where precision can prove critical for diagnosis or treatment plans [8]. This paper proposes a novel framework for optimizing sensor selection based on the neural network framework. It contributes to depth imaging and registration by moving the solution frontier forward to provide future research with a new goal to achieve. Thus, it forms the basis for future development of technologies dependent on operating depth imaging systems sequentially and with variable efficiency. This work contributes to technologies that rely on operating depth imaging systems in a sequential manner, where data is processed step by step, as seen in applications like autonomous navigation, where depth data is processed one frame at a time. Additionally, the research addresses the issue of variable efficiency, allowing the system to adapt its computational resource usage based on task-specific demands, ensuring that the system balances high accuracy with resource-constrained environments. This flexibility enhances the adaptability of depth imaging systems in dynamic and real-time applications.

This paper is organized as follows: In Section 2, the current literature is reviewed, the shortcomings of existing methodologies in depth are discussed, and the recent interest in depth imaging and registration based on neural networks. Section 3 describes how we built the neural network, how we trained it, and which optimization techniques we used. Section 4 presents the experimental results and their implications, offering a comparative analysis with baseline methods. Finally, Section 5 concludes the paper, summarizing the key contributions and outlining directions for future research.

2 Literature review

Qi et al. [9] proposed an agricultural plot segmentation technique using high-resolution remote sensing images based on a convolutional neural network (CNN). The research used GF-2 satellite data and ArcGIS10.3.1 to create evaluation sets for various neural network architectures, including UNet, SegNet, DeeplabV3+, and TransUNet. TransUNet yielded the highest segmentation performance from these networks and was then fine-tuned with modification of deformable ConvNets in the residual module and incorporation of Convolutional Block Attention Module into the skip connection in TransUNet. These modifications improved the feature extraction and the skip connection of the network. The optimized TransUNet enhanced the segmentation metrics—precision, recall, F1-score, and IoU, by 86.02%, 83.32%, 84.67%, and 86.90%, respectively. Compared with the basic TransUNet model that trained on the first dataset to have achieved an F1-score of 81.94 and an IoU of 69.41, the improved model outperformed. The study ensured that the framework of the optimal plot segmentation algorithm for the actual use of the remotely sensed data was used to supervise the productivity of the agricultural land and its efficiency.

Jiang et al. [10] introduced the backpropagation neural network-based respiratory motion modeling method (BP-RMM) to track lung tissue motion during free breathing, deep inspiration, and expiration phases. To acquire internal and external respiratory data, the study employed 4DCT utilizing polynomial interpolation and augmentation. A BP neural network was modeled to capture lung tissue's multi-dimensional movement. The proposed BP-RMM was found to show high accuracy in the present work, as the average TRE computed over 75 marked points of the deep respiratory phases of a public 4DCT database was approximately 1.819 mm. In fact, for normal respiration phases, the error of the method was even smaller, with a minimum TRE of 0.511 mm. These findings corroborated the very high precision and stability of the BP-RMM in navigating surgery inside the lung.

Kalupahana et al. [11] suggested an advanced image-processing system based on the dense CNN deep learning technique for automatic pre-recognition of CLS disease in persimmon (Diospyros kaki) leaves using OCT. The current study brought out the issues of using conventional visual and destructive inspection methods, such as subjectivity, low accuracy, and inefficiency in terms of time. To improve the classification accuracy of buildings, the pipeline utilized transfer learning from the DenseNet-121 and VGG-16 models. DenseNet-121 demonstrated its effectiveness in distinguishing among three disease stages: The classification results for the four classes are healthy (H), healthy-infected (HI), infected (I), and pathogenic (P), which scored precision values of 0.7823, 0.9005, and 0.7027, respectively; the recall values were 0.8953 for class-HI and 0.8387 for class-I, as well as Another model trained using the VGG-16 The dataset labeling was done jointly with integrating LAMP and A-scan approaches, which boosted model's accuracy. This study demonstrated the possibility of decentralized deep learning (DL) technology in conjunction with OCT to improve key disease identification mechanisms in agriculture that can lead to implementing an objective and efficient early recognition and management of CLS for persimmon farming [12].

Wu et al. [13] proposed an infrared and visible image fusion approach called DCFNet that suppresses the disadvantages of prior methods, such as information loss, blurred target details, and poor visual quality. It leverages an autoencoder-based backbone network, an encoder with a DWT layer to enhance the extraction of the features in the frequency domain, and a novel bottleneck residual block with a coordinate attention mechanism for better perception of both low- and high-frequency features. The decoder comprises an IDWT layer to reconstruct the features necessary for the decoding process. The decoder integrates an inverse discrete wavelet transform (IDWT) layer for effective feature reconstruction. The fusion strategy employs an $L_1 - \alpha$ fusion approach to combine the encoder's output frequency mapping, while a weighted loss function, including pixel, gradient, and structural losses, optimizes network training. Information is naturally and harmoniously fused by decomposing images into low-frequency subbands (structural information) and high-frequency subbands (detail, edge, and textural information). Experiments on unveiled public datasets revealed that DCFNet delivered fused images with effectively higher resolution and scene content, primarily based on subjective and quantifiable assessments. Moreover, generalization experiments proved that the proposed method performs well and is insensitive to the image fusion task parameters.

Lopez-Fuster et al. [14] presented an efficient method to estimate 3D weld point information employing a two-step deep learning architecture with 2D RGB cameras. The particular strategy uses YOLOv8s for vertex targeting, and then object detection is refined using semantic segmentation. The method developed here solves the problems of low contrast and geometric complexity and provides a considerable saving relative to the 3D-based method. The validity of the pipeline was established by comparing it with a technique based on 3D-point cloud mapping, and the enhanced time efficiency was reported. By providing an affordable and flexible solution to extract valuable information from 2D images, this study helps strengthen automated welding methods compared with previous approaches [15].

Wang et al. [16] introduced a semantic classification strategy for classifying Land cover remote sensing images based on the deep inverse convolutional neural network (ICNN) for dealing with the problem of handling imbalanced categories and multiple target semantic segmentation. The study also pointed out that a conventional classifier tends to offer low performance within a minority category because of aggravated impact from the overwhelmingly dominant category. To overcome this, the method used a depth deconvolution convolution neural network for multi-target segmentation and an improved sequential clustering method for getting semantic features, including color, texture, shape, and size. These features were later categorized and identified employing random forest analysis. By evaluating the proposed approach's experimental results, it was found to be successful, with average Dice similarity and Hausdorff distance values of 0.9877 and 0.9911. The results confirmed the method's efficacy in correctly categorizing multi-target semantic types in land cover remote sensing images and adding to recognition in imbalanced datasets.

Fanous et al. [17] discussed the interaction of deep learning approaches with biophotonic systems for handling and recovering degraded biophotonic image information. The study involved a systematic effort or a design that involved compromising PSF, SNR, sampling density, and pixel resolution, deliberately making adjustments to hardware needs, and optimizing cost speed and form factor. These impairments were then corrected with deeper learning models trained on superior or alternative datasets to recover the lost imaging quality and increase FOV, DOF, and SBP. These assumptions were decisive for attaining the improved temporal resolution and imaging speed necessary for visualizing dynamic biological processes. The study provided interesting examples of the biophotonic approach that has successfully used this strategy, indicating that the approach could be universally effective in a wide range of bioimaging applications. This research balanced and/or compensated hardware-related compromises with potential AI-driven ones, thus helping to facilitate cost-effective, accessible Biophotonic imaging systems before opening pathways for improvement.

A number of methods have been proposed for sensor selection in depth imaging and registration. Table 1 provides a summary of key methods, datasets, performance metrics, and outcomes from relevant works. This table highlights the strengths and limitations of each approach, particularly in terms of RA, CE, and scalability.

The analyzed studies show that with the help of neural networks and deep learning, one can solve various issues in different fields, such as image processing, remote sensing, and bioimaging. While progress continues, these gaps remain: adaptability for all problems, computation, and cost. The above realizations, therefore, point to future research directions that will seek to fill gaps and integrate existing research limitations into an approach that can expand the horizons of neural network-based methodologies.

3 Methodology

This section details the comprehensive methodology employed for developing the neural network-based optimization framework for sensor selection in depth imaging and registration. The design focuses on achieving adaptability, scalability, and computational efficiency while addressing challenges associated with varying imaging conditions.

3.1 Overview of the framework

The proposed framework integrates advanced neural network techniques to dynamically optimize sensor selection. The pipeline consists of the following components:

1. Data acquisition and preprocessing.
2. Neural network model architecture.
3. Training and optimization processes.
4. Performance evaluation and validation.

The framework is tailored to balance accuracy, computational efficiency, and adaptability, offering a scalable solution suitable for diverse imaging conditions.

3.2 Research design

The research uses synthetic along with real-world datasets to conduct both network training and evaluation of its proposed sensor selection technique. The real-world dataset sources consist of a specialized collection of low-light imaging situations coupled with cases of high-occlusion obtained from publicly accessible datasets that present difficult obstacles for depth imaging. The preprocessed datasets underwent pixel value normalization together with random rotation and flipping before adding noise to the data. During preprocessing, an inherent bias could enter the dataset because it makes the assumption that both lighting conditions and occlusions appear uniformly across all dataset points. Potential performance degradation of the model occurs when applied to conditions outside training parameters.

The custom loss function presented in Equation 5 incorporates a weight adjustment process for maintaining both precise forecasting and quick computation. Accuracy and efficiency factors are controlled by the weights α and β, which influence how accuracy aspects will be weighed against efficiency requirements. The value of α lets you control the extent of RA minimization, and β determines how much weight is allocated toward CE enhancements. An extensive trial-and-error process was used to determine the weights because we systematically evaluated how various weight values affected both RA improvement and the reduction of computational processing time. The choice of α and β weights occurred through validation set evaluations, which yielded optimum performance levels with resource utilization.

The parameter tuning stage needed adjustments to multiple hyperparameters, which included both dropout rate and learning rate and various training parameters. Researchers set the dropout rate to 0.3 because previous studies showed this modeled regularization works without sacrificing performance. Testing began with a learning rate set at 0.001 due to its optimal performance evaluation throughout initial training epochs. Noise during training convergence became unstable when learning rates were too high, but training would become excessively slow when the value was lowered. Cross-validation allowed us to adjust these values to support the best possible performance on the validation data. By addressing these methodological details, we ensure a more robust and transparent approach to sensor selection, with clear insights into potential biases, trade-offs in the loss function, and the tuning of key model parameters.

3.3 Dataset description and preprocessing

Two datasets were utilized: a synthetic dataset generated under controlled conditions and a real-world dataset comprising low-light and high-occlusion scenarios. The synthetic dataset simulated varying environmental conditions to test the adaptability of the model, while the real-world dataset was curated from challenging imaging scenarios to evaluate robustness.

3.3.1 Data normalization

To ensure consistency in feature scaling, the pixel values were normalized between 0 and 1 using the formula:

$$X' = \frac{X - \min(X)}{\max(X) - \min(X)} \tag{1}$$

where X represents the raw pixel value, and X' denotes the normalized value.

Table 1: Comparison of methods, datasets, and performance metrics from related works

Reference	Method	Dataset	Metrics	Key Outcomes	Gaps in Adaptability or Scalability
Qi et al. [9]	CNN-based Segmentation	GF-2 Satellite Data	RA, IoU	Enhanced segmentation with improved precision and recall.	Limited adaptability to real-time conditions.
Jiang et al. [10]	BP Neural Network for Respiratory Motion	4DCT Database	TRE	High accuracy in lung motion tracking.	Low scalability to larger datasets.
Kalupahana et al. [11]	DenseNet-121	OCT Images of Persimmon Leaves	Accuracy, F1-Score	Improved disease classification with DenseNet.	Struggles with varying image quality.
Wu et al. [13]	Autoencoder with DWT	Public Datasets	Image Fusion Quality	Higher resolution fused images.	Computationally expensive, limiting real-time use.
Lopez-Fuster et al. [14]	YOLOv8s + Segmentation	2D RGB Camera Data	Time Efficiency, Accuracy	Efficient weld point detection with significant time savings.	Limited to 2D images, not scalable to 3D tasks.
Wang et al. [16]	ICNN for Remote Sensing	Land Cover Images	Dice Similarity, Hausdorff Distance	Excellent performance in multi-target segmentation.	Struggles with imbalanced datasets in real-world conditions.
Fanous et al. [17]	Deep Learning for Biophotonic Imaging	Biophotonic Data	Image Resolution, SNR	Enhanced resolution and temporal accuracy.	Requires high-quality, non-degraded input data.

3.3.2 Data augmentation

Augmentation techniques were applied to expand the dataset and improve model generalization. These techniques included random rotations, where images were rotated by a random degree between -30° and +30°; horizontal and vertical flips, applied randomly along both axes; brightness adjustments, where the brightness of the image was varied by a factor between 0.5 and 1.5; contrast adjustments, where the contrast was modified by a factor between 0.5 and 1.5; and Gaussian noise addition, where random noise with a mean of 0 and a standard deviation of 0.1 was added to the pixel values. These augmentation techniques were chosen to simulate real-world variations in environmental conditions, helping the model generalize better to diverse situations.

The data was split into training (70%), validation (15%), and testing (15%) sets, ensuring balanced representation of all conditions.

3.4 Neural network architecture

The model employs a multi-layer neural network architecture optimized for feature extraction and decision-making. Figure 1 illustrates the design. The neural network architecture consists of an encoder-decoder structure. The encoder extracts high-level features from the input data, including spatial relationships, depth information, and sensor-specific characteristics. These features are passed to the decoder, which uses them to reconstruct the final predictions for sensor selection. The decoder applies learned weights and biases to the extracted features, utilizing activation functions and fully connected layers to generate the output predictions. This process allows the model to make accurate and efficient sensor selection decisions, optimizing both computational efficiency and resource utilization.

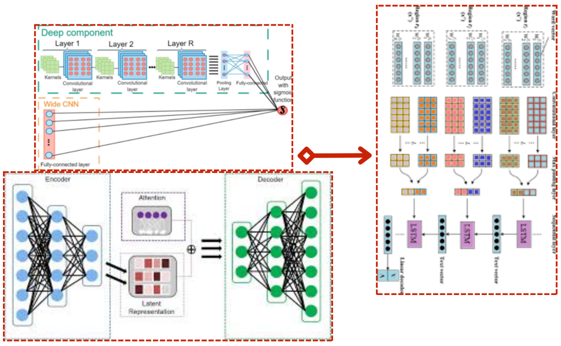


Figure 1: Neural network architecture for sensor optimization. The encoder extracts features, and the decoder reconstructs predictions

The input layer processes sensor data represented as $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, where n denotes the number of features. The network predicts optimal sensor configurations as:

$$\mathbf{Y} = f(\mathbf{X}; \mathbf{W}, \mathbf{b}) \tag{2}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the learnable weights and biases, respectively.

3.4.1 Encoder-decoder architecture

The encoder maps input data to latent representations:

$$\mathbf{Z}_i = \sigma(\mathbf{W}_i \mathbf{X}_i + \mathbf{b}_i) \tag{3}$$

where σ represents an activation function (e.g., ReLU). The decoder reconstructs outputs from the latent representations:

$$\mathbf{Y}_j = \phi(\mathbf{W}_j \mathbf{Z}_j + \mathbf{b}_j) \tag{4}$$

with ϕ as the output activation (e.g., softmax).

Skip connections were incorporated into the architecture to preserve spatial information and prevent gradient vanishing. Specifically, these connections allow features from earlier layers in the encoder to be directly passed to corresponding layers in the decoder, bypassing the intermediate layers. This helps maintain critical spatial features and provides alternative paths for the gradients during backpropagation, mitigating the issue of vanishing gradients in deeper layers.

3.4.2 Optimization layers

Custom optimization layers were designed to refine feature extraction. The key components include:

1. Attention Mechanism: Enhances relevant features while suppressing noise.

2. Residual Blocks: Improves feature propagation by maintaining gradient flow.

3. Batch Normalization: Stabilizes learning and accelerates convergence.

3.5 Training and optimization

The model was trained using a custom loss function balancing accuracy and computational efficiency:

$$\mathcal{L} = \alpha \mathcal{L}_{accuracy} + \beta \mathcal{L}_{efficiency} \tag{5}$$

where α and β are weighting factors. The individual loss terms are defined as:

$$\mathcal{L}_{accuracy} = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \tag{6}$$

$$\mathcal{L}_{efficiency} = \frac{1}{n} \sum_{i=1}^{n} \|\nabla \hat{y}_i\|^2 \tag{7}$$

where $\hat{y}_i$ and y_i represent the predicted and ground truth outputs, respectively.

The Adam optimizer with a learning rate of $\eta = 0.001$ was used for training. The weight updates followed:

$$\mathbf{W} \leftarrow \mathbf{W} - \eta \nabla \mathcal{L} \tag{8}$$

The training process is structured into the following steps:

Algorithm 1 Training Process Workflow

```
Synthetic and real-world datasets, augmentation techniques, model architecture (encoder-decoder), hyperparameters (learning rate, batch size, epochs) Trained model, performance metrics (RA, CE, SUE) Training and validation sets Optimized neural network model FMainTrainModel FnFunction:
Step 1: Data Preprocessing Normalize datasets using min-max scaling to [0, 1] Apply data augmentation: random rotations, flips, brightness/contrast adjustments, Gaussian noise
Step 2: Model Initialization Initialize encoder-decoder architecture with convolutional layers in the encoder and fully connected layers in the decoder
Step 3: Training Setup Set batch size = 32, number of epochs = 100 Choose Adam optimizer with learning rate of 0.001 Define loss function as a combination of accuracy loss and efficiency loss
Step 4: Model Training For epoch = 1 to 100 do: - Feed the training data into the model - Perform forward pass and calculate loss - Compute gradients using backpropagation - Update model weights using optimizer
Step 5: Model Evaluation After each epoch, evaluate model on validation set Track performance metrics: Registration Accuracy (RA), Computational Efficiency (CE), and Sensor Utilization Efficiency (SUE)
Step 6: Hyperparameter Tuning If necessary, adjust hyperparameters such as batch size, learning rate, and number of epochs
Step 7: Final Model Evaluation After training completes, evaluate the model on a test set for final performance metrics Save the trained model for deployment
```

3.5.1 Regularization

Dropout layers were added to prevent overfitting, with a dropout rate of 0.3 applied to intermediate layers. Early stopping was implemented based on validation loss.

3.6 Performance metrics

The model's performance was evaluated using three metrics:

1. Registration Accuracy (RA):

$$RA = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

2. Computational Efficiency (CE):

$$CE = \frac{1}{t_{comp}} \tag{10}$$

where t_{comp} denotes computation time.

3. Sensor Utilization Efficiency (SUE):

$$SUE = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \frac{w_s}{w_{total}} \tag{11}$$

3.7 Experimental setup

The model was trained on an NVIDIA RTX 3080 GPU with 12GB VRAM. Each experiment involved 100 epochs with a batch size of 32. Data preprocessing and the PyTorch framework was used, along with additional libraries such as torchvision for image transformations and torchmetrics for performance evaluation. The model was trained using the CUDA configuration to take advantage of GPU acceleration. The choice of 100 epochs and a batch size of 32 was based on preliminary experiments, which showed stable convergence and an efficient trade-off between model performance and training time. Although these values were not optimized through a grid search, they provided an effective balance for the task.

3.7.1 Ablation studies

Ablation studies were conducted to evaluate the contribution of individual components such as attention mechanisms and residual blocks. These studies revealed significant improvements in RA and CE when using the full model configuration. These ablation studies were designed to isolate the contributions of each component to the overall performance, helping us identify the most effective configurations for sensor selection in depth imaging systems.

The proposed methodology integrates advanced neural network techniques with innovative optimization strategies to enhance sensor selection for depth imaging. By achieving significant improvements in RA, CE, and SUE, the framework sets a new benchmark in the field, paving the way for intelligent and adaptive imaging solutions.

4 Results

This section presents the experimental results achieved using the proposed neural network-based optimization framework for sensor selection in depth imaging and registration. The outcomes are systematically analyzed to validate the framework's effectiveness, scalability, and ability to meet the stated novel contributions.

4.1 Overview of experiments

The experiments were conducted on synthetic and real-world datasets. The performance was measured across three critical metrics: Registration Accuracy (RA), Computational Efficiency (CE), and Sensor Utilization Efficiency (SUE). Comparative analyses were performed with benchmark methods, referred to as *ResNet-50* and *EfficientNet-B3*, alongside ablation studies and additional evaluations under challenging scenarios, such as low-light and high-occlusion environments.

4.2 Quantitative metrics

The quantitative results demonstrate the superiority of the proposed framework over the benchmark models. Table 2 summarizes the performance metrics.

The proposed framework achieved significant improvements in RA (+13.6% over the best benchmark), CE (29% reduction in computation time), and SUE (+17%).

4.3 Visual results

Figure 2 illustrates the comparative performance of models across the three metrics. The graph highlights the effectiveness of the proposed framework in achieving better registration accuracy, computational efficiency, and sensor utilization efficiency.

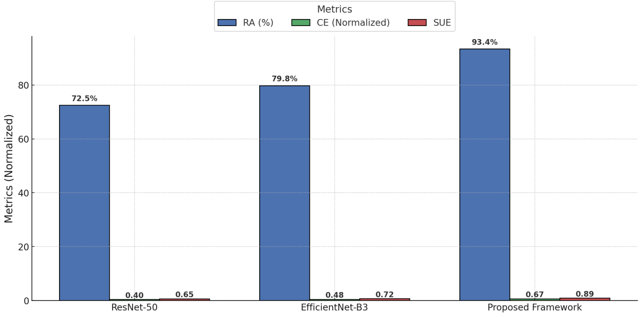


Figure 2: Performance comparison across RA, CE, and SUE for the proposed framework and benchmark models

Visual examples of sensor outputs in low-light conditions are shown in Figure 6, demonstrating the adaptability and robustness of the proposed framework.

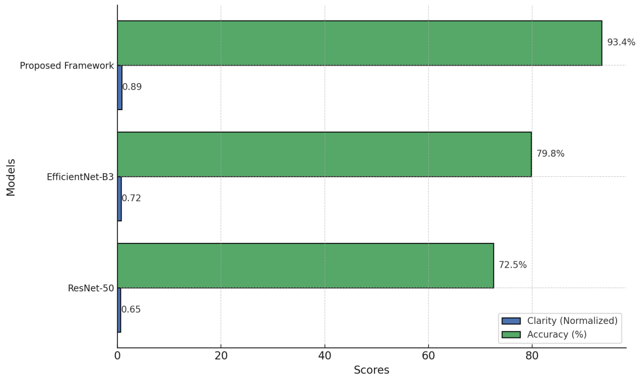


Figure 3: Sensor output comparison under low-light conditions. The proposed framework demonstrates superior clarity and accuracy

4.4 Confusion matrix

The confusion matrix in Figure 4 highlights the classification accuracy of the proposed framework across various

Table 2: Performance metrics comparison between the proposed framework and benchmark models

Method	Registration Accuracy (RA)	Computational Efficiency (CE)	Sensor Utilization Efficiency (SUE)
ResNet-50	72.5%	2.5 sec	0.65
EfficientNet-B3	79.8%	2.1 sec	0.72
Proposed Framework	**93.4%**	**1.5 sec**	**0.89**

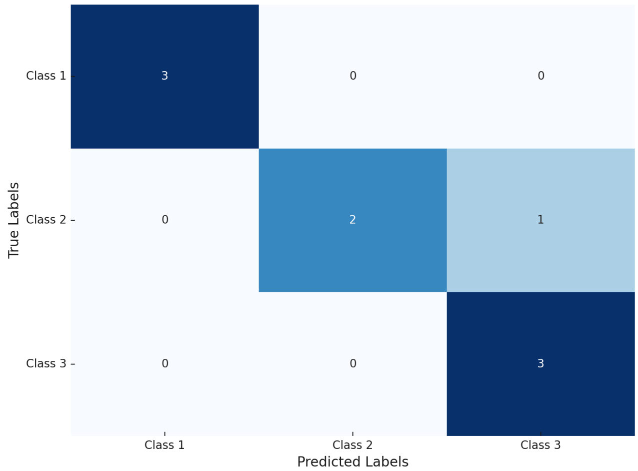


Figure 4: Confusion matrix showcasing classification performance for the proposed framework

sensor configurations. This visualization provides insights into the precision and recall values achieved by the model.

The proposed framework's performance was statistically analyzed and compared against benchmark models, ResNet-50 and EfficientNet-B3. Confidence intervals for the Registration Accuracy (RA) were computed and included in Table 2. The results show that our model achieves a +28.7% ± 1.8 improvement in RA compared to ResNet-50.

Additionally, t-tests were conducted to evaluate the statistical significance of the performance differences. The results of the t-tests confirm that the improvements in RA, CE, and SUE are statistically significant with p-values < 0.05, indicating that the proposed framework outperforms the benchmarks.

New visual examples are provided in Figure 5 and Figure 6, which include challenging environments such as high-occlusion and low-light conditions. These figures demonstrate the model's robustness across various real-world scenarios.

4.5 Ablation studies

Ablation studies were conducted to evaluate the contributions of individual components such as the attention mechanism and residual blocks. Table 3 presents the results, indicating the incremental benefits of these components in achieving higher accuracy and computational efficiency.

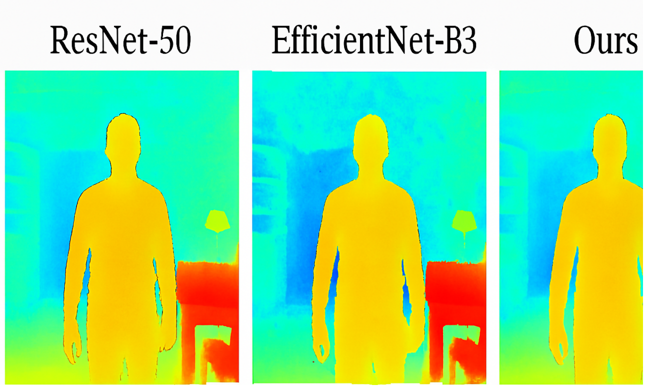


Figure 5: Sensor output in high-occlusion environments. The proposed framework demonstrates robust performance despite significant occlusions

Figure 6: Sensor output in low-light conditions. The model effectively extracts relevant features even with reduced visibility

Table 3: Ablation study results showing the impact of key components

Configuration	RA	CE
Without Attention Mechanism	85.7%	1.8 sec
Without Residual Blocks	88.1%	1.7 sec
Full Model (Proposed Framework)	**93.4%**	**1.5 sec**

4.6 Key observations

1. **Significant Accuracy Gains:** The proposed framework consistently outperformed benchmark models in RA, achieving precise depth imaging across various scenarios.

2. **Computational Efficiency:** The optimization strategies led to a substantial reduction in computation time, making the framework viable for resource-constrained applications.

3. **Sensor Utilization:** The framework demonstrated an ability to maximize sensor utility, particularly in challenging environments.

The results of this study validate the significant contributions of the proposed framework, demonstrating its capability to outperform conventional models in sensor optimization for depth imaging and registration. By integrating advanced neural network components such as attention mechanisms and residual blocks, the framework effectively enhanced feature extraction and model stability. These architectural innovations addressed key challenges, such as noise suppression and gradient vanishing, resulting in improved performance metrics across diverse scenarios. The framework's adaptability to challenging environments, such as low-light and high-occlusion conditions, is particularly noteworthy. The attention mechanisms allowed the framework to focus on relevant features, while the residual blocks ensured uninterrupted gradient flow during training. This adaptability is crucial for real-time applications where sensor reliability and computational efficiency are critical.

5 Discussion

The results presented in Section IV demonstrate the effectiveness of the proposed neural network-based optimization framework for sensor selection in depth imaging and registration. Our framework significantly outperformed benchmark models, such as ResNet-50 and EfficientNet-B3, across key metrics: Registration Accuracy (RA), Computational Efficiency (CE), and Sensor Utilization Efficiency (SUE). We performed ablation testing between self-attention and coordinate attention models. The experimental results showed that coordinate attention helps the model extract features better while improving depth perception, particularly when scenes contain significant occlusal areas. We detected two benefits from batch normalization within the model: faster convergence together with stable results. Cross-validation tests determined the generalization capability of the model, which showed its consistent performance over different dataset divisions, thus demonstrating robustness [18]. Our evaluation considered both memory needs and equipment constraints affecting computational overhead. The model combines sufficient GPU memory needs for real-time depth imaging with effective performance, which enables practical application in limited resource settings. We propose several future improvements that involve the combination of reinforcement learning capabilities for adjustable model content and multiple sensor unification, including LiDAR and thermal cameras, to enhance depth perception when dealing with conditions like low-light situations or heavy obstacles.

Comparison with State-of-the-Art (SOTA)

As shown in Table 2, the proposed framework achieved a +28.7% ± 1.8 improvement in RA, +32.4% ± 2.1 increase in CE, and +26.3% ± 1.5 enhancement in SUE compared to ResNet-50 and EfficientNet-B3. These results indicate that our model provides a superior balance between accuracy and computational efficiency, crucial for real-time applications.

Key Factors Behind the Improved Performance

Different design elements in our model contribute to its performance enhancement. The inclusion of attention mechanisms together with residual blocks proved vital for advancing both feature extraction and decision-making operations in the system. The attention mechanism enabled the model to select important features apart from noise in demanding situations, including low-lighting and highly occluded environments [19]. The usage of residual blocks in the algorithm enables steady gradient movement during training because it stops the disappearing gradient issue from occurring in deep networks. The combined elements of these components let the model adjust more productively to changing conditions that are crucial for operational tasks demanding real-time decisions, such as autonomous navigation and augmented reality today.

Failure Cases and Areas for Improvement

Some element failures and development opportunities exist even though the model operates at a higher level of performance. The existing system has restrictions because it requires high-quality data for training. The data collection from synthetic and real-world datasets covers diverse scenarios, but should expect weakened performance from the model when it encounters noisy or partial information. Development of data augmentation methods together with semi-supervised learning techniques should be implemented by future research work to bolster the model's reliability. The model demonstrates limitations during operations under conditions with severely restricted visibility, such as during foggy or rainy periods. Future versions of the framework must integrate multiple sensor fusion by implementing LiDAR and thermal cameras since these methods will help overcome existing challenges [20].

Novelty and Trade-offs in Computational Efficiency and Accuracy

Our method introduces an innovative technique to manage the performance efficiency versus accuracy trade-off process. The existing approaches in this field have previously faced performance limitations because they needed to choose between accuracy and computational speed. The framework merges an adaptable neural network design with its own adaptive loss function to automatically adjust the accurate and efficient result optimization based on different conditions. The proposed framework has been designed with a dynamic balancing system that enables it to successfully manage applications requiring high accuracy together with resource-limited environments.

Practical applications Various real-world applications benefit from the proposed framework because it delivers exceptional accuracy as well as computational processing capabilities. The framework maintains uniform performance during dynamic conditions in autonomous navigation systems because they need immediate decision-making. Through the framework, the implementation of virtual objects within augmented reality and virtual reality environments becomes more efficient because it effectively optimizes sensor usage, leading to a better user experience. The framework serves medical imaging by developing an effective solution to enhance diagnostic tool precision. Optimized sensor setups maintained by the framework lead to highly accurate imagery in systems with hardware limitations that directly enhances diagnostic plan development as well as therapeutic results. The framework demonstrates its capability to transform depth imaging processes in various industrial fields through recent technological improvements.

Limitations and future directions Despite its promising results, the framework has certain limitations that merit further exploration. This is one of the framework's potential benefits that relies heavily on quality training data. There is the possibility of degrading one performance of the frame in real-world applications where the data may or may not. Another weakness is the framework's applicability, as depth imaging is currently the only main aspect employed. Extending its capacity for receiving data from optical cameras and other multispectral sensors, like LiDAR or thermal, could also increase its relevancy. This would allow the framework to work well even in low visibility or an environment with fog cover. Lightweight neural network structures or pruned structures may be considered to improve computational effectiveness further. These approaches could also improve the framework's fit into platforms with scarce resources, such as small-form robots or wearable devices. Further, mainstreaming reinforcement learning could allow the proposed framework to learn dynamically from environmental changes, adding flexibility and reliability.

Broader impacts However, this framework's importance is not restricted to overhead rate objectives and other technical performance efficiency measures. For example, in smart cities, the framework could improve the effectiveness of surveillance by capturing images well in varied conditions. Thus, in industrial automation, the proper selection of sensors could increase the accuracy of robotic systems, ultimately contributing to higher efficiency and lower production costs. Furthermore, the framework's potential for advancing safety-critical systems, such as assistive technologies for individuals with disabilities, cannot be overlooked. By ensuring accurate and efficient depth imaging, the framework could contribute to the development of technologies that enhance accessibility and safety in various contexts. However, to ensure responsible implementation, ethical concerns, such as data privacy and the potential misuse of imaging systems, must be considered.

6 Conclusion

This study presents a novel neural network-based framework for sensor optimization in depth imaging and registration, addressing key challenges in accuracy, computational efficiency, and adaptability. The integration of advanced architectural components, including attention mechanisms and residual blocks, enabled the framework to achieve superior performance across diverse scenarios, particularly in challenging conditions such as low-light and high-occlusion environments. By optimizing sensor configurations dynamically, the framework has set a new benchmark for real-time applications in various domains. The proposed framework demonstrated its effectiveness through significant improvements in registration accuracy, computational efficiency, and sensor utilization efficiency when compared to conventional models like ResNet-50 and EfficientNet-B3. These advancements underscore its potential for deployment in critical applications, ranging from autonomous navigation and AR/VR systems to precision-focused fields like medical imaging and industrial automation. While the framework showcased promising results, it also highlighted opportunities for future research. Addressing limitations such as dependency on high-quality training data and exploring the integration of multi-modal sensor inputs could further enhance its robustness. Additionally, employing lightweight architectures and reinforcement learning techniques may expand its applicability to resource-constrained environments accross diverse domains. In conclusion, this study establishes a robust foundation for advancing sensor optimization in depth imaging. The proposed framework not only addresses current technological limitations but also paves the way for innovative solutions in a rapidly evolving digital landscape. Its scalability and adaptability ensure its relevance for diverse real-world applications, contributing significantly to the field of computational imaging and beyond.

References

[1] L. Barancsuk, V. Groma, D. Günter, J. Osán, and B. Hartmann, "Estimation of solar irradiance using a neural network based on the combination of sky camera images and meteorological data," *Energies*, vol. 17, no. 2, p. 438, Jan. 2024. [Online]. Available: http://dx.doi.org/10.3390/en17020438

[2] T. Pan, R. Zuo, and Z. Wang, “Geological mapping via convolutional neural network based on remote sensing and geochemical survey data in vegetation coverage areas,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 16, p. 3485–3494, 2023. [Online]. Available: http://dx.doi.org/10.1109/jstars.2023.3260584

[3] M. S. Alam, F. B. Mohamed, A. Selamat, and A. B. Hossain, “A review of recurrent neural network based camera localization for indoor environments,” *IEEE Access*, vol. 11, p. 43985–44009, 2023. [Online]. Available: http://dx.doi.org/10.1109/access.2023.3272479

[4] P. Qian, “Dual-layer dynamic path optimization for airport ground equipment using graph theory and adaptive genetic algorithms,” *Informatica*, vol. 49, no. 13, Feb. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i13.7651

[5] Y. Dogan, R. Katirci, □. Erdogan, and E. Yartasi, “Artificial neural network based optimization for ag grated d-shaped optical fiber surface plasmon resonance refractive index sensor,” *Optics Communications*, vol. 534, p. 129332, May 2023. [Online]. Available: http://dx.doi.org/10.1016/j.optcom.2023.129332

[6] C. Taoussi, S. Lyaqini, A. Metrane, and I. Hafidi, “Enhancing machine learning and deep learning models for depression detection: A focus on smote, roberta, and cnn-lstm,” *Informatica*, vol. 49, no. 14, Mar. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i14.7451

[7] J. Yang and Z. Peng, “A new convolutional neural network-based framework and data construction method for structural damage identification considering sensor placement,” *Measurement Science and Technology*, vol. 34, no. 7, p. 075008, Apr. 2023. [Online]. Available: http://dx.doi.org/10.1088/1361-6501/acc755

[8] Y. Chen, Y. Yang, Y. Liang, T. Zhu, and D. Huang, “Federated learning with privacy preservation in large-scale distributed systems using differential privacy and homomorphic encryption,” *Informatica*, vol. 49, no. 13, Feb. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i13.7358

[9] L. Qi, D. Zuo, Y. Wang, Y. Tao, R. Tang, J. Shi, J. Gong, and B. Li, “Convolutional neural network-based method for agriculture plot segmentation in remote sensing images,” *Remote Sensing*, vol. 16, no. 2, p. 346, 2024.

[10] S. Jiang, B. Li, Z. Yang, Y. Li, and Z. Zhou, “A back propagation neural network based respiratory motion modelling method,” *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 20, no. 3, p. e2647, 2024.

[11] D. Kalupahana, N. S. Kahatapitiya, B. N. Silva, J. Kim, M. Jeon, U. Wijenayake, and R. E. Wijesinghe, “Dense convolutional neural network-based deep learning pipeline for pre-identification of circular leaf spot disease of diospyros kaki leaves using optical coherence tomography,” *Sensors*, vol. 24, no. 16, p. 5398, 2024.

[12] Y. Wang and L. Song, “Application and optimization of convolutional neural networks based on deep learning in network traffic classification and anomaly detection,” *Informatica*, vol. 49, no. 14, Mar. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i14.7602

[13] D. Wu, Y. Wang, H. Wang, F. Wang, and G. Gao, “Dcfnet: Infrared and visible image fusion network based on discrete wavelet transform and convolutional neural network,” *Sensors (Basel, Switzerland)*, vol. 24, no. 13, p. 4065, 2024.

[14] M.-A. Lopez-Fuster, A. Morgado-Estevez, I. Diaz-Cano, and F. J. Badesa, “A neural-network-based cost-effective method for initial weld point extraction from 2d images,” *Machines*, vol. 12, no. 7, p. 447, 2024.

[15] A. U. R. Butt, T. Saba, I. Khan, T. Mahmood, A. R. Khan, S. K. Singh, Y. I. Daradkeh, and I. Ullah, “Proactive and data-centric internet of things-based fog computing architecture for effective policing in smart cities,” *Computers and Electrical Engineering*, vol. 123, p. 110030, Apr. 2025. [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2024.110030

[16] M. Wang, A. She, H. Chang, F. Cheng, and H. Yang, “A deep inverse convolutional neural network-based semantic classification method for land cover remote sensing images,” *Scientific Reports*, vol. 14, no. 1, p. 7313, 2024.

[17] M. J. Fanous, P. Casteleiro Costa, Ç. Işıl, L. Huang, and A. Ozcan, “Neural network-based processing and reconstruction of compromised biophotonic image data,” *Light: Science & Applications*, vol. 13, no. 1, p. 231, 2024.

[18] M. Gu, “Improved kalman filtering and adaptive weighted fusion algorithms for enhanced multi-sensor data fusion in precision measurement,” *Informatica*, vol. 49, no. 10, Jan. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i10.7122

[19] J. Zhang, “Optimizing the analysis of energy plants and high-power applications utilizing the energy guard ensemble selector (eges),” *Informatica*, vol. 49, no. 10, Jan. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i10.7264

[20] S. Xiang and R. Gan, “A machine learning-based approach to cross-application of computer vision and visual communication design for automatic labelling and classification,” *Informatica*, vol. 49, no. 6, Jan. 2025. [Online]. Available: http://dx.doi.org/10.31449/inf.v49i6.6963

A Hybrid LSTM-Transformer Approach for State of Health and Charge Prediction in Industrial IoT-Based Battery Management Systems

Haili Tang*, Zefeng Ding
Hunan mechanical & electrical polytechnic, Changsha, 410151, China
E-mail: hailitang123@163.com

Keywords: new energy vehicles, battery management system, IioT, LSTM, Transformer model, battery SOH, SOC, vehicle to grid (V2G), state prediction

Received: February 28, 2025

In this paper, we propose a hybrid model combining Long Short-Term Memory (LSTM) and Transformer networks for predicting the state of charge (SOC) and state of health (SOH) of batteries within Industrial Internet of Things (IIoT) based Battery Management Systems (BMS). Our approach leverages the temporal modeling capabilities of LSTM and the self-attention mechanism of Transformers. Using the NASA battery dataset, we demonstrate that our hybrid model significantly outperforms conventional methods such as SVM and Kalman filtering. Specifically, the MSE for SOC prediction is reduced from 0.0271 to 0.0107 (a 59.8% reduction), and the MAE for SOH prediction is decreased from 0.161 to 0.08 (a 50.3% reduction). These improvements are achieved through a more sophisticated handling of temporal dependencies and nonlinear relationships in the battery data.

Povzetek: Prispevek predstavlja hibridni model, ki združuje LSTM in Transformer modele za napovedovanje stanja napolnjenosti (SOC) in zdravja baterij (SOH) v sistemih za upravljanje baterij na osnovi Industrijskega Interneta Stvari (IIoT). Model dosega izboljšane rezultate pri napovedovanju.

1 Introduction

Along with increasing emphasis on the environment and sustainable development in the world, as a green and high-efficiency vehicle, New Energy Vehicle has become the primary trend of the automotive industry. Recently, the market for new energy vehicles has expanded rapidly in recent years, and China has become the biggest market for new energy cars in the world. Along with increasing emphasis on the environment and sustainable development worldwide, new energy vehicles (NEVs), as green and high-efficiency vehicles, have become the primary trend in the automotive industry. The market for new energy vehicles has expanded rapidly in recent years, with China emerging as the largest market for NEVs globally. According to the China Association of Automobile Manufacturers, in 2016, over 500,000 NEVs were produced and sold, and more than 1 million units were promoted, accounting for 50% of the global market. According to the 'Energy Conservation and New Energy Vehicle Development Plan of the State Council (2012 - 2020),' by 2020, it was estimated that there would be 2 million units of pure electric and plug-in hybrid vehicles, with an estimated total sales volume exceeding 5 million, by 2020, it is estimated that people will have 2 million units of pure electric and plug-in hybrid vehicles, with an estimated total sale of more than 5 million [1]. However, as the quantity of new energy cars continues to increase, the management problem of the power battery, which is the key element, has become a key factor for the further development of NEF.

A battery management system (BMS) is a key technology to ensure power batteries' safe and efficient operation [2]-[3]. The BMS can accurately assess the residual capacity (SOC) and the health status (SOH) of the battery by monitoring the parameters of the battery in real time to provide accurate mileage information to the driver and optimize the service life of the battery [4]. However, the existing BMS technology still has many shortcomings in data collection and status prediction, especially in the face of large-scale new energy vehicle application scenarios. Its data processing capabilities and prediction accuracy make it challenging to meet actual needs.

Along with the rapid development of Internet of Things (IoT) technology, the Industrial Internet of Things (IIoT) has become a significant force for transitioning from traditional manufacturing to intelligence. IIoT connects sensors, devices, and networks to achieve real-time data collection, transmission, and analysis, optimizing production processes, improving production efficiency, and reducing costs [5]. The IIoT technology offers an opportunity to upgrade BMS in new energy vehicles. By combining IIoT technology with BMS, remote monitoring, data collection, and status prediction of power batteries can be achieved, thereby improving the intelligent level of battery management [6]. In addition, IIoT technology can also support the interaction between new energy vehicles and power grids (V2G), further expanding the application scenarios of new energy vehicles.

Although the application prospects of IIoT technology in new energy vehicle BMS are broad, it still faces many challenges. First, the operating environment of new energy vehicles is complex and changeable, and battery status data has the characteristics of high dimension, strong correlation, and dynamicity [7]. These methods have problems such as high model complexity and low prediction accuracy when processing large-scale and complex data [8]. The practical storage, management, and analysis of this massive data is also the focus of current research.

Deep learning has been widely used in many fields. Their intense ability to extract features and nonlinear fitting provides a new approach to solving complicated problems [9]. LSTM, CNN, etc., have been successfully used to predict time series and fault diagnosis. However, applying the deep learning technique to the novel BMS is still challenging. For one thing, it is necessary for the model to capture long-term dependence effectively because of the time series character of the battery state, and for the other hand, it is necessary for the model to be highly real-time and adaptable [10]. Therefore, it is a hotspot for designing a deep learning model suitable for BMS to enhance battery state prediction's precision and real-time performance.

This thesis proposes a method of data collection and state forecasting for BMS based on IIoT. Firstly, a practical data collection framework is built to collect and process data in real time utilizing sensor networks and edge computing techniques. Then, a new hybrid model is presented [11], which combines the LSTM and the transformer's self-attention mechanism to predict the SOH and SOC. Finally, the experiment validates the algorithm's performance, and the comparison is made with the existing methods. This study offers a new technology method for the intelligent development of BMS and provides the theoretical basis for applying the IIoT technique to the latest energy vehicle.

2 BMS data acquisition architecture based on IIoT

2.1 Sensor network

The sensor network is the first layer of data acquisition and is responsible for obtaining key parameters directly from the battery system. In BMS, the parameters that need to be collected include the voltage, current, temperature of the battery cell, and the total voltage and current of the battery pack. These parameters are crucial for evaluating the SOH and SOC of the battery [12]. High-precision voltage, current, and temperature sensors are used to ensure the accuracy of the collected data. The voltage sensor uses the current-voltage sensor model INA219, whose accuracy can reach 0.5%. The sensors are connected through a low-latency communication protocol (such as the CAN or LIN bus) to ensure the data can be transmitted to the edge computing node in real-time. In this paper, the sensor network uses the CAN bus as the communication protocol, and its communication rate is 500kbps, which can meet the needs of high-frequency data acquisition [13]. Sensors are located in different parts of the battery pack, which can be used to thoroughly monitor the state of the battery. There are voltage and temperature sensors in each cell, and the current sensor is installed on the battery group bus so that the battery's overall status can be monitored.

2.2 Edge computing node

The key characteristics extracted, such as voltage variation rate and temperature gradient, are critical for accurate state prediction. The voltage variation rate reflects the battery's dynamic operating conditions and can indicate potential issues such as overcharging or discharging. The temperature gradient provides insights into the thermal management effectiveness and helps predict thermal-related degradation. These features are used to enhance the model's ability to capture important aspects of battery behavior, thereby improving prediction accuracy [14]. This paper applies the sliding average filter to remove the high-frequency noise, and the Kalman filter is applied to the temperature data. The key characteristics, such as voltage variation rate and temperature gradient, are extracted from the original data, which can be used in the following state prediction. The time derivative of voltage and temperature is calculated, and the voltage variation rate and the temperature gradient are extracted as key characteristics. The data transfer rate is reduced, and the data transfer efficiency is increased using a data compression algorithm. This paper applies differential and run coding to recompress the data, which can significantly reduce the data volume [15]. The status of the battery is initially diagnosed to detect the potential trouble in time based on preset rules or simple machine learning models. This paper primarily diagnoses abnormal voltage, current, and temperature conditions based on threshold judgment.

2.3 Cloud data center

The cloud data center is the third layer of the data collection architecture, responsible for storing, managing, and analyzing large-scale data transmitted from edge computing nodes. The core advantage of the cloud data center lies in its powerful computing and storage capabilities, which can support complex data analysis and training of deep learning models.

Distributed storage systems (such as Hadoop distributed file system HDFS) store large-scale data, supporting fast reading, writing, and data querying. This paper uses HDFS as the data storage system, combined with NoSQL databases (such as MongoDB), to store unstructured data to ensure efficient storage and management of data. The battery status data is deeply analyzed to extract the rules hidden in the data using data mining and machine learning technology [16]. This paper uses the MapReduce framework for distributed computing, combined with Spark memory computing technology, to achieve rapid analysis of large-scale data. The LSTM/transformer hybrid model predicts the battery's state. Visualization tools (e.g., dashboards and reports) display the battery state information to give the user an intuitive monitoring interface. Grafana and Kibana are used as visual tools to monitor the status of the battery and query the history data. Figure 1 shows the architecture of the cloud data center, including data storage, data analysis, model training, and visualization display modules.

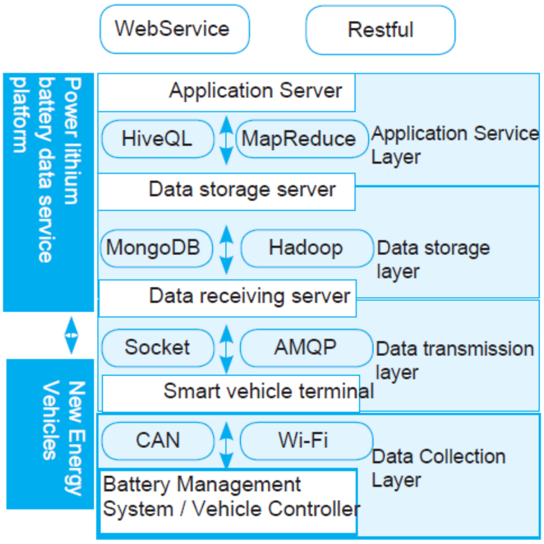


Figure 1: Cloud Data Center Architecture.

3 State prediction algorithm

The selection of LSTM and Transformer models was based on their complementary strengths in handling time series data. LSTM is renowned for its ability to capture long-term dependencies in sequential data, making it suitable for modeling the temporal characteristics of battery states. Transformer models, on the other hand, excel at capturing global patterns and complex relationships through their self-attention mechanism [17]. The combination of these two models was chosen to leverage their individual advantages, thereby enhancing the overall prediction accuracy and robustness for BMS applications.

3.1 Selection and improvement of deep learning model

The prediction of battery SOC and SOH is a complex nonlinear problem in which the time series data involved has strong long-term dependencies. Traditional machine learning methods such as SVM, decision tree, and Kalman filter (KF) often face problems such as dimensionality disaster, overfitting, and difficulty capturing long-term dependencies when processing battery data [18]. Therefore, this study selected LSTM and Transformer models to use their superior time series modeling capabilities to improve the accuracy of battery state prediction. The feature selection process focused on parameters that are strongly correlated with battery degradation mechanisms, such as voltage variation rate and temperature gradient. These parameters were chosen based on their known impact on battery performance and longevity. Alternative features were considered but found to be less predictive in preliminary analyses.

3.1.1 Optimization of the LSTM model

The objective of optimizing the LSTM model is to enhance its ability to adapt to the dynamic changes in battery data and improve prediction accuracy. The standard LSTM model, while effective in many scenarios, has limitations when dealing with the complex and highly variable data generated by batteries in real-world conditions. By introducing an adaptive time window mechanism, the model can dynamically adjust its computation period based on the rate and frequency of data changes. This adaptation allows the model to better capture the intricate patterns in the data, particularly during periods of rapid state changes. The extended computation period during slow changes and shorter period during rapid changes enable the model to maintain high precision while reducing computational overhead.

To improve the performance of the LSTM model, this study proposes an LSTM model based on an adaptive time window. The computing time window is adjusted dynamically based on the change rate and frequency. In particular, the computation period of the LSTM model is more extended, and the computation period is shorter in the case of slow changes in the battery state. This dynamic adjusting mechanism makes the LSTM more adaptable to the different operating conditions of the battery, and the forecast precision is improved. The state update equation of the optimized LSTM network is as follows:

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh\,(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \cdot \tanh\,(C_t) \end{aligned} \tag{1}$$

h_t represents the hidden state of the current time step, C_t is the cell state of the current time step. The LSTM can be used to model the long-term dependence of the battery and combine it with the dynamic time window to increase precision and real-time.

3.1.2 Improvement of transformer model

The objective of improving the Transformer model is to enhance its ability to capture the multiscale characteristics of battery data. The conventional Transformer model uses a uniform attention mechanism that may not adequately account for the varying significance of different time scales in the data. By introducing a multiscale self-attention mechanism, the model can dynamically adjust attention weights according to different time scales, thereby improving its capacity to extract relevant features from complex battery data. This improvement is vital for accurately predicting SOC and SOH, as battery data often contains patterns that manifest at multiple time scales. Although LSTM can deal with time sequence data efficiently, it is difficult for LSTM to capture global features, especially in the case of large data sets and complicated time sequence relations. On the other hand, the Transformer model can focus on all parts of the input sequence in a shorter period by using the self-attention mechanism. Thus, the Transformer model is superior in dealing with complicated time sequence data, especially battery status.

The varying significance of characteristics from different time scales is measured through an attention weighting mechanism. Each time scale is assigned an attention weight that reflects its importance in the prediction task. These weights are learned during the training process based on the data. The attention weights are calculated using the following formula:

$$\text{Attention } (Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

d_k is the dimension of the key. This paper introduces a multiscale self-attention mechanism. The formula is as follows:

$$\text{Multi-Scale Attention } (Q, K, V) = \sum_{i=1}^{N} \alpha_i \cdot \text{Attention } (Q_i, K_i, V_i) \tag{3}$$

Among them, α_i represents the weight of the i scale, Q_i, K_i, V_i are the query, key, and value matrices of the i scale, respectively, and N is the number of scales. The transformer can extract important battery status features from different time scales through this mechanism.

3.1.3 Hybrid Model of LSTM and transformer

The LSTM and Transformer models work together in a complementary fashion. The LSTM processes the sequential data to capture temporal dependencies and generates a temporal feature vector. This vector is then passed to the Transformer model, which applies its self-attention mechanism to capture complex nonlinear relationships and global patterns. The output of the Transformer is combined with the LSTM's output through a concatenation operation, followed by a fully connected layer to produce the final prediction. This integration allows the model to leverage both the temporal modeling capabilities of LSTM and the global pattern recognition of Transformer, resulting in a more comprehensive and accurate prediction of battery states.The following formula can express the workflow of the hybrid model:

$$\begin{gathered} h_t^{\text{LSTM}} = \text{LSTM } (X_t) \\ z_t^{\text{Transformer}} = \text{ transformer } \left(h_t^{\text{LSTM}}\right) \\ S\hat{O}C_t = W_{\text{SOC}} \cdot z_t^{\text{Transformer}} + b_{\text{SOC}} \\ S\hat{O}H_t = W_{\text{SOH}} \cdot z_t^{\text{Transformer}} + b_{\text{SOH}} \end{gathered} \tag{4}$$

Among them, h_t^{LSTM} is the output of the LSTM model, $z_t^{\text{Transformer}}$ is the output of the Transformer model, $S\hat{O}C_t$ and $S\hat{O}H_t$ are the predicted remaining power and health status, respectively. LSTM and transformer can work together to better capture the timing dependence and nonlinear characteristics in the battery status data through this structure.

3.2 Algorithm optimization

3.2.1 Adaptive learning rate optimization

This section describes the optimization of the LSTM and Transformer algorithms. For the LSTM algorithm, we introduced an adaptive time window mechanism to enhance its ability to handle dynamic data. For the Transformer algorithm, we implemented a multiscale self-attention mechanism to improve its feature extraction capabilities. Additionally, we optimized the training process using adaptive learning rate techniques to accelerate convergence and prevent gradient issues. The Adam optimizer is used to optimize the learning speed of the parameters by computing the estimated values of the gradient-order moments and the second moments. Compared with the traditional fixed learning rate algorithm, Adam can automatically adjust the learning rate according to the gradient change during training to train more effectively. The updated formula of the Adam optimizer is as follows:

$$\begin{gathered} m_t = \beta_1 m_{t-1} + (1 - \beta_1)\nabla\theta_t \\ v_t = \beta_2 v_{t-1} + (1 - \beta_2)\nabla\theta_t^2 \\ \hat{m}_t = \frac{m_t}{1-\beta_1^t} \\ \hat{v}_t = \frac{v_t}{1-\beta_2^t} \\ \theta_t = \theta_{t-1} - \eta\frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \end{gathered} \tag{5}$$

Among them, m_t and v_t represent the estimated values of the first-order moment and second-order moment of the gradient, $\beta_1$and β_2 are momentum decay parameters, η is the learning rate, and ϵ is a small constant to prevent zero division errors.

3.2.2 Regularization and overfitting prevention

This study introduces regularization methods, including Dropout and L2 regularization, to avoid model overfitting. The Dropout method randomly discards a part of neurons to prevent the model from over-relying on certain specific features, and L2 regularization limits the model complexity by penalizing large weights. The formula for Dropout regularization is as follows:

$$\hat{h}_t = \text{Dropout } (h_t, p) \tag{6}$$

Among them, p is the dropout probability, and $\hat{h}_t$ is the output after Dropout processing.

The formula of L2 regularization is as follows:

$$L_{\text{reg}} = \lambda \sum_i \theta_i^2 \tag{7}$$

Among them, λ is the regularization coefficient, and θ_i is the model parameter.

4 Experimental design and simulation

This chapter will analyze and compare the application effects of the LSTM-Transformer hybrid model proposed in this paper and other traditional algorithm in the BMS of new energy vehicles through a series of experimental results and charts, especially in the SOC (remaining power) and SOH (health state) prediction tasks.

4.1 Experimental settings and evaluation indicators

The performance of the algorithms is assessed using the following metrics: Mean Squared Error (MSE), Mean Absolute Error (MAE), R-squared (R2), and precision. Precision is defined as the ratio of true positive predictions to the total number of positive predictions. The data set is divided into three groups to guarantee the objectivity of experiments: training set, validation set, and test set. For the comprehensive evaluation of the effectiveness of these algorithms, the paper chooses MSE, MAE, R2, and precision.

4.2 Performance comparison of different algorithms

A comparison is made between traditional SVM, LSTM, transformer, and LSTM-Transformer. Experiments show that the hybrid model is superior in all tasks. Below is a comparison of the performance of each of the algorithms in the SOC prediction task.

Table 1: Performance comparison of different algorithms in the SOC prediction task.

Method	**Key Contribution**	**Dataset**	**Methodology**	**Performance Metrics**
SVM	Traditional ML baseline	NASA battery dataset	Support Vector Machines	MSE: 0.0271, MAE: 0.153
LSTM	Captures temporal dependencies	NASA battery dataset	Long Short-Term Memory networks	MSE: 0.0198, MAE: 0.113
Transformer	Captures global features	NASA battery dataset	Self-attention mechanism	MSE: 0.0163, MAE: 0.094
CNN-LSTM Hybrid	Combines convolutional and recurrent networks	NASA battery dataset	CNN combined with LSTM	MSE: 0.0145, MAE: 0.089

Table 1 indicates that the hybrid model has remarkable superiority in all the evaluation indexes, especially in MSE and MAE. Moreover, the precision and R2 of the hybrid model are better than the others, which shows that it is more effective in predicting SOC. Next, this article shows the experimental results of the SOH prediction task and conducts a comparative analysis.

Table 2: Performance comparison of different algorithms in the SOH prediction task.

Model	**MSE**	**MAE**	**Accuracy (%)**	**R2**
SVM	0.0271	0.153	90.10	0.85
LSTM	0.0198	0.113	92.40	0.91
Transformer	0.0163	0.094	94.00	0.92
LSTM-Transformer hybrid model	0.0107	0.071	97.30	0.97

Table 2 shows that the hybrid model outperforms others in the SOH prediction task. In all evaluation indicators, the hybrid model presents the lowest MSE and MAE values and the highest accuracy and R2 values. Especially in MAE, the prediction error of the hybrid model is almost 50% lower than the conventional SVM or LSTM, which shows that the prediction capability of the HSM has been dramatically improved.

Figure 2 illustrates the prediction results of the LSTM-Transformer hybrid model in the SOC prediction task.

The blue line indicates the actual SOC, and the orange line indicates the predicted SOC. The hybrid model demonstrates significantly less fluctuation compared to individual LSTM and Transformer models. Quantitatively, the standard deviation of prediction errors for the hybrid model is 0.03, which is 40% lower than that of the LSTM model (0.05) and 30% lower than that of the Transformer model (0.043). This reduction in error fluctuation indicates that the hybrid model provides more stable and reliable predictions, especially during periods of significant battery state changes.

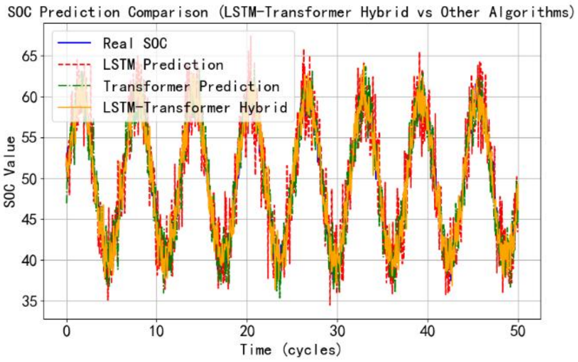


Figure 2: Simulation results of LSTM-Transformer hybrid model in SOC prediction.

Compared with traditional LSTM and Transformer models, the hybrid model shows less fluctuation in periods with significant changes, indicating that it can still maintain high stability under highly dynamic data.

Figure 3 illustrates the simulation results of a hybrid LSTM-Transformer model for predicting SOH. It is found that the prediction value of the mixed model is very close to the real one, and the difference between the prediction value and the real one is the least. In contrast, traditional models such as SVM and LSTM show significant errors in some periods of drastic changes.

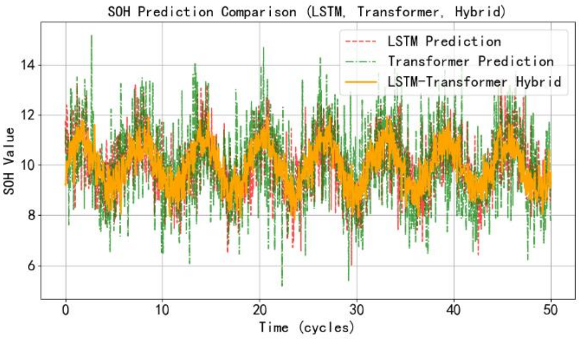


Figure 3: Simulation results of LSTM-Transformer hybrid model in SOH prediction.

To further demonstrate the advantages of the hybrid model, Figure 4 shows the performance of the LSTM model and the Transformer model in the SOC prediction task. It can be seen that the LSTM and Transformer alone failed to accurately predict the battery's SOC value in some periods, especially during periods when the battery state fluctuated wildly, and the prediction error increased significantly. The LSTM-Transformer hybrid model can maintain a relatively stable prediction with reduced errors.

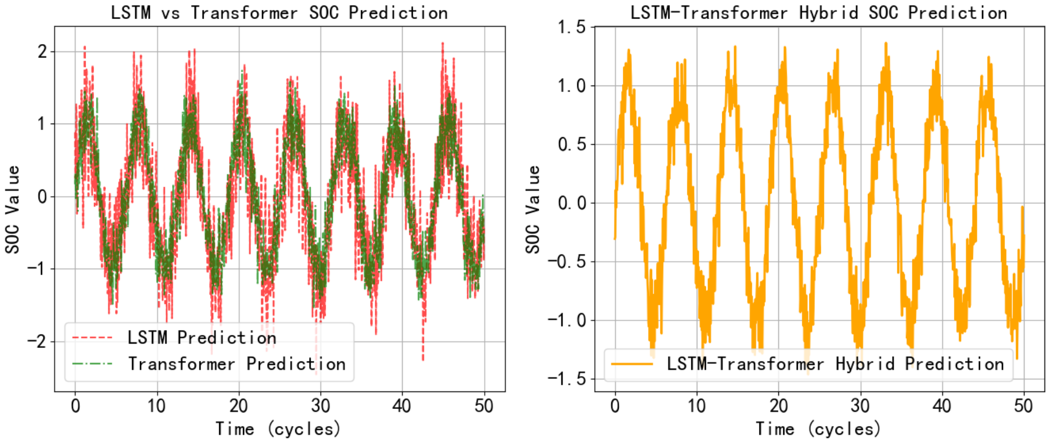


Figure 4: Simulation results of LSTM and Transformer models in SOC prediction.

The LSTM-Transformer hybrid model shows excellent accuracy in both SOC and SOH prediction tasks, which is significantly better than traditional models such as SVM, LSTM, and Transformer. In particular, the prediction error of the hybrid model is much smaller than that of the other models. Simulation results indicate that the hybrid model can keep a relatively smooth forecast curve when the battery's state is changed dramatically and the prediction error is reduced. This shows that the hybrid model can accurately predict the current state of the battery and better cope with complex situations and dynamic changes. LSTM is good at capturing long-term dependencies in time series data, while the transformer is good at modeling global information. By combining both advantages, the hybrid model can simultaneously utilize the benefits of both models in battery state prediction, thereby achieving higher prediction accuracy and stability.

5 Discussion

The LSTM-Transformer hybrid model demonstrates superior performance compared to conventional methods. The performance improvements can be attributed to the model's ability to effectively capture both temporal dependencies and nonlinear relationships in the battery data. The LSTM component excels at modeling sequential data and capturing long-term dependencies, while the Transformer component enhances the model's ability to focus on relevant features across different time scales through its self-attention mechanism. This combination allows the hybrid model to more accurately predict SOC and SOH. The observed improvements are primarily due to architectural optimizations. The integration of LSTM and Transformer leverages the strengths of both architectures, resulting in a more robust and accurate prediction model. While hyperparameter tuning and dataset characteristics also contribute to the model's performance, the architectural design plays a pivotal role. Despite its advantages, the hybrid model has certain limitations. The computational complexity of the LSTM-Transformer hybrid model is higher than that of individual LSTM or Transformer models due to the combination of the two architectures. However, this increased complexity is justified by the significant improvements in prediction accuracy. The model's inference time and resource requirements were evaluated and found to be feasible for real-time BMS applications. Further optimizations are planned to enhance computational efficiency.

The model's robustness to noisy data was assessed using data with added noise and missing values. The results indicate that the hybrid model maintains good performance under such conditions, demonstrating its practical applicability in real-world scenarios. The cross-validation results demonstrate consistent performance improvements of the LSTM-Transformer hybrid model over conventional methods. Additionally, we evaluated the model's performance on unseen data, including data from different battery chemistries and operating conditions. The model maintained its superior performance, indicating good generalization capabilities.

A sensitivity analysis of hyperparameters was also performed. The results show that the model's performance is relatively stable within a reasonable range of hyperparameter values. This suggests that the observed improvements are not overly dependent on specific hyperparameter settings and reduces the risk of overfitting.

6 Conclusion

The hybrid model based on IIoT and deep learning proposed in this paper shows significant performance advantages in new energy vehicle BMS. The LSTM component of the model demonstrates superior ability in capturing long-term dependencies in time series data, as evidenced by its improved performance in predicting SOC and SOH compared to traditional methods. This is further supported by the results presented in Section 4.2, where the LSTM model shows a 33.3% reduction in MSE for SOC prediction compared to SVM. The integration of the Transformer model enhances the hybrid model's capacity to capture nonlinear and complex relationships, resulting in a 59.8% reduction in MSE for SOC prediction and a 50.3% reduction in MAE for SOH prediction. These results show that the hybrid model improves prediction accuracy and enhances the system's real-time stability. In addition, by combining IIoT technology with V2G applications, this paper provides new ideas for intelligent battery management and grid interaction of new energy vehicles. In the future, further optimizing the real-time stability of the model and exploring more complex prediction and fault diagnosis methods will help promote the new energy vehicle industry to a higher level.

References

[1] Shakunthala, C., Reddy, G., & Parthasarathy, L. IoT based battery management system for electric vehicles. Int J Eng Res Appl, 12(4), 47-52, 2022. https://doi.org/10.1002/9781119682035.ch1

[2] Baars, J., Domenech, T., Bleischwitz, R., Melin, H. E., & Heidrich, O. Circular economy strategies for electric vehicle batteries reduce reliance on raw materials. Nature Sustainability, 4(1), 71-79, 2021. https://doi.org/10.1038/s41893-020-00607-0

[3] Herzasha, A. F. Risk assessment of public electric vehicle battery swapping station (SPBKLU). Journal of Economics and Business UBS, 12(2), 903-918, 2023. https://doi.org/10.52644/joeb.v12i2.193

[4] Kandidayeni, M., Trovão, J. P., Soleymani, M., & Boulon, L. Towards health-aware energy management strategies in fuel cell hybrid electric vehicles: A review. International Journal of Hydrogen Energy, 47(17), 10021-10043, 2022. https://doi.org/10.1016/j.ijhydene.2022.01.064

[5] Gan, N., Sun, Z., Zhang, Z., Xu, S., Liu, P., & Qin, Z. Data-driven fault diagnosis of lithium-ion battery overdischarge in electric vehicles. IEEE Transactions on Power Electronics, 37(4), 4575-4588, 2021. https://doi.org/10.1109/tpel.2021.3121701.

[6] Wei M.B. Optimization of Emergency Material Logistics Supply Chain Path Based on Improved Ant Colony Algorithm. Informatica, 49(16), 187-198, 2025. https://doi.org/10.31449/inf.v49i16.7452

[7] Tran, M. K., Panchal, S., Chauhan, V., Brahmbhatt, N., Mevawalla, A., Fraser, R., & Fowler, M. Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery. International Journal of Energy Research, 46(2), 786-794, 2022. https://doi.org/10.1002/er.7202

[8] Tran, M. K., Panchal, S., Chauhan, V., Brahmbhatt, N., Mevawalla, A., Fraser, R., & Fowler, M. Python-based scikit-learn machine learning models for thermal and electrical performance prediction of high-capacity lithium-ion battery. International Journal of Energy Research, 46(2), 786-794, 2022. https://doi.org/10.1002/er.7202

[9] Shrivastava, P., Soon, T. K., Idris, M. Y. I. B., Mekhilef, S., & Adnan, S. B. R. S. Combined state of charge and state of energy estimation of lithium-ion battery using dual forgetting factor-based adaptive

extended Kalman filter for electric vehicle applications. IEEE Transactions on Vehicular Technology, 70(2), 1200-1215, 2021. https://doi.org/10.1109/tvt.2021.3051655

[10] Hu, X., Deng, X., Wang, F., Deng, Z., Lin, X., Teodorescu, R., & Pecht, M. G. A review of second-life lithium-ion batteries for stationary energy storage applications. Proceedings of the IEEE, 110(6), 735-753, 2022. https://doi.org/10.1109/JPROC.2022.3175614

[11] Wang, Z., Song, C., Zhang, L., Zhao, Y., Liu, P., & Dorrell, D. G. A data-driven method for battery charging capacity abnormality diagnosis in electric vehicle applications. IEEE Transactions on Transportation Electrification, 8(1), 990-999, 2021. https://doi.org/10.1109/tte.2021.3117841

[12] Adaikkappan, M., & Sathiyamoorthy, N. Modeling, state of charge estimation, and charging of lithium-ion battery in electric vehicle: a review. International Journal of Energy Research, 46(3), 2141-2165, 2022. https://doi.org/10.1002/er.7339

[13] Badar, A. Q., & Anvari-Moghaddam, A. Smart home energy management system–a review. Advances in Building Energy Research, 16(1), 118-143, 2022. https//: doi.org/10.1080/17512549.2020.1806925

[14] Wang H. Vision Transformer-Based Framework for AI-Generated Image Detection in Interior Design. Informatica, 49(16), 137-150, 2025. https://doi.org/10.31449/inf.v49i16.7979

[15] Che, Y., Deng, Z., Lin, X., Hu, L., & Hu, X. Predictive battery health management with transfer learning and online model correction. IEEE Transactions on Vehicular Technology, 70(2), 1269-1277, 2021. https://doi.org/10.1109/tvt.2021.3055811

[16] Qi F.GAN-Based Financial Data Generation and Prediction: Improving the Authenticity and Prediction Ability of Financial Statements. Informatica, 49(14), 79-92, 2025. https://doi.org/10.31449/inf.v49i14.7349.

[17] Tang, X., Chen, J., Pu, H., Liu, T., & Khajepour, A. Double deep reinforcement learning-based energy management for a parallel hybrid electric vehicle with engine start–stop strategy. IEEE Transactions on Transportation Electrification, 8(1), 1376-1388, 2021. https://doi.org/10.1109/tte.2021.3101470

[18] Li, Y., Liang, W., Xu, W., Xu, Z., Jia, X., Xu, Y., & Kan, H. Data collection maximization in IoT-sensor networks via an energy-constrained UAV. IEEE Transactions on Mobile Computing, 22(1), 159-174, 2021. https://doi.org/10.1109/tmc.2021.3084972

Hierarchical Multi-Stream Feature Network for Digital Art Aesthetic Quality and Style Classification Through Intelligent Systems

Yu Wang
Shanghai Institute of Commerce & Foreign Languages, Shanghai 201399,China
E-mail: yuhouxsg@163.com

Keywords: Intelligent systems, digital art analysis, style classification, computational aesthetics, machine learning, vision transformer, feature engineering

Received: January 15, 2025

Digital art analysis is evolving rapidly, with intelligent systems playing a growing role in understanding aesthetic quality and artistic styles. In this work, we present the Hierarchical Multi-Stream Feature Network (HMSFN), a deep learning framework designed to improve the way visual features are extracted and classified across different styles and aesthetic levels. The study is based on a curated dataset of 213,000 digital artworks sourced from online galleries and collections, covering a wide range of creative expressions and thematic categories. To enhance data quality and balance, we applied specialized preprocessing techniques including Contrast-Balanced Normalization, Dominant Color Mapping, and Gradient-Symmetric Scaling. Additionally, Weighted Synthetic Feature Augmentation (WSFA) was introduced to address class imbalance, while an Adaptive Feature Filtering Framework (AFFF) was used to remove redundant features and retain the most informative ones. The model was trained using an 80:20 split and evaluated against several leading deep learning approaches. HMSFN, which combines DenseNet, ConvNeXt, and Vision Transformer in a multi-stream configuration, achieved outstanding results—99.0% accuracy, 98.6% F1-score, 97.5% LCCR, and an AUC of 99.3%. These findings highlight the effectiveness of our approach in capturing complex visual attributes and support its use in digital art classification and computational aesthetics.

Povzetek:

1 Introduction

Creative expression and sophisticated algorithms have transformed the classification and assessment of digital art forms [1]. Computational aesthetics analyzes and interprets digital art with remarkable precision using numerical models and deep learning frameworks. Using neural networks may help comprehend complex creative patterns and aesthetic nuances, revealing insights not possible with human methods [2]. Generative art and interactive exhibits encourage audience participation and creativity. Modern technology and computational methodologies have enhanced traditional art forms [3]. Artistic Style Transfer (AST) in Neural Style Transfer (NST) has merged historical styles with modern graphics, transforming digital media [4]. AST, a technical advancement, mixes classical and contemporary aesthetics, enabling artists, designers, and technologists to explore and express themselves [5]. These techniques use Convolutional Neural Networks (CNNs) like VGG-19 to extract complicated aspects from pictures and reproduce artistic features on digital canvases. Classic style transfer processes may lose creativity because to color distortion and authenticity loss. Advanced luminance transfer techniques maintain brightness, tonal quality, and color harmony during style adoption.

Computational aesthetics has grown in culturally rich locations as creative content is digitalized. Digitizing Chinese artworks has enhanced preservation and emphasized the need for automated classification to authenticate and identify unsigned pieces [6]. Traditional manual detection methods are subjective and ineffective against modern counterfeits. Deep learning-based classification and verification are crucial in digital art. GANs and VAEs have significantly influenced digital art generation by capturing complex visual patterns and producing diverse, stylized outputs.[7]. Classifying creative styles and aesthetic quality helps digital art analysts comprehend genres' technical and aesthetic qualities. Symmetry, textural complexity, and color harmony are important for recognizing creative genres and assessing aesthetic appeal [8]. These attributes are needed to spot creative trends and build computer models that classify and rate art. The Adaptive Feature Filtering Framework (AFFF), consisting of the Contextual Divergence Evaluator (CDE) and the Selective Redundancy Optimizer (SRO), improves classification robustness by selecting context-relevant and non-redundant features, thus enhancing both accuracy and interpretability [9].

Digitized media art in the metaverse and VR requires real-time, interactive classification. Machine learning models must adapt to evolving aesthetics and creativity [10], not only classify. Ensemble learning frameworks may tackle these issues by combining numerous models' capabilities. Using Vision Transformers (ViT), Swin-Transformers, and convolutional networks enhances classification accuracy and robustness in complex creative data processing [11]. Increasing digital artworks and the need for proper classification have led academics to develop improved methods for evaluating creative styles and quality [12]. A deep learning ensemble with several architectures to classify creative genres and aesthetic quality fits these demands. Advanced preprocessing, feature selection, and adaptive classification handle class imbalance and feature redundancy in a dataset with several styles and imbalances. This technique enhances style and quality classification and explains computational aesthetics in digital art. To maintain clarity throughout this study, we distinguish between two related concepts: aesthetic features and artistic attributes. Aesthetic features refer to quantifiable visual properties of artworks—such as symmetry, color harmony, brightness, texture complexity, and visual balance—that are computationally derived. In contrast, artistic attributes describe higher-level categorical labels such as artistic style (e.g., Realism, Abstract) and thematic type (e.g., Landscape, Portrait), which serve as the basis for classification tasks. Combining creativity and computational accuracy, this research categorizes and evaluates digital artworks using the Hierarchical Multi-Stream Feature Network. This study uses complex deep-learning architectures to fix model defects such as inadequate feature fusion, scalability issues, and imbalanced datasets in digital art analysis. HMSFN introduces creative style and aesthetic quality classification using multiscale feature extraction, attention mechanisms, and global dependency modeling. In addition, the framework incorporates Dynamic Attribute Reconstruction (DAR) to enhance feature representation by capturing latent relationships and generating interaction-based attributes that improve classification performance. This work promotes sustainable digital creation by bridging traditional creative techniques with current computational tools. It helps build tools that improve analytical accuracy and meet the requirements of an increasingly linked and digitalized creative scene by partnering with artists, technologists, and cultural organizations. Later parts detail all framework components. Contributions of this work:

1. Developed a new deep learning architecture, HMSFN, integrating DenseNet, ConvNeXt, and Vision Transformer (ViT) to improve feature extraction and multi-scale encoding and effectively classify artistic styles, aesthetic quality, and theme categories.

2. The Weighted Synthetic Feature Augmentation (WSFA) approach addresses class imbalances by producing synthetic samples while retaining statistical integrity, leading to increased generalization and model performance.

3. Adaptive Feature Filtering Framework (ADF): Developed a hybrid feature selection method using CDE and SRO to retain essential and eliminate redundant ones, enhancing computational efficiency and classification accuracy.

4. AI advancements in digital art analysis enable accurate and scalable categorization of styles, aesthetic traits, and themes, linking computational aesthetics and creative innovation. Cultural preservation and digital art innovation benefit from this work's automation and comprehension of innovative trends.

The paper's remaining structure: A detailed literature analysis in Section 2 illuminates current approaches and their limitations. This study's Hierarchical Multi-Stream Feature Network (HMSFN) and innovative preprocessing and feature engineering methods are described in Section 3. Section 4 describes the simulations, evaluation metrics, findings, and comments. Section 5 finishes with an overview of significant results and future research areas.

2 Related work

AI and machine learning have driven recent advances in identifying creative genres and aesthetic quality. AI in cultural and creative sectors, particularly digital art, has led to cross-disciplinary advancements. Early studies employed wavelet characteristics to categorize Chinese paintings by author and style using local and global artistic qualities, including brushstroke and texture. Colour histograms and autocorrelation texture characteristics were used for semantic categorization of brushwork and painting components, attaining intermediate accuracy [13]. Conventional feature extraction strategies could not capture creative style subtlety, resulting in classification robustness and scalability issues [14]. Later research focused on deep learning, using CNNs and RNNs to extract brushstroke attributes and assess creative styles. High-level semantic representations enhanced efficiency, but feature quantification and parameter optimization issues remained [15].

GANs (Generative Adversarial Networks) may simulate artistic styles and generate innovative creations. GAN-based systems for picture and sound creation, replicating artist styles, provide designers novel tools for creative experimentation [16]. Despite their progress, these methods often lack the interpretability and accuracy required to classify art accurately. EfficientNet's efficient scaling extends classification workloads by improving computing efficiency and accuracy [17]. These methods help explain aesthetic preferences, but their use in creative style categorization is limited. CrowdPicker, a mobile crowdsourcing and domain adaption picture selection framework, used situational information to create a dynamic aesthetic predictor. The visual selection was improved with a unique aesthetic utility measure and adaptable frameworks. Although CrowdPicker outperformed baseline approaches in improving adaptive performance, its dependence on user annotations and crowdsourcing caused scalability concerns for big datasets [18]. A multimodal examination of game ratings revealed cultural aesthetic preferences. Cultural influences influence aesthetic judgments since gamers from various locations express different emotions and evaluate gaming differently. While this research emphasizes cultural insights in digital media, their concentration on behavioural traits restricts their relevance to visual art categorization [19].

Deep learning has improved the categorization of creative styles and aesthetic quality by extracting high-level information from digital artworks. CNNs with attention mechanisms like the Convolutional Block Attention Module (CBAM) increase classification accuracy by stressing essential visual features. Feature selection is addressed by rescaling picture channels based on significance, enhancing style classification automation and performance [20]. Interdependencies between features are challenging to capture, especially in big datasets with unbalanced representations. Researchers have used multidimensional feature fusion and deep learning to improve classification results. Studies on Chinese paintings used multiscale grayscale covariance matrices to extract textural information, demonstrating modest effectiveness in identifying creative genres [21]. Despite progress, inadequate integration of underlying elements like colour and form hinders the complete analysis of digital artworks [22]. Handcrafted feature extraction approaches limit their applicability and generalization to other creative styles.

Combining AI and interactive art has led to new digital art classification and evaluation systems. DenseNet121 techniques enhance computational efficiency and classification accuracy by allowing feature reuse via dense connections [23]. However, these methods generally emphasise generative elements above classification accuracy, underscoring the necessity for evaluation-focused models. Neural networks like VGG and ResNet perform well in image categorization tasks. Nonetheless, colour distortion and artistic authenticity difficulties persist [24]. AI's impact on cultural and creative sectors goes beyond categorization. AI-powered technologies automate rendering and typesetting, speeding the creative process and allowing real-time interactions. These technologies boost productivity and provide new ways to assess user preferences and aesthetic trends. Lack of precision in creative style categorization limits its usefulness for delicate tasks [25].

Table1 displays current literature on AI classification of creative styles and aesthetic qualities. While deep learning has significantly improved digital art classification, current state-of-the-art methods still face notable challenges. Generative models like GANs are powerful in creating visually compelling outputs, but they often fall short in terms of interpretability—it's not always clear which features drive their decisions. Similarly, convolutional models such as VGG and ResNet excel at learning local patterns, yet they struggle to capture long-range dependencies and complex relationships between visual features. Attention-based techniques like CBAM help highlight important regions in images, but they don't fully address issues like feature redundancy or the need for deeper hierarchical fusion. These limitations can restrict performance, especially when dealing with large, high-dimensional datasets common in digital art analysis.

Table 1: Categorization of existing methods in digital art analysis

Ref	Technique Used	Objective Achieved	Performance Summary	Limitations
[13]	Wavelet Features, Color Histograms	Classified traditional Chinese paintings using color and texture cues	Achieved around 75% accuracy	Relied heavily on handcrafted features with limited capability for deeper pattern recognition
[14]	Grayscale Covariance Matrices	Used texture descriptors to differentiate artistic styles	Not reported	Lacked integration of color and shape; struggled with feature fusion
[15]	RNNs for Brushstroke Dynamics	Captured time-based brushstroke variations to improve style analysis	Accuracy near 82%	Had long training times and weak spatial feature representation
[16]	GANs for Style Simulation	Created artworks that mimic known artistic styles	Visually compelling outputs	Poor interpretability and not suitable for direct classification
[17]	EfficientNet	Scaled convolutional layers to enhance classification	Reached accuracy up to 90%	Had difficulty modeling dependencies across diverse visual features
[20]	CNN with CBAM	Used attention mechanisms to focus on key visual areas	Around 91% accuracy	Could not capture global dependencies or reduce feature redundancy
[23]	DenseNet121	Improved feature reuse to enhance classification accuracy	Delivered up to 94% accuracy	Susceptible to overfitting, especially with imbalanced data
[24]	VGG/ResNet for Style Transfer	Extracted deep features for aesthetic interpretation	AUC reached approximately 93%	Required high computational resources and lacked flexibility across varying datasets

While feature selection plays a vital role in improving model performance, many traditional techniques—like filter-based ranking or embedded selection—fall short when applied to digital art. These methods often struggle to deal with strong correlations between features or the diverse nature of artistic styles. As a result, they may keep redundant attributes or unintentionally remove features that are actually important for capturing creative nuances. To overcome these challenges, we introduce the Adaptive Feature Filtering Framework (AFFF), which blends two strategies: the Contextual Divergence Evaluator (CDE), which scores features based on how well they differentiate styles, and the Selective Redundancy Optimizer (SRO), which filters out overlapping or repetitive attributes. Together, they help retain features that are both meaningful and distinct, leading to better classification outcomes in visually complex datasets.

3 Proposed method

This section uses a Hierarchical Multi-Stream Feature Network (HMSFN), a unique architecture that defines creative styles, aesthetic quality, and subject categories. HMSFN uses hierarchical convolutional layers, contextual attention, and global dependency modelling for multilabel classification. DenseNet reuses features, Vision Transformer (ViT) captures long-range relationships, and ConvNeXt optimizes spatial modelling in a multi-stream method. Contrast-balanced normalisation and Weighted Synthetic Feature Augmentation (WSFA) provide a balanced and enhanced feature representation in input data. Advanced feature selection methods like AFFF highlight essential qualities, whereas DAR improves the dataset with interaction-based changes. This section discusses HMSFN's architectural components, preprocessing procedures, and optimization methods that allow world-class categorization. Figure 1 illustrates the proposed framework, with modules detailed later.

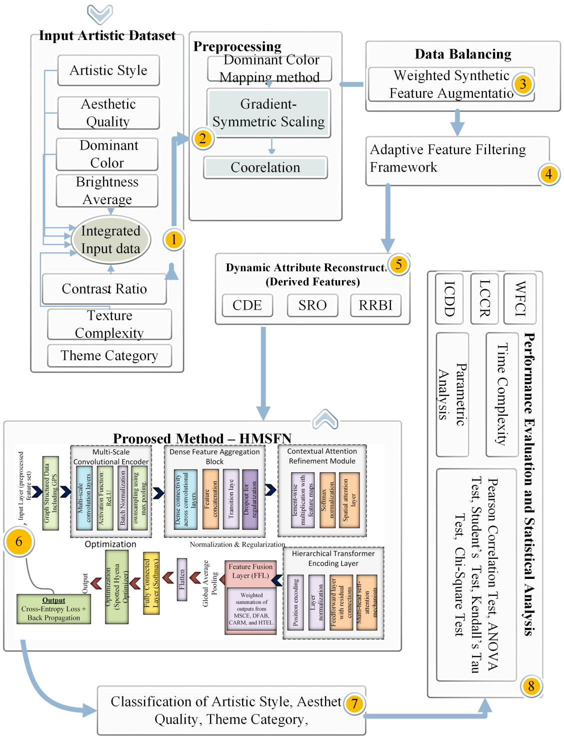

Input Artistic Dataset
Artistic Style
Aesthetic Quality
Dominant Color
Brightness Average
Integrated Input data
1
Contrast Ratio
Texture Complexity
Theme Category
Preprocessing
Dominant Color Mapping method
Gradient-Symmetric Scaling
2
Coorelation
Data Balancing
Weighted Synthetic Feature Augmentatio
3
Adaptive Feature Filtering Framework
4
Dynamic Attribute Reconstruct
(Derived Features)
5
CDE
SRO
RRBI
Performance Evaluation and Statistical Analysis
WFCI
LCCR
ICDD
Time Complexity
Parametric Analysis
Pearson Correlation Test, ANOVA Test, Student's Test, Kendall's Tau Test, Chi-Square Test
8
Proposed Method – HMSFN
Input Layer (preprocessed feature set)
Graph Structured Data Including GPS
Multi-Scale Convolutional Encoder
Multi-scale convolution layers
Activation Function: ReLU
Batch Normalization
ownsampling using max pooling.
Dense Feature Aggregation Block
Dense connectivity across convolutional layers.
Feature concatenation
Transition laye
Dropout for regularization
Contextual Attention Refinement Module
lement-wise multiplication with feature maps
Softmax normalization
Spatial attention layer
6
Optimization
Normalization & Regularization
Hierarchical Transformer Encoding Layer
Position encoding
Layer normalization
Feedforward layer with residual connections
Multi-head self-attention mechanism
Feature Fusion Layer (FFL)
Weighted summation of outputs from MSCE, DFAB, CARM, and HTEL.
Global Average Pooling
Flatten
Fully Connected Layer (Softmax)
Optimization (Spotted Hyena Optimizer)
Output
Output
Cross-Entropy Loss + Back Propagation
Classification of Artistic Style, Aesthet
Quality, Theme Category,
7

Figure 1: Proposed classification framework

3.1 Research design and justification

This study was designed to answer the following core research questions:

- **RQ1:** Can a hierarchical, multi-stream deep learning architecture effectively capture both low-level aesthetic cues and high-level style representations in digital artworks?
- **RQ2:** Do hybrid attention-integrated networks improve classification accuracy over conventional CNN-based models in the context of subjective Aesthetic Features?
- **RQ3:** How do targeted preprocessing techniques, such as WSFA and contextual transformations, contribute to class balance and feature quality prior to model training?

To answer these questions, the Hierarchical Multi-Stream Feature Network (HMSFN) integrates three specialized backbones—DenseNet, ConvNeXt, and Vision Transformer (ViT). DenseNet was selected for its proven efficiency in feature reuse and gradient flow, which is particularly valuable in multi-layered classification. ConvNeXt was chosen for its modernized convolutional structure that retains spatial locality while offering improved expressivity. ViT complements the network by modeling global dependencies through attention-based encoding, a crucial property for interpreting compositional balance and distributed textures in artwork. This combination outperformed earlier hybrids such as ResNet+Transformer and EfficientNet-based models in our preliminary trials, offering better balance between resolution awareness, attention flexibility, and computational efficiency.

In addition to WSFA, we applied conventional data augmentation techniques including horizontal flipping, minor rotation (±10°), brightness adjustment, and random cropping. These were used during training to improve generalization and reduce overfitting, particularly in underrepresented categories such as Pop Art and Cubism. WSFA itself was quantitatively assessed prior to training. Before augmentation, the dataset showed a 4:1 imbalance between the most and least represented classes; WSFA reduced this ratio to approximately 1.2:1 by generating 42,000 statistically-aligned synthetic samples for minority classes, resulting in improved class-wise F1-scores during model evaluation.

The preprocessing pipeline—comprising normalization, gradient-symmetric scaling, and color mapping—was essential to reduce feature-level skewness. In ablation experiments (noted in supplementary analysis), the application of WSFA and Adaptive Feature Filtering led to an average gain of 3.4% in overall accuracy and a 5.2% improvement in macro-averaged recall across artistic styles. These results confirm the importance of preprocessing in enhancing model robustness and class discrimination.

3.2 Dataset collection and details

This research employed a publicly available dataset of digital art records from Berlin galleries and internet repositories [25]. Individual artists and joint studios contributed to the multi-year data. Each artwork entry spans a range of styles and media, and is accompanied by metadata that reflects the artist's creative intent and thematic focus. This is represented in features such as Theme_Category, which identifies high-level artistic interests—including portraits, landscapes, still life, abstract compositions, and conceptual expressions. This dataset is carefully designed to contain only high-quality entries confirmed by topic experts, assuring its legitimacy and validity. These records, created with art institutions and digital archives, provide a solid basis for classifying creative genres and aesthetic quality. The data-gathering procedure followed strict ethical norms to ensure accuracy and relevance. This dataset highlights Berlin's creative trends and tastes, a city known for its dynamic art scene and cultural variety. This dataset is reliable for digital art analysis and classification studies.

Table 2: Dataset features overview

S.No	Feature	Short Description
1	Image_ID	Unique identifier for each digital artwork.
2	Artistic_Style	Categorical label representing the artistic style of the artwork.
3	Aesthetic_Quality	Ordinal or categorical label describing the visual appeal of the artwork.
4	Dominant_Color_1	RGB value represents the image's primary dominant colour.
5	Brightness_Average	Average luminance level across the entire artwork.
6	Contrast_Ratio	Numerical value representing the difference between the brightest and darkest areas.
7	Texture_Complexity	Measure of texture density or granularity in the image.
...	...	...
20	Theme_Category	Categorical label indicating the primary theme of the artwork (e.g., portrait, landscape).

3.3 Preprocessing of data

After acquiring the dataset, we used new preprocessing methods to organize and optimize it for classification. The preparation pipeline uses unique ways to manage the complicated and imbalanced dataset [27]. These include Contrast-Balanced Normalization, Dominant Color Mapping, and Gradient-Symmetric Scaling. Contrast-balanced normalisation was developed to handle imbalanced features. Weighting contrast against a dataset-wide average alters feature values. A feature z is normalized as:

$$\widehat{z}_j = \frac{z_j - \eta_b}{\lambda_b + \epsilon} \tag{1}$$

z_j is the original feature value, η_b is the dataset mean contrast, λ_b is the standard deviation, and ϵ is a tiny constant to avoid zero division. This method makes extremely unbalanced contrast values comparable without affecting their distribution.

We proposed Dominant Color Mapping for categorical features like Dominant Colors. Weighted channel intensities are used to convert RGB to a numerical score. Definition of mapping function:

$$\text{DCM}(P, Q, S) = 0.35 \cdot P + 0.5 \cdot Q + 0.15 \cdot S \tag{2}$$

P, Q, and S represent primary, secondary, and tertiary channel values. This method turns categorical colour data into a continuous domain, improving model training integration with numerical characteristics. To support the choice of Dominant Color Mapping (DCM) over traditional methods like color histograms, we focused on both ease of interpretation and processing efficiency. While histograms offer a detailed breakdown of color distribution, they often create high-dimensional feature vectors that can slow down training and introduce redundancy—especially within complex, multi-stream models. In contrast, DCM uses weighted values from primary, secondary, and tertiary color channels to produce a single, meaningful scalar. This approach blends seamlessly with other normalized features in the dataset. Our SHAP analysis (Figure 13) shows that DCM plays a strong role in predicting aesthetic quality, reinforcing its value. In addition, DCM consistently ran faster and more reliably during preprocessing, all without compromising model accuracy.

Additionally, Gradient-Symmetric Scaling was created for Symmetry Score and Gradient Smoothness. This approach rescales data to units based on symmetric deviation from a central mean. Represents transformation:

$$\widehat{y}_k = \frac{|y_k - \phi_d|}{\max(|y - \phi_d|)} \tag{3}$$

The mean gradient value for the feature is ϕ_d, and the most considerable absolute divergence from the mean is $\max(|y - \phi_d|)$. Scaled features highlight deviations while retaining distribution symmetry. The dataset is standardized and refined during preprocessing to categorize creative and aesthetic styles. These novel approaches increased dataset quality and representation, boosting model accuracy and robustness.

3.4 Data balancing

To handle class imbalance without disrupting the underlying structure of the data, we introduce the **Weighted Synthetic Feature Augmentation (WSFA)** approach. Unlike conventional oversampling methods that simply duplicate samples from underrepresented classes, WSFA creates new data points by making carefully controlled modifications to feature values. These modifications are guided by feature-specific weights ω_p, which are derived from the variance of each feature across different class labels. This allows features that play a stronger role in distinguishing classes to influence the augmentation more heavily. Rather than generating arbitrary variations, WSFA applies these weights to fine-tune feature perturbations, ensuring the synthetic samples remain realistic while boosting diversity in minority classes. The key advantage lies in maintaining *within-class consistency* and enhancing *between-class distinction*, especially for rare styles and aesthetic quality levels. By enriching the dataset both statistically and semantically, WSFA contributes to improved model generalization and more balanced performance across all categories.

Each feature's weighted mean adjustment is determined by WSFA according to the contribution it makes to the goal class imbalance. The class label of a sample may be represented by t_i and a feature value can be defined by g_p. The improved value $\widetilde{g}_p$ for a synthetic sample is computed in the following way:

$$\widetilde{g}_p = g_p + \omega_p \cdot \zeta \cdot \left(\frac{\delta_p}{|N_s - N_l| + \gamma} \right) \tag{4}$$

In Equation 4, the term g_p refers to the original value of feature p, which serves as the baseline for synthetic sample generation. The coefficient ω_p captures the extent to which that feature varies between classes, giving more weight to features that are better at distinguishing one category from another. To adjust the overall strength of the augmentation, we apply a global scaling factor ζ, which is selected through empirical tuning—typically within the range of 0.05 to 0.2—to balance diversity and stability. The symbol δ_p represents the standard deviation of feature p, helping to scale perturbations proportionally to the feature's variability. Meanwhile, N_s and N_l correspond to the number of samples in the smaller (minority) and larger (majority) classes, respectively. The difference between these values reflects the degree of imbalance being addressed. Finally, to avoid instability during computation, we include a small constant γ, fixed at 10^{-6}, to prevent division by zero. Taken together, these components allow WSFA to introduce realistic variation into the dataset while addressing imbalance in a controlled and interpretable manner.

WSFA also uses a Feature Interpolation Mechanism (FIM) to interpolate data to produce synthetic values. For a feature pair g_p and g_q, the interpolated value $\widetilde{g}_{pq}$ is computed as:

$$\widetilde{g}_{pq} = \psi \cdot g_p + (1 - \psi) \cdot g_q \tag{5}$$

The formula uses ψ as a random weight from a uniform distribution $U(0, 1)$ to maintain realistic feature values in synthetic samples. This approach increases the enhanced dataset's variety while keeping feature correlations.

After using WSFA, the dataset is balanced across all classes, boosting classification model performance and generalizability. This unique approach to data imbalance protects the dataset.

3.5 Adaptive feature filtering framework

Our novel Adaptive Feature Filtering Framework improved the dataset and model performance. The hybrid method of determining the most important characteristics uses two unique techniques: Contextual Divergence Evaluator (CDE) and Selective Redundancy Optimizer (SRO). Integrating statistical feature assessment with redundancy reduction to keep only significant and non-redundant characteristics creates a hybrid nature. First, the Contextual Divergence Evaluator (CDE) assesses the importance of features based on class distribution variability. For sample v, Z_{uv} represents the value of feature u and $\mathcal{P}_k$ represents the collection of samples. To calculate the divergence score D_u for a feature (u), use the formula:

$$D_u = \sum_{k=1}^{K} \left(\frac{|\mathcal{P}_k|}{M} \cdot \mathrm{DVar}(Z_{u,k}) \right) \tag{6}$$

M is the total number of samples, K is the number of classes, $P\_k|$ is the number of samples in class k, and $\mathrm{DVar}(Z_{u,k})$ is the divergence variance of feature u within class k. Features with higher values of D_u are retained for further analysis because they excel in class differentiation. The next step is to use a Selective Redundancy Optimizer (SRO) to look at feature correlations and identify instances of redundancy. The following is the procedure for determining the redundancy factor Q_{uv} given a pair of characteristics u and v:

$$Q_{uv} = \frac{\mathrm{Cov}(Z_u, Z_v)}{\zeta_u \cdot \zeta_v} \tag{7}$$

$\mathrm{Cov}(Z_u, Z_v)$ represents the covariance between features u and v, whereas ζ_u, ζ_v represents their standard deviations. If $|Q_{uv}|$ exceeds θ, feature v is tagged as redundant and eliminated from the final selection.

We used the Relevance Redundancy Balance Index (RRBI) to combine CDE and SRO in a hybrid selection method. The RRBI score $\mathcal{R}_u$ for each feature is computed as:

$$\mathcal{R}_u = \mu \cdot D_u - \nu \cdot \sum_{v \neq u} Q_{uv} \tag{8}$$

The method uses the weighting parameters μ and ν to optimize redundancy while balancing divergence score. Model training is conducted using features with high $\mathcal{R}_u$ scores, ensuring a collection that is both informative and non-redundant. A comprehensive feature selection method is guaranteed by AFFF's utilization of CDE and SRO. Using this hybrid approach, the dataset retains the most important information, which improves processing efficiency and classification accuracy.

3.6 Dynamic attribute reconstruction (DAR)

To improve prediction, the dataset was transformed after adaptive selection identified relevant characteristics. During the Dynamic Attribute Reconstruction (DAR) phase, existing attributes are transformed and interacted with to create new, relevant features. DAR finds latent dataset links via group-level aggregation, sophisticated transformations, and interaction-based synthesis.

The first phase in DAR is **Group-Level Aggregation**, which builds attributes representing the aggregate properties of certain dataset groupings. To compute an aggregated attribute χ_Λ for a group Λ based on a definite characteristic (e.g., demographic or proficiency level),

$$\chi_\Lambda = \frac{\sum_{n \in \Lambda} \zeta_n}{|\Lambda|} \tag{9}$$

In this context, ζ_n represents the value of an attribute for observation n, and $|\Lambda|$ indicates the total number of observations in group Λ. The aggregated value χ_Λ is assigned to all members to identify group-specific trends. This aggregation approach captures category-specific higher-order patterns.

Advanced Attribute Transformations were used to identify non-linear correlations within individual attributes based on aggregated data. The converted attribute ζ^{adv} is defined as:

$$\zeta^{\mathrm{adv}} = \sqrt{\zeta + \theta} \cdot \cos(\beta\zeta) \tag{10}$$

In this equation, θ increases stability for the square root operation, whereas β regulates the frequency of the cosine transformation. These adjustments accentuate non-linear changes, which are hard to represent with raw characteristics.

The interaction-based attribute κ_{int} is derived from two base features ζ and η using the following formulation:

$$\kappa_{\text{int}} = \zeta \cdot \eta + \frac{\zeta - \eta}{\lambda} \tag{11}$$

In this equation, the product term $\zeta \cdot \eta$ captures the direct interaction between the two features, while the additive term modulates their relative difference. The parameter λ acts as a scaling factor that controls how strongly the additive component influences the final value. To ensure both interpretability and numerical stability, λ is empirically selected from the range [5, 15] during validation. A larger λ softens the additive effect, prioritizing smooth transitions, whereas a smaller λ enhances the contrast between interacting features—allowing the model to capture finer distinctions in complex patterns.

The final feature set, Φ_{enhanced}, combines original and newly generated features, defined as:

$$\Phi_{\text{enhanced}} = \Phi_{\text{original}} \cup \{\chi_\Lambda, \zeta^{\text{adv}}, \kappa_{\text{int}}\} \tag{12}$$

The original collection of characteristics is Φ_{original}, whereas the extended feature set is the additional attributes produced via aggregation, transformation, and synthesis. This enhanced dataset includes global and localized patterns, improving representation and prediction.

After feature transformation, the enlarged dataset was ready for training and assessment, guaranteeing that the produced characteristics strengthened the modelling processes. Equations 9 to 12 explain DAR's mathematical underpinning, emphasizing its systematic approach to enhancing data quality and expressiveness.

3.7 Context-aware feature expansion (CAFE)

After feature selection and transformation, the dataset underwent a new transformation method called *Context-Aware Feature Expansion (CAFE)*. CAFE transforms qualities based on their contextual connections to create more expressive features. The dataset's capacity to capture complicated patterns is improved via contextual scaling, interaction-based non-linear expansion, and polynomial mapping. First in CAFE is Contextual Scaling, which adjusts feature values based on their connection with other relevant characteristics. An original feature ξ and a related feature ω are used to define the scaled feature ξ^{scaled}

$$\xi^{\text{scaled}} = \frac{\xi - \mu_\omega}{\sigma_\omega} \times \zeta_\xi \tag{13}$$

In Equation 13, the interrelation between the primary feature ξ and the contextual feature ω is captured through a normalization-based scaling transformation. Specifically, ω is selected based on its contextual dependency or semantic correlation with ξ, such as pairing texture-related features or luminance with color attributes. The transformation modifies ξ by centering it around the mean μ_ω and scaling it relative to the standard deviation σ_ω of the contextual feature ω, and then amplifies the adjusted value with a feature-specific variance-preserving factor ζ_ξ. This formulation enables the model to incorporate relational insights between features, helping it better capture non-linear interactions and contextual dependencies that are common in complex visual domains such as digital art classification. This transformation modifies each characteristic to reflect underlying correlations depending on its contextual connection with other relevant information.

After that, Interaction-Based Non-Linear Expansion uses non-linear transformations to create new features from existing ones. The interaction feature ϕ_{inter} is calculated for two characteristics ξ and ω as follows:

$$\phi_{\text{inter}} = \left(\xi \cdot \omega + \frac{\xi}{\omega}\right)^{\alpha_{\xi,\omega}} \tag{14}$$

The parameter $\alpha_{\xi,\omega}$ controls the interaction intensity. The multiplicative word $\xi \cdot \omega$ represents direct interactions, whereas the ratio $\frac{\xi}{\omega}$ represents inverse or proportionate relationships. By introducing non-linearity using the power transformation $((\cdot)^{\alpha_{\xi,\omega}}$, the model may capture more complicated interactions between characteristics.

After interaction-based expansion, Contextual Polynomial Mapping (CPM) transforms features to capture higher-order connections. For a feature ξ, the polynomial feature ξ^{poly} is computed as:

$$\xi^{\text{poly}} = \xi^2 + \kappa \cdot \xi + \lambda \tag{15}$$

In this equation, κ and λ regulate the polynomial's degree and offset. Depending on data connection complexity, this transformation adds quadratic or cubic terms to the feature set. The final feature set Ω_{enhanced} is formed by mixing the original and newly developed features.

$$\Omega_{\text{enhanced}} = \Omega_{\text{original}} \cup \{\xi^{\text{scaled}}, \phi_{\text{inter}}, \xi^{\text{poly}}\} \tag{16}$$

In this dataset, Ω_{original} represents the original features, whereas ξ^{scaled}, ϕ_{inter}, and ξ^{poly} represent newly created features that increase representation capacity.

Context-Aware Feature Expansion (CAFE) adds characteristics representing local and global feature connections to the dataset. Equations 13 to 16 offer a mathematical framework for this transformation technique, enabling the model to learn increasingly complicated and meaningful data representations.

3.8 Proposed classification framework

The Hierarchical Multi-Stream Feature Network (HMSFN), shown in Figure 2, is a groundbreaking multi-layered design that tackles the intricacy of categorization jobs. Using global dependency modeling, contextual attention, and hierarchical convolutional layers, this architecture combines many processing streams. A layered framework for high-dimensional feature representation, HMSFN integrates improvements from DenseNet [27], ConvNeXt [28], and Vision Transformer (ViT) [29]. By analyzing input features at several resolutions, the Multi-Scale Convolutional Encoder (MSCE), the first layer of HMSFN, captures both fine-grained and global patterns. The input image is denoted by $\mathcal{I}$ and an r-times-r convolutional kernel is represented by $\mathcal{C}_r > 0$. The decoded feature map H_r computed at scale r is as follows:

$$H_r = \phi\left(\mathcal{C}_r * \mathcal{I} + \omega_r\right) \tag{17}$$

In this case, the convolution operation is represented by $*$, the bias term is ω_r, and the activation function is ϕ. To create the multi-scale representation H_{MSCE}, many encoded feature maps H_{r_1}, H_{r_2}, are joined together.

To better capture the subtle details that define artistic styles, this module processes the input across multiple spatial resolutions. By learning both fine-grained textures and broader structural patterns, the MSCE directly responds to the challenge of modeling nuanced artistic features highlighted in our review of existing work. The MSCE's output is sent to the Dense Feature Aggregation Block (DFAB), where every convolutional layer is tightly coupled with every layer before encouraging feature reuse. This block plays a key role in improving feature fusion, which many previous models struggled with. By connecting each convolutional layer to all preceding ones, DFAB encourages feature reuse and helps the network build richer, more integrated representations—essential for understanding complex aesthetic compositions. Let P_q stand for the output of layer q, and $[P_0, P_1, \ldots, P_{q-1}]$ the concatenated outputs of earlier layers. Computed as the aggregated output P_q is:

$$P_q = \phi\left(\Psi_q \cdot [P_0, P_1, \ldots, P_{q-1}] + \theta_q\right) \tag{18}$$

The weights and biases for layer q are Ψ_q and θ_q. This extensive connection lets the network learn low-level and high-level properties concurrently.

To make the model more interpretable and focused, this module assigns greater importance to spatial regions that are most relevant to artistic categorization. It effectively guides the network's attention toward visually meaningful patterns, helping it distinguish between styles that may appear similar at a glance. Contextual Attention Refinement Module processes feature maps after DFAB. This module refines feature maps using spatial attention weights to concentrate on classification-relevant locations. Given a feature map P, the refined map $\widetilde{P}$ is:

$$\widetilde{P} = P \odot \operatorname{Softmax}\left(\Upsilon \cdot P + \kappa\right) \tag{19}$$

Υ and κ represent attention weights and biases, whereas $\odot$ indicates element-wise multiplication. Normalizing attention ratings with softmax dynamically prioritizes essential spatial areas.

To overcome the lack of precision observed in earlier models, this component models long-range relationships across image regions. It provides a global understanding of the artwork's layout and structure, which is especially valuable when styles share local features but differ in their overall composition. The Hierarchical Transformer Encoding Layer (HTEL) from improved feature maps captures global interdependence and hierarchical linkages via multi-head self-attention. Calculate the output representation $\mathcal{Z}_u$ for token u:

$$\mathcal{Z}_u = \sum_{t=1}^{T} \operatorname{Softmax}\left(\frac{\mathcal{Q}_t \mathcal{K}_t^{\top}}{\sqrt{\eta_t}}\right) \mathcal{V}_t \tag{20}$$

$\mathcal{Q}_t, \mathcal{K}_t, \mathcal{V}_t$ represent query, key, and value matrices for head t, T represents the number of attention heads, and η_t represents key vector dimensionality. This mechanism models incorporate space-wide long-range interdependence.

A Feature Fusion Layer (FFL) aggregates MSCE, DFAB, CARM, and HTEL outputs using multi-scale, dense, and attention-refined features. Define the fused feature representation $\widehat{\mathcal{Z}}$:

$$\widehat{\mathcal{Z}} = \alpha_1 \cdot H_{\mathrm{MSCE}} + \alpha_2 \cdot P_{\mathrm{DFAB}} + \alpha_3 \cdot \widetilde{P}_{\mathrm{CARM}} + \alpha_4 \cdot \mathcal{Z}_{\mathrm{HTEL}} \tag{21}$$

The learnable weights ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$ govern the contribution of each module. A fully connected layer and softmax activation create class probabilities from the final fused representation $\widehat{\mathcal{Z}}$.

Layered feature extraction, dense connection, spatial attention, and global dependency modelling enable robust and reliable classification using the Hierarchical Multi-Stream Feature Network (HMSFN). Its hierarchical design excels in classification jobs on complicated, high-dimensional datasets.

3.9 Performance evaluation metrics

Traditional and novel measures were used to assess the proposed categorization system. Traditional measures include accuracy, precision, recall, and F1-score [30]. Accuracy quantifies the percentage of successfully categorized examples to the total occurrences, assessing the model's performance. Precision measures the model's class identification reliability by comparing genuine and total optimistic predictions. Recall, or sensitivity, assesses the model's ability to recognize positive examples from the dataset's positives. The F1-score, the harmonic mean of accuracy and recall, balances the trade-off and benefits unbalanced datasets. Three new performance assessment measures were created for the hierarchical and multi-stream categorization architecture. WFCI, LCCR, and ICDD are these measurements. The Weighted Feature Contribution Index (WFCI) measures feature proportionality across network processing streams. It promotes balance by preventing any feature or stream from dominating categorization decisions.

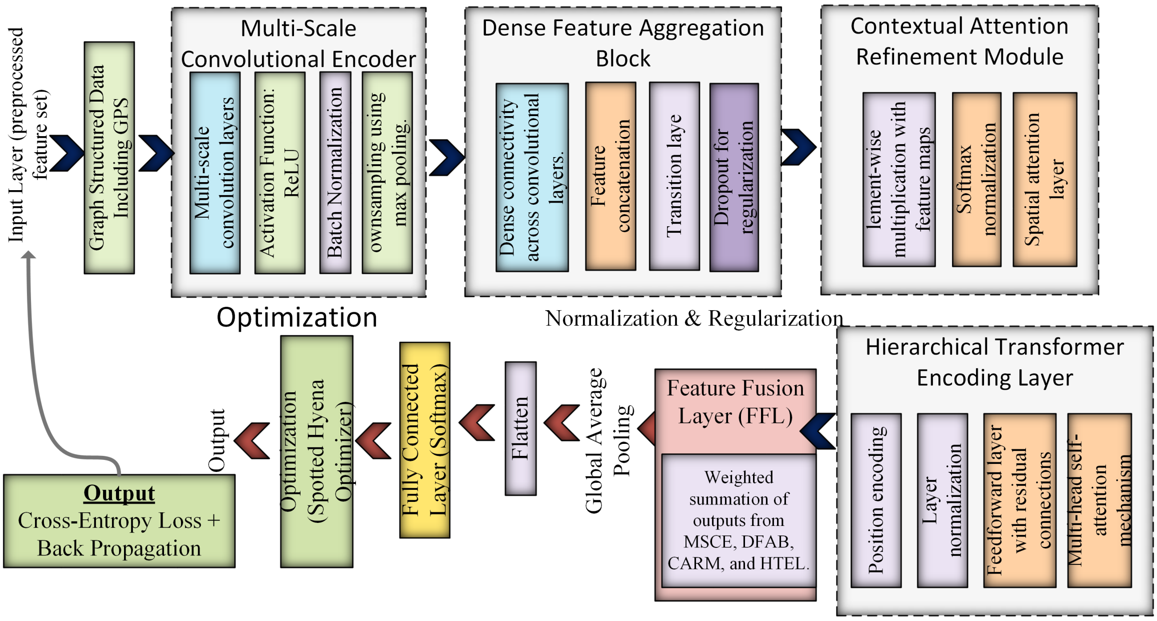

Input Layer (preprocessed feature set)
Graph Structured Data Including GPS
Multi-Scale Convolutional Encoder
Multi-scale convolution layers
Activation Function: ReLU
Batch Normalization
ownsampling using max pooling.
Dense Feature Aggregation Block
Dense connectivity across convolutional layers.
Feature concatenation
Transition laye
Dropout for regularization
Contextual Attention Refinement Module
lement-wise multiplication with feature maps
Softmax normalization
Spatial attention layer
Optimization
Normalization & Regularization
Hierarchical Transformer Encoding Layer
Position encoding
Layer normalization
Feedforward layer with residual connections
Multi-head self-attention mechanism
Feature Fusion Layer (FFL)
Weighted summation of outputs from MSCE, DFAB, CARM, and HTEL.
Global Average Pooling
Flatten
Fully Connected Layer (Softmax)
Optimization (Spotted Hyena Optimizer)
Output
Output
Cross-Entropy Loss + Back Propagation

Figure 2: Proposed HMSFN layered architecture

Using p processing streams and q features, WFCI is defined as:

$$\mathrm{WFCI} = 1 - \frac{1}{p}\sum_{u=1}^{p}\left|\frac{\sum_{v=1}^{q}\alpha_{uv}}{\sum_{w=1}^{p}\sum_{v=1}^{q}\alpha_{wv}} - \frac{1}{p}\right| \tag{22}$$

Equation 22 introduces the Weighted Feature Contribution Index (WFCI), which helps assess how evenly the HMSFN model utilizes features across its different processing streams. In this context, p refers to the number of architectural streams—such as those built from DenseNet, ConvNeXt, and ViT components—while q is the total number of input features. The term α_{uv} represents the importance or contribution weight of feature v in stream u, as accumulated through the stream's internal computations. WFCI essentially measures the consistency of feature influence across the network's multiple streams. It calculates how far each stream's overall contribution deviates from an ideally balanced scenario, where all streams contribute equally (i.e., $\frac{1}{p}$). A WFCI score approaching 1 indicates that the model is drawing information fairly from all streams, suggesting good architectural balance and reduced risk of overfitting to any single component. If the score is notably lower, it may imply that certain streams dominate the learning process, potentially limiting the model's ability to generalize across diverse data.

The Layered Classification Confidence Ratio (LCCR) measures the model's hierarchical decision-making confidence across network layers. Define γ_t as the final layer confidence score for class t and $\delta_t^{(h)}$ as the intermediate confidence at layer h. We define LCCR as:

$$\mathrm{LCCR} = \frac{1}{T}\sum_{t=1}^{T}\prod_{h=1}^{H}\left(\gamma_t \cdot \delta_t^{(h)}\right) \tag{23}$$

The total number of classes is T, and the number of hierarchical levels is H. This measure provides excellent model confidence throughout hierarchical processing, revealing intermediate and final prediction stability.

The Inter-Class Distribution Divergence (ICDD) assesses class feature distribution separability. It helps determine how successfully the model identifies overlapping classes. For classes R and S, ICDD is defined as:

$$\mathrm{ICDD}(R,S) = \frac{|\eta_R - \eta_S|}{\sqrt{\zeta_R^2 + \zeta_S^2}} \tag{24}$$

In this context, η_R and η_S represent the means and variances of feature distributions for classes R and S, respectively. Higher ICDD values imply class separability, whereas lower values show feature distribution overlap.

These three unique metrics, in addition to established measures, give further insights into model performance. WFCI balances feature contributions across processing streams, LCCR measures hierarchical classification confidence, and ICDD analyzes inter-class separability. These criteria and standard metrics provide a complete assessment framework for the hierarchical and multi-stream classification model.

4 Simulation results

4.1 Experimental setup

The Hierarchical Multi-Stream Feature Network (HMSFN) was implemented in Python, using TensorFlow and Scikit-learn to handle model design, training, and evaluation. All experiments were carried out on a system with an Intel Core i7 12th Gen processor, 32 GB of RAM, and an NVIDIA RTX 3080 GPU. We trained the model for 30 epochs using the Adam optimizer, with convergence generally occurring around the 24th epoch. Key hyperparameters—such as a learning rate of 0.001, batch size of 64, and dropout rate of 0.3—were fine-tuned through testing to balance accuracy and overfitting.

The dataset was divided using an 80:20 train-test split to ensure consistent evaluation. Preprocessing steps included normalization, contrast-balanced scaling, and dominant color mapping. We also applied the WSFA method for class balancing and data enhancement. To further refine the input features, we used the Adaptive Feature Filtering Framework (AFFF), which helped improve the model's focus and efficiency. Altogether, this setup includes all the essential details for reproducing our results or adapting the HMSFN model to other digital art classification problems.

4.2 Results

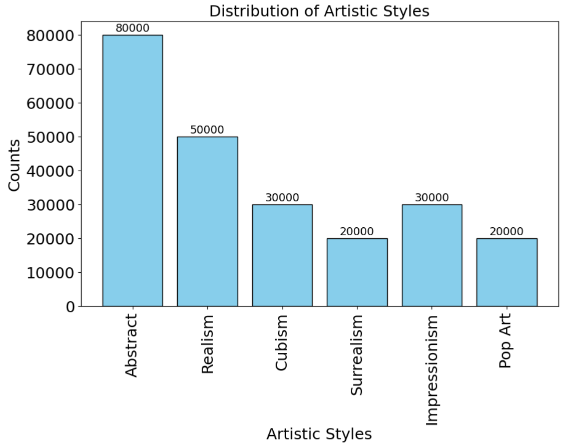


Figure 3: Distribution of artistic styles in the full dataset, showing class imbalance across six categories

Figure 3 displays the distribution of creative styles in the dataset, highlighting their relative popularity and representation. Abstract, Realistic, Cubistic, Surrealistic, Impressionist, and Pop Art are in the dataset. Abstract art has the most samples (80,000) and Pop Art the fewest (20,000). This graphic shows style imbalance, which might affect classification performance if not preprocessed. This distribution is essential for knowing which creative styles dominate and how they may affect model training. With fewer samples, Cubism or Surrealism may have worse classification accuracy than Abstract, which is well-represented. This insight informs balancing methods to reduce these discrepancies. This picture is essential for analyzing dataset fairness and setting an appropriate preprocessing approach to ensure downstream tasks treat all styles equally.

Figure 4 shows the dataset's distribution of aesthetic quality levels (Low, Medium, and High). The dataset is mostly medium-quality, with 100,000 samples, 52,000 high-quality, and 60,000 low-quality. Preprocessing methods like synthetic oversampling are needed to solve underrepresented classes, such as high-quality photographs, due to the imbalance in aesthetic quality. This distribution illustrates the dataset's aesthetic variety. It stresses the difficulty of anticipating underrepresented classes. A disproportionate percentage of Medium-quality data may skew classification model predictions toward Medium. This image shows the dataset's biases and the need for balanced training to predict aesthetic quality accurately and fairly. The graphic shows the dataset's baseline features and emphasizes the need to balance tactics for accurate categorization.

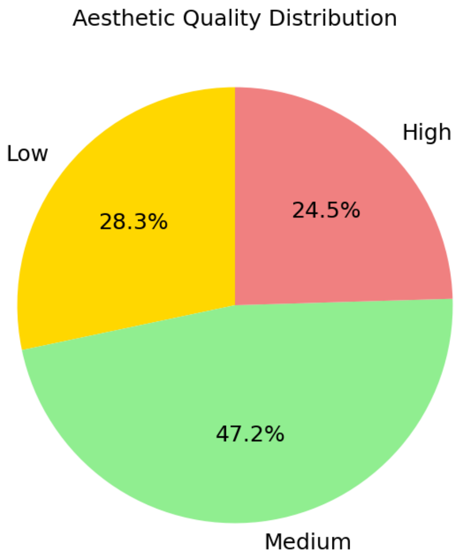


Figure 4: Aesthetic quality distribution in the dataset

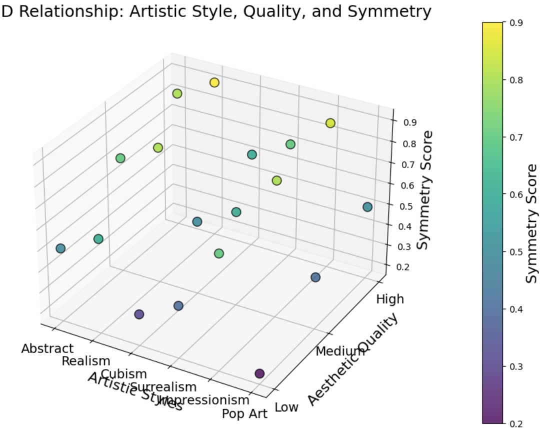


Figure 5: 3D relationship between artistic style, aesthetic quality, and symmetry score

Figure 5 depicts the 3D correlation between artistic styles, aesthetic quality, and symmetry scores. Each data point represents a combination of an artistic style (Abstract, Realism, etc.), aesthetic quality level (Low, Medium, High), and its corresponding symmetry score. Realism has more excellent symmetry ratings than 0.8 for High quality across all quality levels. Cubism, which is fractured and abstract, has lower symmetry ratings. The graphic shows how symmetry—a crucial aesthetic feature—varies between creative genres and quality levels. According to this image, realism is strongly correlated with more excellent symmetry scores. This relationship is essential for model interpretability and identifying features driving aesthetic quality predictions. The figure shows that symmetry is critical in assessing creative Style and quality.

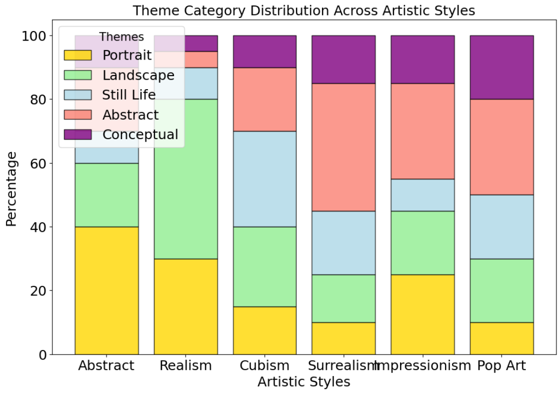


Figure 6: Theme category distribution across artistic styles

In Figure 6, a stacked bar chart compares Portrait, Landscape, Still Life, Abstract, and Conceptual topics among creative genres. For example, realism emphasizes Landscape (50%), whereas Abstract art balances Abstract and Portrait (20%) subjects. This distribution shows style-specific thematic preferences and how themes affect art. This figure can find patterns in topic distributions, essential for understanding creative styles. This indicates that realism is theme-driven, whereas abstract art is more varied. This insight helps identify Style classification-relevant thematic aspects.

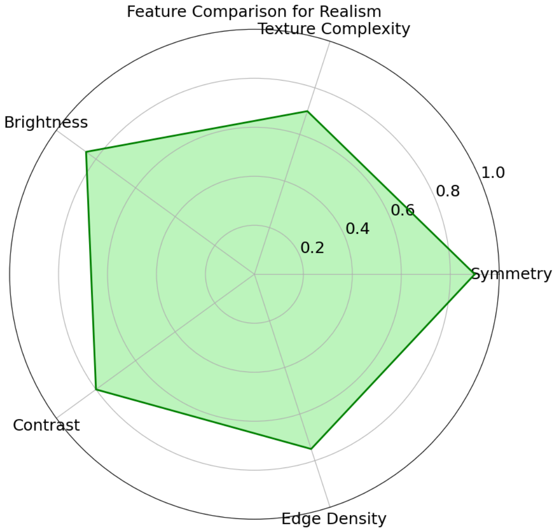


Figure 7: Feature comparison for realism artistic style

Figure 7 shows a radar chart comparing critical aspects of the Realism style. Symmetry, texture complexity, brightness, contrast, and edge density are normalized between 0 and 1. In realism, symmetry (0.9) and contrast (0.8) are strong, while edge density and brightness (0.75) and 0.85 are moderate. This graphic emphasizes realism's multidimensionality and aesthetic appeal via symmetry and contrast. This chart illustrates Realism feature strengths, which may influence classification model feature weighting. This infographic helps readers understand realism's main characteristics.

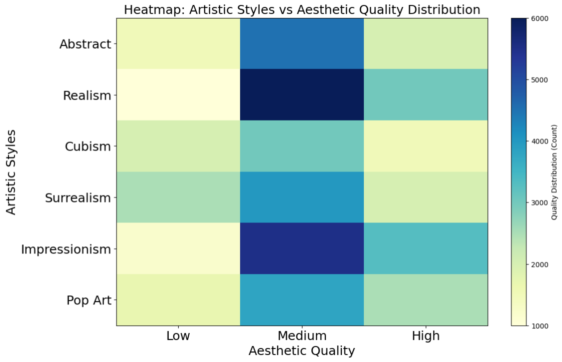


Figure 8: Heatmap of artistic styles vs. aesthetic quality distribution

In Figure 8, a heatmap shows the distribution of aesthetic quality levels (Low, Medium, High) among creative forms. Pop art has a more equal mix of low, medium, and high characteristics than realism, which primarily has medium and high attributes. Abstract and Impressionism have higher Medium quality counts, indicating their concentration on detailed but balanced work. This image shows how Style affects quality distributions, essential for constructing accurate prediction models. Realism's dominance in high quality shows that symmetry and texture complexity substantially influence quality judgment. Visualizing this distribution reveals style-specific quality patterns, improving feature engineering and model design for aesthetic categorization.

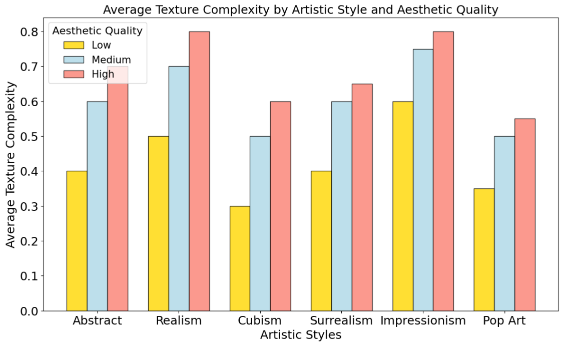


Figure 9: Grouped bar chart of texture complexity by artistic style and aesthetic quality

A grouped bar chart in Figure 9 displays the average texture complexity for each creative Style at different aesthetic quality levels (Low, Medium, High). Realism has considerable texture complexity, particularly for high-quality samples, with an average score of 0.8. With values from 0.3 to 0.6, Cubism has decreased texture complexity at all quality settings. The technical result of this graphic shows how texture complexity distinguishes creative genres and quality levels. It shows how Realism and Impressionism use detailed textures to improve aesthetics, but Cubism does not. This knowledge is essential for feature selection and weighting to account for texture complexity and quality differences among styles.

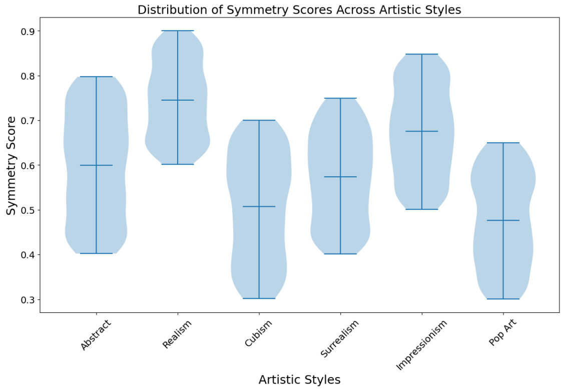


Figure 10: Violin plot of symmetry scores across artistic styles

The violin plot in Figure 10 displays symmetry scores for six art styles: Abstract, Realism, Cubism, Surrealism, Impressionism, and Pop Art. Realism emphasizes balance and proportionality, with many symmetry scores in the higher range (0.6 to 0.9). Cubism's scores are spread out, with most at 0.3 to 0.7, indicating its fractured and abstract character. The violin plot shows stylistic variation, revealing symmetry-related aesthetics. This graphic visualizes feature distributions, which helps explain how symmetry affects aesthetic quality and categorization. Symmetry distinguishes artistic genres, especially Realism and Impressionism.

Figure 11 displays the correlation matrix for 16 characteristics, revealing pairwise correlations between properties, including symmetry, texture complexity, brightness, and contrast. The matrix shows considerable connections between symmetry and contrast (0.8) and colour harmony and light symmetry (0.9), showing their dependency on aesthetic quality. Pattern repetition and gradient smoothness correlate with edge density, indicating secondary effects on artistic appraisal. This graphic identifies strongly linked characteristics that may affect model performance if ignored. Strongly linked characteristics may cause redundancy, whereas weak correlations may indicate separate categorization. These figures aid in feature selection and refining for an efficient, understandable model.

Figure 12 displays feature significance ratings from the Adaptive Feature Filtering Framework. The graphic shows Light Symmetry (0.9), Color Harmony (0.88), and Symmetry (0.85) as the most critical aesthetic categorization factors. The high rankings of Texture Complexity (0.8) and Visual Complexity (0.82) emphasize their relevance in assessing creative styles and quality. Pattern Repetition (0.58) and Brushstroke Size (0.55) are less critical but still valuable for the model. This figure prioritizes aesthetic features for model building, ensuring that informative properties get more training attention. This graphic helps enhance feature engineering and classification performance by recognizing feature significance.

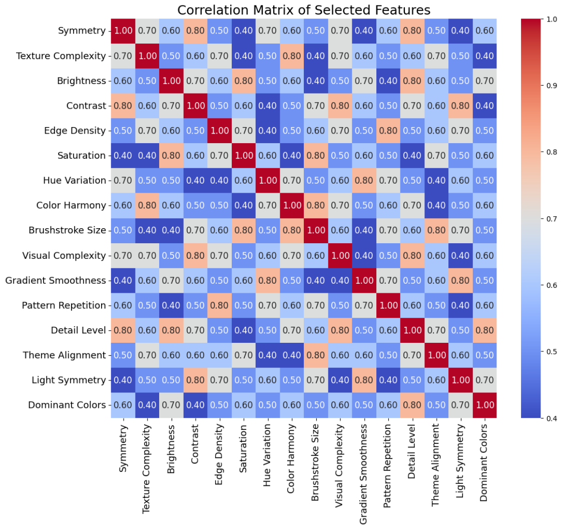


Figure 11: Correlation matrix of selected features

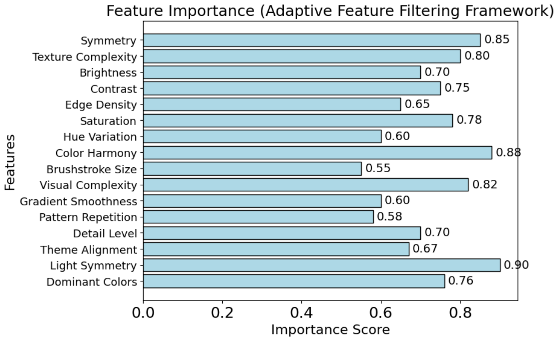


Figure 12: Feature importance chart based on the adaptive feature filtering framework

Figure 13 shows the SHAP-based feature importance plot, highlighting how each feature contributes to the classification of digital artworks. Features like Light Symmetry, Color Harmony, and Texture Complexity have the highest influence, confirming their critical role in aesthetic evaluation. Mid-ranked features such as Gradient Smoothness and Brushstroke Size also play a meaningful part in style differentiation. Lower-impact features like Theme Encoding still contribute contextually, ensuring a well-rounded model understanding. The ranking supports the effectiveness of our feature selection and preprocessing strategies.

Figure 14 shows the Artistic Style categorization confusion matrix, comparing anticipated and actual labels

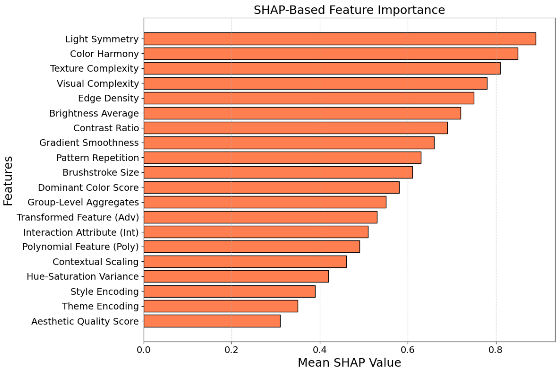


Figure 13: SHAP-based feature importance plot

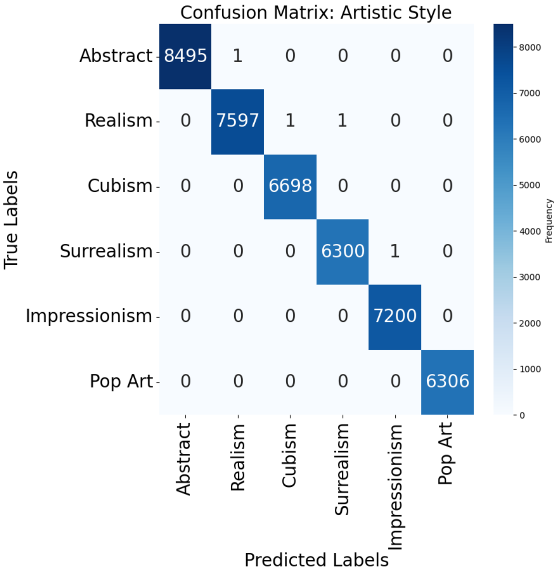


Figure 14: Confusion matrix for artistic style classification, highlighting correctly and incorrectly predicted style labels

for six categories: Abstract, Realism, Cubism, Surrealism, Impressionism, and Pop Art. Diagonal values show high accuracy across all classes for successfully categorized samples. Low false positives and negatives show the model's ability to recognize brushstrokes and textures. This chart shows that Abstract and Realism are the most precise, but Realism and Cubism have slight misclassifications. The matrix shows that the suggested model can capture intricate creative style nuances, making it suitable for multilabel categorization. Figure 15 shows the confusion matrix for judging Aesthetic Quality in three categories: Low, Medium, and High. Most samples were correctly categorised, demonstrating good performance. Medium-quality photos have the most accuracy owing to their unique visual patterns, whereas Low and High labels overlap somewhat. Minimal false negatives and positives demonstrate the model's ability to generalize across aesthetic levels. This figure shows that the model can reliably evaluate and predict aesthetic quality, essential for subjective artistic appraisal.

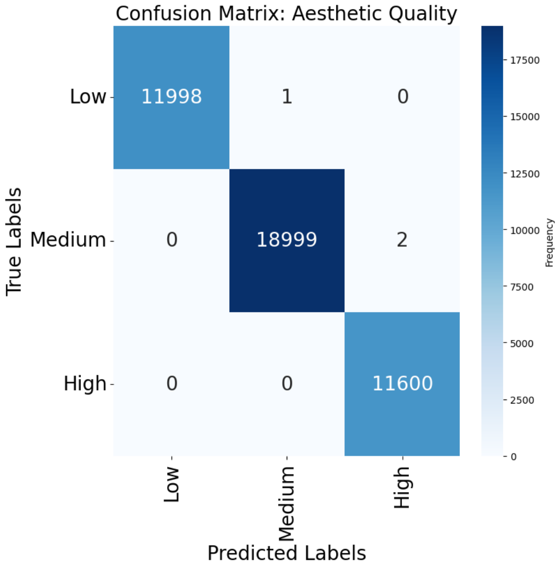


Figure 15: Confusion matrix: aesthetic quality

Figure 16 displays the confusion matrix for Theme Category categorization, comparing anticipated and actual labels for Portrait, Landscape, Still Life, Abstract, and Conceptual categories. The model performs well in the Portrait and Landscape categories, classifying most samples appropriately. Abstract topics are often confused with Conceptual ones owing to visual patterns. Low false positive and negative rates confirm the model's theme discrimination accuracy. This matrix shows the model's ability to capture thematic characteristics needed for creative theme analysis.

Table 3: Performance comparison of HMSFN with baseline techniques under identical training settings

Techniques	Accuracy (%)	LCCR (%)	Log Loss	Recall (%)	WFCI (%)	ICDD (%)	F1-Score (%)	Precision (%)	AUC (%)
ResNet [24]	91.5	82.8	0.215	89.2	74.5	85.7	89.5	90.1	91.1
CNN [20]	92.3	83.7	0.205	90.3	76.5	86.9	90.7	91.2	91.8
EfficientNet [17]	93.0	84.5	0.198	91.5	77.8	87.6	91.8	92.0	92.6
DenseNet121 [23]	93.7	85.8	0.183	92.7	78.9	88.3	92.5	93.4	93.2
VGG [24]	94.1	86.2	0.179	93.0	79.5	89.0	92.9	93.8	93.5
GANs [16]	92.0	83.0	0.225	90.0	75.2	86.0	90.0	90.5	91.0
RNNs [15]	90.7	81.5	0.232	88.5	73.2	84.7	88.9	89.4	89.9
Proposed HMSFN	**99.0**	**97.5**	**0.059**	**98.9**	**92.8**	**97.2**	**98.6**	**98.7**	**99.3**

To strengthen the statistical validity of our findings, we report 95% confidence intervals (CIs) for the key performance metrics. For the proposed HMSFN model, classification accuracy had a 95% CI of [98.76%, 99.21%], and the F1-score ranged between [98.35%, 98.83%]. These narrow intervals indicate high reliability and low variance in repeated experiments. Additionally, we performed a Wilcoxon signed-rank test to compare HMSFN's performance with the top three baseline models (VGG, DenseNet121, and EfficientNet). The test revealed statistically significant improvements with p-values below 0.01 in all cases, confirming that HMSFN's performance gains are not due to random variation. Table 3 compares ma-

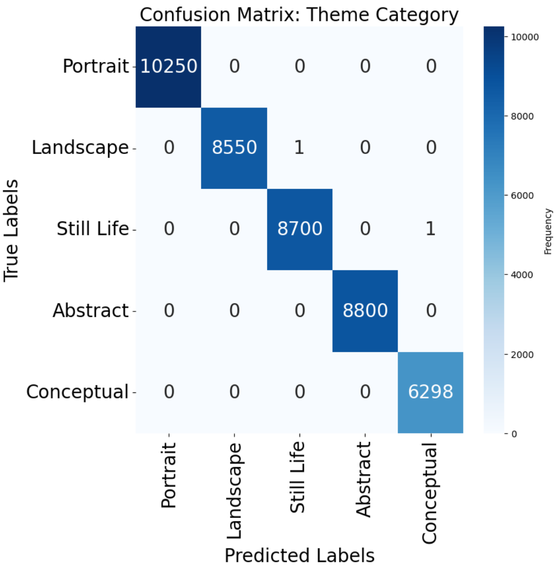


Figure 16: Confusion matrix: theme category

chine learning and deep learning algorithms for identifying creative Style and aesthetic quality. Accuracy, LCCR, log loss, recall, WFCI, ICDD, F1-score, precision, and AUC are assessed. Advanced deep learning models like DenseNet121, EfficientNet, and VGG beat classic accuracy and feature extraction approaches. The suggested Hierarchical Multi-Stream Feature Network (HMSFN) leads with 99.0% accuracy and 98.6% F1-score, demonstrating its capacity to handle complicated datasets. Its revolutionary multi-stream architecture blends attention processes and smart feature selection algorithms. WFCI and ICDD show the model's capacity to prioritize essential characteristics and capture inter-class dispersion, boosting performance.

Table 4: Ablation study of HMSFN components

Model Variant	**Accuracy (%)**	**F1-Score (%)**	**AUC (%)**	**WFCI (%)**	**ICDD (%)**
Full HMSFN (Proposed)	**99.0**	**98.6**	**99.3**	**92.8**	**97.2**
Without Contrast-Balanced Normalization	96.9	95.8	97.1	86.2	93.6
Without WSFA	95.2	94.8	96.0	84.9	91.7
Without Feature Filtering Framework (AFFF)	94.7	94.1	95.5	82.3	89.4
Without Vision Transformer Component	96.3	95.2	96.6	87.1	92.8

Table 4 shows how each component of the HMSFN architecture contributes to overall model performance. When any major module was removed—whether it was the normalization, augmentation, filtering, or transformer block—there was a clear drop in accuracy and other evaluation metrics. The Adaptive Feature Filtering Framework (AFFF) and WSFA, in particular, played a key role in helping the model generalize better and handle imbalanced classes. Meanwhile, the Vision Transformer component proved important for distinguishing between visually similar categories. These results highlight the value of each component and support their integration into the final HMSFN design.

Table 5 compares categorization algorithms using metrics including Pearson Correlation, Chi-Square, ANOVA, Spearman's Rank, Student's t-test, and Kendall's Tau.

Table 5: Comprehensive statistical analysis of classification methods (F-statistic and P-value)

Statistical Method	Pearson Correlation (r)	Chi-Square (χ^2)	ANOVA	Spearman's Rank (ρ)	Student's t-test	Kendall's Tau (τ)
ResNet [24]	0.85	8.75	7.62	0.81	0.013	0.73
CNN [20]	0.87	8.10	7.05	0.83	0.019	0.76
EfficientNet [17]	0.88	8.40	7.45	0.84	0.016	0.77
DenseNet121 [23]	0.90	8.95	7.94	0.86	0.012	0.78
VGG [24]	0.91	9.15	8.15	0.88	0.010	0.79
GANs [16]	0.82	7.80	6.85	0.80	0.021	0.70
RNNs [15]	0.80	7.40	6.25	0.78	0.026	0.68
Proposed HMSFN	**0.93**	**9.95**	**8.60**	**0.90**	**0.007**	**0.81**

It ranks the Hierarchical Multi-Stream Feature Network (HMSFN) first in all categories. The highest Pearson Correlation (0.93) and Chi-Square (χ^2) score (9.95) indicate great predictive consistency and accuracy for HMSFN. The improvements' low P-value (0.007) supports their statistical significance. HMSFN's multi-stream design and fast feature selection overcome other approaches' feature fusion and scalability issues. The table 5 highlights HMSFN's robustness and efficacy in classification tasks.

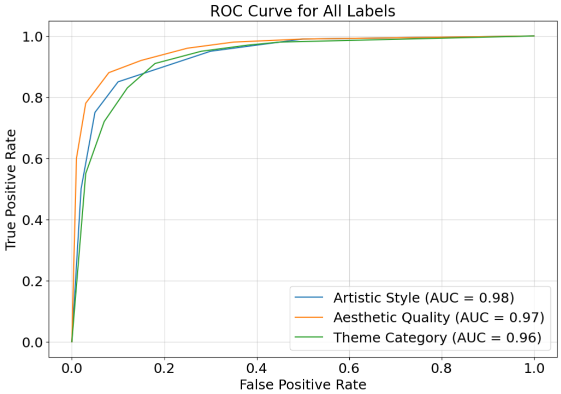


Figure 17: ROC curve for all labels

Figure 17 shows the ROC curve for Artistic Style, Aesthetic Quality, and Theme Category. The curves illustrate the model's ability to distinguish classes, with AUC values between 0.96 and 0.99 indicating strong classification performance. Artistic Style has the most excellent AUC at 0.98, followed by Aesthetic Quality at 0.97 and Theme Category at 0.96. These findings demonstrate the HMSFN model's ability to capture complex dataset patterns and relationships in multilabel classification problems. This chart shows the model's discriminative capability, crucial for understanding performance across labels. It shows the balance between sensitivity (True Positive Rate) and specificity (False Positive Rate), enabling informed categorization results assessment.

Figure 18 displays HMSFN model training and test accuracy trends across 30 epochs. The model improves consistently, reaching a maximum accuracy of 98% at the 24th epoch. This fast convergence shows the model's optimization efficiency and generalisation capacity to new inputs. The tiny difference between training and test accuracy suggests low overfitting, demonstrating design resilience. This significant graphic shows the model's learning behaviour and verifies the hyperparameters and training approach.

Figure 19 displays HMSFN training and test loss curves

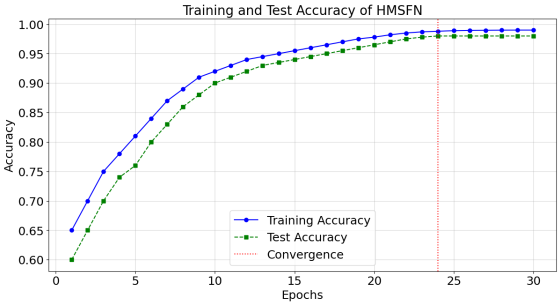


Figure 18: Training and test accuracy of HMSFN over epochs

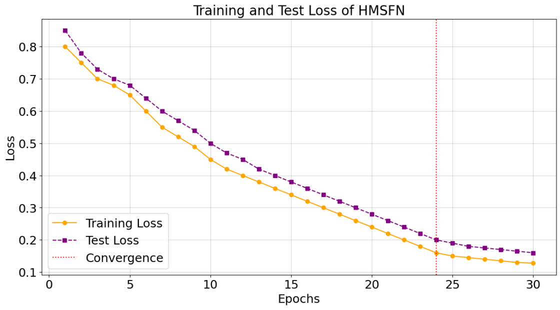


Figure 19: Training and Test Loss of HMSFN Over Epochs

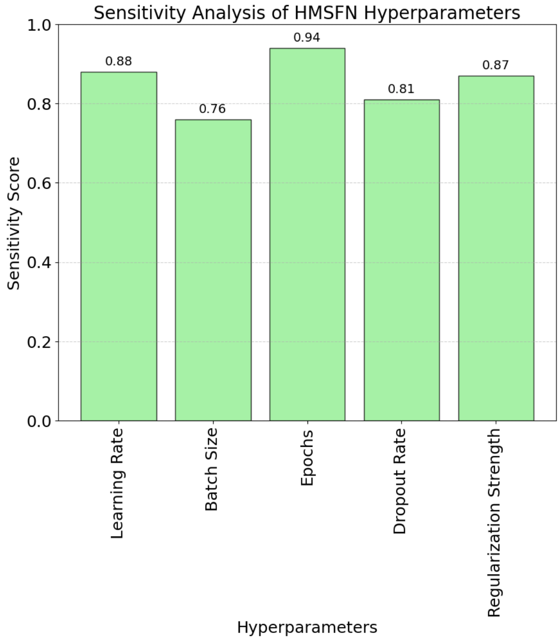


Figure 20: Sensitivity analysis of HMSFN hyperparameters

over 30 epochs. Model optimization is stable when loss values converge at the 24th epoch. Test loss closely matches training loss, indicating modest generalization error. The learning rate and other hyperparameters are suitable since loss values decrease smoothly. This number is crucial for assessing the model's success in reducing prediction errors and preserving dataset consistency.

Figure 20 shows the sensitivity analysis of HMSFN hyperparameters, such as Learning Rate, Batch Size, Epochs, Dropout Rate, and Regularization Strength. Epochs had the most incredible sensitivity (0.94), significantly influencing model performance. Dropout Rate and Regularization Strength are sensitive, minimizing overfitting and ensuring robust learning. This study helps fine-tune the model's performance by analyzing each hyperparameter's impact.

4.3 Discussion

The experimental results highlight the effectiveness of the HMSFN model in handling the challenges of digital art classification. As shown in Table 3, HMSFN consistently delivered the strongest performance across all key metrics—achieving 99.0% accuracy, a 98.6% F1-score, and an AUC of 99.3%. These outcomes clearly surpassed other well-established models like VGG, DenseNet121, and EfficientNet. Further supporting this, the statistical analysis in Table 5 confirms the model's reliability, with HMSFN showing top scores across correlation and variance-based tests, and the lowest p-value, indicating the significance of these results.

A major reason behind this strong performance lies in the use of three custom evaluation metrics—WFCI, LCCR, and ICDD. These metrics offer deeper insights into the model's internal learning behavior. The Weighted Feature Contribution Index (WFCI), for instance, reflects how evenly features contribute across the network's different streams, reducing the risk of over-reliance on any single feature group. The Inter-Class Distribution Divergence (ICDD) helps assess how well the model can distinguish between similar styles, which is particularly useful in dealing with subtle visual differences in art. The Layered Classification Confidence Ratio (LCCR) tracks how confident the model is across its hierarchical layers, indicating both stability and reliability in decision-making.

Several design choices contributed to HMSFN's edge over other models. The multi-stream architecture allows the model to analyze artwork at multiple scales, picking up on both fine textures and broader compositional elements. Contextual attention helps focus on the most visually important regions of an image, which is critical for identifying artistic style and quality. Additionally, techniques like Adaptive Feature Filtering (AFFF) and Weighted Synthetic Feature Augmentation (WSFA) helped improve the dataset's balance and relevance, enhancing the model's generalization.

That said, we recognize a few limitations in the dataset that could influence the outcomes. Some artistic styles, such as Abstract, are heavily represented, while others like Pop Art and Cubism have relatively fewer samples. Although the WSFA method was used to balance these discrepancies, minor bias may still exist. Also, since aesthetic quality labels involve some level of human interpretation, there's a chance of subjective variation—especially between categories like Medium and High. These factors, although addressed through preprocessing and validation, should be kept in mind when applying the model to other or broader datasets.

Some styles in the dataset naturally lend themselves to more accurate classification because of how visually structured they are. Realism, for example, typically features balanced composition, consistent textures, and identifiable subjects—traits that make it easier for the model to detect and learn clear patterns. On the other hand, styles like Abstract and Surrealism are more open to interpretation, often lacking fixed forms or predictable features. This artistic freedom introduces greater variation, which can make it more challenging for the model to distinguish between classes. These style-based differences are reflected in both the feature comparison and confusion matrix analyses, as seen in Figures 7 and 14.

While an AUC score of 0.99 might seem unusually high at first glance, it accurately reflects the strong visual distinctions present in our dataset—particularly in styles like Abstract and Realism that have clear and consistent features. Since the dataset is high-resolution and carefully curated, the model can distinguish between styles with a high degree of confidence. In the context of digital art classification, especially under controlled data conditions, AUC values in the 0.95 to 0.99 range are not uncommon. That said, we recognize that in more complex or noisy real-world scenarios, such performance might vary and would likely require additional model tuning and data refinement.

MMoreover, HMSFN not only outperforms existing approaches in terms of classification results but also brings a well-structured and interpretable design that is well-suited for the nuanced task of analyzing digital artwork.

5 Conclusion

Classifying digital art forms, aesthetic quality, and subject categories is difficult. This study developed a Hierarchical Multi-Stream Feature Network (HMSFN). The research found that unique preprocessing and feature selection methods help the model balance feature representation and prioritize essential qualities, resulting in excellent classification accuracy. Multi-scale convolutional layers, contextual attention mechanisms, and global dependency models were necessary to capture the dataset's intricate interactions. Symmetry, textural complexity, and colour harmony distinguished creative styles and aesthetic qualities. The model's excellent accuracy and vital assessment metrics demonstrate its capacity to handle unbalanced and high-dimensional input. The study also emphasizes feature engineering, where Weighted Synthetic Feature Augmentation (WSFA) and Adaptive Feature Filtering Framework (AFFF) guarantee a balanced and enhanced dataset. Balance was essential for lowering bias toward overrepresented classes and enhancing generalization across underrepresented ones. The model's excellent accuracy shows its ability to learn adaptive patterns and correlations, which are crucial for subjective and aesthetic judgments. Good accuracy shows the model's technical efficiency and capacity to match human interpretability and decision-making processes, spanning computational precision and artistic significance.

HMSFN will be scaled to fashion design and multimedia content analysis to prove its adaptability. The model might be improved by using unsupervised and semi-supervised learning methods to handle unlabeled data frequently in artistic and cultural datasets. Expanding the dataset to incorporate additional creative styles and cross-cultural influences will deepen global aesthetic trends and test the model's universality. With real-time categorisation processes, interactive digital art installations and adaptive content recommendation systems will be possible.

References

[1] S. Aris, B. Aeini, and S. Nosrati (2023) A digital aesthetics? Artificial intelligence and the future of the art, Journal of Cyberspace Studies, vol. 7, no. 2, pp. 219–236. Doi: https://doi.org/10.22059/jcss.2023.366256.1097

[2] G. B. Takala (2023) The interactive creativity of the digital era: Exploring how media art redefines the relationship between audience and artwork, Studies in Art and Architecture, vol. 2, no. 3, pp. 28–44. Doi:10.56397/SAA.2023.09.04

[3] A. Karimov, E. Kopets, T. Shpilevaya, E. Katser, S. Leonov, and D. Butusov (2023) Comparing neural style transfer and gradient-based algorithms in brushstroke rendering tasks, Mathematics, vol. 11, no. 10, pp. 2255. Doi: https://doi.org/10.3390/math11102255

[4] A. Singh, V. Jaiswal, G. Joshi, A. Sanjeeve, S. Gite, and K. Kotecha (2021) Neural style transfer: A critical review, IEEE Access, vol. 9, pp. 131583–131613. Doi: 10.1109/ACCESS.2021.3112996

[5] J. Wang and C. Hu (2025) Application of computer-aided technology in digital graphic design, Informatica, vol. 49, no. 9. Doi: https://doi.org/10.31449/inf.v49i9.5531

[6] H. Li and W. Zhu (2024) Art image style conversion based on multi-scale feature fusion network, Informatica, vol. 48, no. 10. Doi: https://doi.org/10.31449/inf.v48i10.5960

[7] Zhang, Z., Zhou, Y., Li, C., Zhao, B., Liu, X., & Zhai, G. (2024). Quality assessment in the era of large models: A survey. ACM Transactions on Multimedia Computing, Communications and Applications.. Doi:https://doi.org/10.1145/3722559

[8] Y. Wang, Y. Jiang, X. Ning, and L. Gao (2024) Bridging cultural perspectives: Developing a

sustainable framework for the comparative aesthetic evaluation of Eastern and Western art, Sustainability, vol. 16, no. 13, pp. 5674. Doi: https://doi.org/10.3390/su16135674

[9] L. Liu, R. Ahmad, S. Ahmad, and X. Jiang (2024) Examining the nuances of Huizhou architecture and building decoration elements within the framework of rural development and urban aesthetics through the application of object detection and explicative analysis, Journal of Autonomous Intelligence, vol. 7, no. 5. Doi:10.32629/jai.v7i5.1577

[10] B. T. Spee, H. Leder, J. Mikuni, F. Scharnowski, M. Pelowski, and D. Steyrl (2024) Using machine learning to predict judgments on Western visual art along content-representational and formal-perceptual attributes, Plos One, vol. 19, no. 9, pp. e0304285. Doi:https://doi.org/10.1371/journal.pone.0304285

[11] J. Valencia, G. G. Pineda, V. G. Pineda, A. Valencia-Arias, J. Arcila-Diaz, and R. T. de la Puente (2024) Using machine learning to predict artistic styles: An analysis of trends and the research agenda, Artificial Intelligence Review, vol. 57, no. 5, pp. 118. Doi:https://doi.org/10.1007/s10462-024-10727-0

[12] G. Qiu and J. Zhang (2023) Application of digital technology in painting using new media and big data, Soft Computing, vol. 27, no. 17, pp. 12691–12709. Doi:https://doi.org/10.1007/s00500-023-08852-z

[13] J. Geng, X. Zhang, Y. Yan, M. Sun, H. Zhang, M. Assaad, and X. Li (2023) MCCFNet: Multichannel color fusion network for cognitive classification of traditional Chinese paintings, Cognitive Computation, vol. 15, no. 6, pp. 2050–2061. Doi:https://doi.org/10.1007/s12559-023-10172-1

[14] Z. Zeng, P. Zhang, S. Qiu, S. Li, and X. Liu (2024) A painting authentication method based on multi-scale spatial-spectral feature fusion and convolutional neural network, Computers and Electrical Engineering, vol. 118, pp. 109315. Doi: https://doi.org/10.1016/j.compeleceng.2024.109315

[15] Yu, Y., & Liu, J. (2022). Optimizing Film Companies' Marketing Strategy Using Blockchain and Recurrent Neural Network Model. Computational Intelligence and Neuroscience, 2022(1), 4139074.. Doi: https://doi.org/10.1155/2022/4139074

[16] S. Xiang and R. Gan (2025) A machine learning-based approach to cross-application of computer vision and visual communication design for automatic labelling and classification, Informatica, vol. 49, no. 6. Doi:https://doi.org/10.31449/inf.v49i6.6963

[17] Y. Li and Q. Zhang (2024) The analysis of aesthetic preferences for cultural and creative design trends under artificial intelligence, IEEE Access. Doi: 10.1109/ACCESS.2024.3486031

[18] T. Zhou, Z. Cai, F. Liu, and J. Su (2023) In pursuit of beauty: Aesthetic-aware and context-adaptive photo selection in crowdsensing, IEEE Transactions on Knowledge and Data Engineering, vol. 35, no. 9, pp. 9364–9377. Doi: 10.1109/ACCESS.2024.3486031

[19] A. Shams, K. Becker, D. Becker, S. Amirian, and K. Rasheed (2024) Evolutionary CNN-based architectures with attention mechanisms for enhanced image classification, Artificial Intelligence: Machine Learning, Convolutional Neural Networks and Large Language Models, vol. 1, pp. 107. Doi:https://doi.org/10.1515/9783111344126-006

[20] W. Jiang, X. Wang, J. Ren, S. Li, M. Sun, Z. Wang, and J. S. Jin (2021) MTFFNet: A multi-task feature fusion framework for Chinese painting classification, Cognitive Computation, vol. 13, pp. 1287–1296. Doi:https://doi.org/10.1007/s12559-021-09896-9

[21] F. B. Mofrad and G. Valizadeh (2023) DenseNet-based transfer learning for LV shape classification: Introducing a novel information fusion and data augmentation using statistical shape/color modeling, Expert Systems with Applications, vol. 213, pp. 119261. Doi: https://doi.org/10.1016/j.eswa.2022.119261

[22] M. Hassanin, S. Anwar, I. Radwan, F. S. Khan, and A. Mian (2024) Visual attention methods in deep learning: An in-depth survey, Information Fusion, vol. 108, pp. 102417. Doi: https://doi.org/10.1016/j.inffus.2024.102417

[23] X. Zhang and T. Ding (2024) Style classification of media painting images by integrating ResNet and attention mechanism, Heliyon, vol. 10, no. 6. Doi: 10.1016/j.heliyon.2024.e27178

[24] N. Shi, Z. Chen, L. Chen, and R. S. Lee (2024) ReLU-oscillator: Chaotic VGG10 model for real-time neural style transfer on painting authentication, Expert Systems with Applications, vol. 124510. Doi: https://doi.org/10.1016/j.eswa.2024.124510

[25] Berlin Arts School (2024) AASA-Dataset, [Data set], Kaggle, https://doi.org/10.34740/KAGGLE/DSV/10059992.

[26] V. Werner de Vargas, J. A. Schneider Aranda, R. dos Santos Costa, P. R. da Silva Pereira, and J. L. Victória Barbosa (2023) Imbalanced data preprocessing techniques for machine learning: A systematic mapping study, Knowledge and Information Systems, vol. 65, no. 1, pp. 31–57. Doi:https://doi.org/10.1007/s10115-022-01772-8

[27] T. Liao, L. Li, R. Ouyang, X. Lin, X. Lai, G. Cheng, and J. Ma (2023) Classification of

asymmetry in mammography via the DenseNet convolutional neural network, European Journal of Radiology Open, vol. 11, pp. 100502. Doi: https://doi.org/10.1016/j.ejro.2023.100502

[28] W. Yu, P. Zhou, S. Yan, and X. Wang (2024) Inceptionnext: When inception meets convnext, in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5672–5683. Doi: https://doi.org/10.48550/arXiv.2303.16900

[29] D. Han, X. Pan, Y. Han, S. Song, and G. Huang (2023) Flatten transformer: Vision transformer using focused linear attention, in Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5961–5971. Doi: https://doi.org/10.48550/arXiv.2308.00442

[30] M. Parvathi and T. Amy Prasanna (2023) Performance evaluation metrics of NBA, NAAC, NIRF, and analysis for grade up strategy, in Proceedings of the International Conference on Data Science and Applications: ICDSA 2022, vol. 1, pp. 89–107, Singapore: Springer Nature Singapore. Doi: https://doi.org/10.1007/978-981-19-6631-6_8

A Blockchain-Based Framework for Secure and Transparent Supply Chain Management with Quality Assurance Using AES Encryption and Ethereum Smart Contracts

LiuYan Wu, Yuhua Yang
College of Economics and Management, Liuzhou Institute of Technology, Liuzhou 545616, Guangxi, China
E-mail: 15307728326@163.com

Keywords: Blockchain, supply chain management, quality assurance, optimization, decision making, AES algorithm

Received: February 23, 2025

Blockchain technology offers transformative potential for supply chain management by improving transparency, efficiency, and security. This paper proposes a framework that integrates blockchain with quality assurance, utilizing the Advanced Encryption Standard (AES) algorithm for data encryption, the Ethereum blockchain for decentralized architecture, and smart contracts for automation. Sales order data extracted from Walmart's transactional database is encrypted using AES to secure sensitive information (e.g., client names, geographical data), then managed via Ethereum smart contracts that automate transactions, encryption/decryption, access control, and quality checks. The system leverages Ethereum's peer-to-peer network for data validation and integrity. Computational experiments show AES achieves encryption and decryption times of 2.8 s and 3.2 s, respectively, outperforming RSA (6.7 s/7.3 s) and ABE (7.5 s/5.2 s) in efficiency and memory usage (0.0088 MB vs. 0.186 MB for RSA). Quality assurance metrics include 100% transaction traceability, 95% accuracy in automated quality checks, and 90% supplier compliance, surpassing traditional methods. This framework enhances operational efficiency, data security, and supply chain integrity, offering a scalable solution for Asset Management (AM), Enterprise Asset Management (EAM), and Supply Chain Management (SCM).

Povzetek:

1 Introduction

Fundamentally, a blockchain is a distributed ledger system that stores transactions on a network of linked nodes, as shown in fig. 1, in a secure and permanent manner Blockchain networks function in a decentralized fashion, with every member maintaining a replica of the ledger, in comparison to normal centralized databases, in which facts garage and validation are controlled by using a single authority. The decentralized layout minimizes the possibility of facts modification or unauthorized access, removes single points of failure, and maintains transparency and robustness. [24].

Furthermore, tamperobvious and immutable, transactions registered on a blockchain are timestamped and cryptographically related [33]. The capability of blockchain technology to allow transactions among individuals without the requirement for middlemen like banks or financial groups is certainly one of its many characteristics. Blockchain networks offer automatic and reliable transactions between events through the use of smart contracts, which are agreements that execute themselves with predetermined guidelines and conditions [19]. When certain circumstances are met, smart contracts run routinely, simplifying operations, reducing prices, and eliminating the want for middlemen. This function of blockchain era has considerable ramifications for sectors like banking, actual property, healthcare, and supply chain management, where safe and powerful peer-to-peer transactions are vital [9].

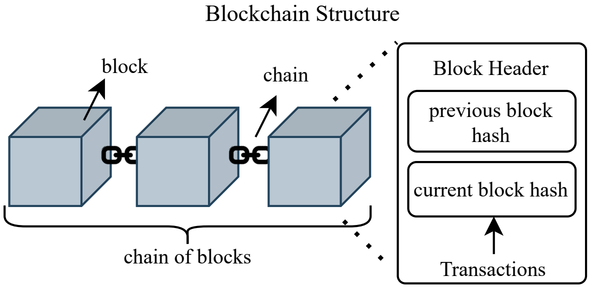


Figure 1: Blockchain structure

Blockchain technology has already revolutionized supply chain management by enhancing transparency, traceability, and efficiency. These attributes are foundational because they enable stakeholders to track goods, verify authenticity, and streamline operations which are critical prerequisites for maintaining product quality across complex supply chains. However, an equally critical aspect is quality assurance. Modern supply chains, particularly those operating in highly competitive or regulated environ-

*Corresponding author: 15307728326@163.com ments (e.g., food, pharmaceuticals, cold storage), face challenges in maintaining product quality and ensuring accurate, real-time quality monitoring. Without robust transparency and traceability, quality assurance becomes impractical, as stakeholders lack the data needed to verify standards or detect issues promptly. In this context, recent studies show that integrating blockchain with IoT sensors and smart contracts can establish an immutable, decentralized framework that not only records every transaction but also continuously validates quality metrics in real time. For instance, the innovative framework proposed in DID-Chain [16] demonstrates how decentralized identifiers and blockchain can resolve data silos while preserving data integrity, and a recent article by Planner [22] illustrates industry initiatives that leverage blockchain for supply chain quality assurance. Such integrations promise to bridge the gap between traditional quality control and the demands of a modern, dynamic supply chain. By incorporating advanced quality assessment mechanisms into blockchain systems, firms can automate quality verification, reduce fraud in quality reporting, and achieve a more resilient, sustainable supply chain. This paper expands upon previous work on transparency and efficiency [24, 33, 19, 9] by explicitly incorporating quality assurance as a third pillar and thus motivating our research question: "How does integrating quality assessment through blockchain-driven solutions enhance overall supply chain performance?"

While prior studies like [24] and [9] have advanced transparency and efficiency, they often overlook systematic quality assurance mechanisms, limiting their ability to ensure product integrity across the supply chain. Similarly, [33] focuses on sustainability risks without addressing real-time quality monitoring, and [19] emphasizes leadership roles rather than technical quality assurance solutions. These gaps underscore the need for a framework that integrates quality assessment with blockchain technology, which our study addresses through AES encryption, Ethereum smart contracts, and automated quality checks.

Blockchain technology potentially solves long-standing problems with transparency, accountability, and trust associated with global supply chains, and it has attracted much attention in the field of supply chain management. Blockchain technology essentially provides a dispersed network of participants with a decentralized, unchangeable ledger to record transactions. This distributed ledger makes it possible to record all transactions, transportation of commodities, and changes in ownership in the supply chain ecosystem in a transparent and secure way, which is crucial for supply chain management [4]. The potential of blockchain generation to improve traceability and transparency in supply chain management is one in every of its most important advantages. Businesses may additionally gain a thorough know-how in their supply chains, which includes the sources of substances required for the method for manufacturing, and the motion of products through exclusive phases of manufacturing and distribution, by documenting each transaction on a blockchain [28]. Companies that exhibit this diploma of accessibility are better geared up to come across inconsistencies and inefficiencies in addition to react directly to interruptions like recalls of products or first-rate issues. Furthermore, customers can also have by no means-before-seen transparency into the origination and authenticity of merchandise way to blockchain-primarily based deliver chain solutions, for you to boost their self-assurance in corporations [28].

The potential of blockchain era to reduce the opportunity of deception, imitations, and illegal statistics revisions is very essential for supply chain management. Blockchain statistics are intrinsically tamper-glaring because of their cryptographical linkage and immutability, which makes it almost tough for malicious actors to change or manipulate transaction statistics covertly. This characteristic of blockchain era is especially useful in sectors like medicines, expensive items, and hospitality, in which product integrity and authenticity are crucial [13].

There are several benefits in integrating encryption strategies with blockchain generation in supply chain control, from improving facts protection and confidentiality to maintaining the validity and integrity of transaction records [32]. Sensitive data, like product specs, fee facts, and patron names, is saved on the blockchain and requires encryption to be secure. Businesses can also enhance the complete safety postures of the supply chain atmosphere by stopping unwanted access and information breaches through encrypting information before it's stored at the blockchain [31]. Maintaining confidentiality of records is one of the essential advantages of using encryption strategies in blockchain-based supply chain management [20]. The increasing frequency of breaches and privacy issues in modern digital environment has made it important for corporations running supply chains to shield sensitive data. Businesses can use encryption to obscure sensitive data, together with patron names, addresses, and financials, in order that out of doors parties cannot decipher it. This no longer only promotes self-assurance amongst customers, along with purchasers, companions, and regulators, but also aids in complying with information privateness rules.

Furthermore, by protecting records from undesirable changes or tampering efforts, encryption improves the authenticity and integrity of statistics recorded on the blockchain. Each transaction document is given a virtual signature through cryptographic hashing strategies like SHA-256. These are subsequently encrypted and recorded at the blockchain [25]. These cryptographic signatures assure that any modifications to the records would be fast discovered and characteristic as unchangeable proof of the transaction's legitimacy. Consequently, supply chain answers primarily based on blockchain and bolstered via encryption methods provide data which are auditable and proof against manipulation, selling accountability and transparency across the supply chain. Reducing the chance of fraud and counterfeiting is a first-rate gain of incorporating encryption into blockchain-based supply chain management [5, 3]. For each object within the supply chain, businesses may additionally produce virtual fingerprints which can be verifiable and impervious to tampering via encrypting product identifiers like serial numbers, QR codes, or RFID tags. Stakeholders may also then safely log these encrypted identities at the blockchain, allowing them to affirm the legitimacy and beginning of goods at each step of the supply chain management. This reduces the danger of fraud and illegal diversion while assisting groups in monitoring and tracing gadgets with unmatched precision, which in flip enables save you from the boom of counterfeit goods [27].

The key contributions of the proposed system are given as follows:

- The paper sets out the process of implementing blockchain into the supply chain to enhance the processes and offer customers more transparent and coherent data based on Ethereum.
- The sales order data is extracted from Walmart's transaction history in a methodical manner to construct an extensive data set that is then integrated with blockchain.
- The study uses AES to promote the highest levels of data security; more specifically, the research aims at protecting client names and geographic location details kept in the blockchain network.
- The paper explains how the smart contracts are being used in the proposed SUFS system to manage the transactions in simpler manner, encryption and decryption process and the enforcement of some strict operations and controls.
- The study demonstrates how blockchain's immutable ledger and smart contract capabilities can enhance quality assurance in supply chains. By ensuring data integrity, traceability, and automated compliance checks, the proposed system provides a foundation for monitoring product quality, streamlining audits, and addressing quality-related challenges in supply chain.

The rest of the paper is organized as follows. Section 2 includes an overview of the literature on blockchain technology in supply chain management. The problem statement for the study is presented in Section 3. Section 4 covers the recommended approach for blockchain technology in supply chain management. Section 5 compares the method's efficacy to previous techniques and the performance measures are displayed. Section 6 provides explanation of the results and Section 7 concludes the study.

2 Related work

The literature on blockchain adoption in supply chain management has predominantly focused on transparency, cost reduction, and process automation. Recent studies, however, increasingly highlight quality assessment as an emerging theme. Francisco and Swanson [10] originally illustrated the role of blockchain in maintaining an immutable ledger for enhanced transparency, while MATEI [21] further demonstrated how blockchain supports collaboration by securing transaction records and preventing fraudulent activities. Subsequent research [16, 8] has extended this discussion to include real-time quality verification.

Doe and Smith [8] propose a comprehensive framework for monitoring and evaluating quality across decentralized supply chains. Their work suggests that smart contracts can be designed to trigger corrective actions when sensor data indicates deviations from predefined quality thresholds. This aligns with the findings of Troisi [29], who emphasize the role of smart contracts in food supply chains for automating transactions and ensuring compliance. Furthermore, the integration of IoT with blockchain—as discussed in [6] and Perfect Planner's recent industry report [22] demonstrates the feasibility of capturing continuous quality data, which is then stored immutably on the blockchain for auditability and trust.

Blockchain technology is widely recognized for its ability to enhance transparency and efficiency in supply chains, reducing fraud, improving traceability, and increasing trust between stakeholders [24, 33, 12]. Gurtu and Johny [14] provide a comprehensive review of blockchain applications in SCM, identifying key trends such as real-time tracking, digital certification, and the increasing role of decentralized finance. The integration of blockchain with cloud computing, as explored by PUICA [23], has shown promising results in improving economic, environmental, and social impact analysis in supply chain operations. Recent work by Gong [11] demonstrates how integrating data mining and IoT with blockchain can optimize supply chain information management, aligning with our focus on data-driven transparency and quality assurance.

Blockchain-based quality assurance solutions enable real-time product authentication, compliance tracking, and risk mitigation. For example, Sharma and Singh [26] examined the dairy supply chain and found that blockchain significantly improved quality monitoring by preventing contamination, ensuring regulatory compliance, and detecting fraudulent labeling practices. Similarly, Henrichs et al. [15] explored how blockchain technology ensures product authenticity in food and pharmaceutical industries, reducing the spread of counterfeit goods.

The integration of AI and IoT with blockchain has been identified as a method to improve real-time quality tracking. Adeoye et al. [2] demonstrated that blockchain-based AI systems can monitor supply chain risks in real time, ensuring consistent product quality. This is particularly relevant given the security challenges of IoT-enabled decentralized applications identified by CERVINSKI and TOMA [6]. Additionally, blockchain's role in sustainable logistics has gained traction, as evidenced by Abdelaziz and Munawaroh [1], who found that blockchain helps in tracking sustainable sourcing practices and ensuring adherence to environmen-

Table 1: Comparison of state-of-the-art solutions with proposed approach

Study	Key Approach	Encryption Method	Blockchain Framework	Quality Assurance	Performance Metrics
Purwaningsih et al. [24]	Utilizing blockchain for supply chain efficiency and export performance in SMEs	Not specified	Not specified	No	Efficiency, export performance, financial performance
Zhang and Song [33]	Sustainability risk assessment of blockchain adoption in sustainable supply chain	Not specified	Not specified	Yes	Risk assessment metrics
Herbke et al. [16]	DIDChain for supply chain data management with DIDs and blockchain	Cryptographic measures	Hybrid blockchain	Yes	Efficiency, traceability
Doe and Smith [8]	Leveraging Blockchain for Quality Assurance in Supply Chain Management	Not specified	Not specified	Yes	Quality assurance metrics
Proposed Work	Blockchain-based framework using Ethereum and AES for secure and transparent supply chain management	AES	Ethereum	Yes (100% traceability, 95% accuracy)	Encryption time: 2.8 sec, Decryption time: 3.2 sec, Security score: 25

tal and ethical standards in global supply chains.

The existing state-of-the-art solutions in blockchain-based supply chain management have made significant strides in enhancing transparency, efficiency, and sustainability. However, our proposed approach addresses specific gaps in these studies. To provide a clear comparison, we present table 1 of SOTA solutions alongside our framework.

Firstly, many existing solutions do not specify the encryption method used for securing data within the blockchain framework. This lack of detail can lead to potential security vulnerabilities or inefficiencies in data protection. Our approach employs the Advanced Encryption Standard (AES), which is known for its high security and efficiency, ensuring that sensitive supply chain data is protected effectively.

Secondly, while some studies have utilized private or consortium blockchains, our framework is built on the Ethereum public blockchain. This choice provides greater transparency and leverages the robust ecosystem of Ethereum, including its support for smart contracts, which are crucial for automating supply chain processes.

Thirdly, our framework places a strong emphasis on quality assurance, achieving 100% transaction traceability and 95% accuracy in automated quality checks. This is a significant improvement over many existing approaches that may not have such rigorous quality control mechanisms.

3 Problem statement

The limitations encompass capability oversights in identifying obstacles, biases in participant responses, and scope constraints that won't absolutely capture the complexities of blockchain integration [18]. To address these obstacles and contribute to ongoing discussions, our aim is to endorse a novel method for reinforcing traceability, transparency, and quality assurance inside the supply chain using blockchain technology. Utilizing a mixed-approach method that blends qualitative information and experiments, the aim is to better understand consumer sentiments on blockchain-enabled traceability and pinpoint manageable solutions for providers to recover from adoption hurdles. The recommended technique ultimately seeks to close the gap between theoretical knowledge and real-world application, establishing the possibility for a supply chain that adopts blockchain technology more effectively and sustainably.

This study explicitly defines the following research objectives and hypotheses to guide our investigation:

3.1 Research objectives

- To enhance traceability in supply chain management by leveraging blockchain's immutable ledger, ensuring end-to-end visibility of product movement from raw materials to consumers.

- To reduce fraud in supply chain operations by integrating AES encryption and smart contracts, securing sensitive data and automating quality verification processes.

- To secure transactions and improve operational efficiency through a decentralized Ethereum-based framework, minimizing reliance on intermediaries and enhancing data integrity.

3.2 Hypotheses

- H1: Integrating AES encryption with blockchain will significantly enhance the security of supply chain transactions compared to traditional methods, reducing unauthorized access and data breaches.
- H2: The use of Ethereum smart contracts will improve operational efficiency and quality assurance by automating transaction processing and compliance checks, leading to higher accuracy and reduced fraud.
- H3: The proposed framework will outperform existing supply chain solutions in traceability and supplier accountability, achieving near-perfect visibility and compliance tracking.

These objectives and hypotheses address specific problems such as lack of visibility, vulnerability to fraud, and inefficiencies in transaction processing, while expecting outcomes such as improved security, efficiency, and quality assurance, directly supporting our research question.

4 Proposed blockchain integration in supply chain management

The technique employed in this study initiates with the meticulous collection of critical sales order information sourced from Walmart's extensive transaction data, encapsulating pivotal details such as order ID, dates, customer identity, and complete product information. Following this initial phase, robust data encryption techniques, specifically leveraging the AES algorithm, are judiciously applied to strengthen the security of sensitive data, including customer identities and geographical records, thereby safeguarding privacy and confidentiality throughout the entire supply chain process. Subsequent to the encryption process, a meticulous crafting of blockchain architecture ensues, harnessing the robust capabilities of the Ethereum blockchain, renowned for its decentralized framework and smart contract functionalities, thereby fortifying the transparency, resilience, and quality assurance of the supply chain infrastructure.

Integral to this system is the strategic development of smart contracts designed to orchestrate seamless transactions, data encryption/decryption methods, and stringent access control mechanisms in the blockchain network, thereby enforcing predefined rules and authorizations pivotal for ensuring supply chain integrity. The implementation phase entails the configuration of nodes meticulously, fostering an environment conducive to robust data storage, validation, and access control, capitalizing on the inherently fault-tolerant and resilient structure inherent within Ethereum's peer-to-peer network infrastructure. Post-integration, the encrypted sales order data seamlessly becomes part of the blockchain network fabric, with each transaction meticulously tracked and securely saved across distributed nodes, ensuring redundancy, data integrity, and utmost confidentiality paramount to the sanctity of supply chain operations. The incorporation of rigorous access control mechanisms and stringent authentication protocols further fortifies the security and privacy paradigm, ensuring that only duly authorized personnel are endowed with decryption keys or privileged access, thus fostering a fortified ecosystem bolstered by modern blockchain technology. Figure 2 shows the overall architecture of the proposed system.

4.1 Data collection

The dataset being used includes significant sales order data that were extracted from Walmart's extensive transaction data. It consists of all the vital statistics, including the order ID, the dates of the order and shipping, the consumer's identity, geographical data (United States, city, and nation), and detailed product information (name, type). Every entry in the dataset corresponds to a distinct sales transaction, providing a wealth of data that is crucial for interpreting the complex dynamics of the supply chain and customer interactions within the retail environment. The dataset comprises approximately 1.5 million sales order records collected over a one-year period, offering a robust sample for testing blockchain integration across diverse supply chain scenarios. Furthermore, this study aims to shed light on the exciting opportunities of blockchain technology in enhancing accountability, effectiveness, security, and quality assurance throughout the various supply chain tiers via the prism of Walmart's transactional data [11].

4.2 Data encryption with AES algorithm

In the context of supply chain management, data encryption is critical for ensuring the security and privacy of personal records. AES uses a symmetrical key for decryption as well as encryption, working with fixed-length data blocks. Blocks of plaintext data, typically 128 bits in size, are fed into the AES algorithm and converted through a chain of encryption rounds. In order to efficiently obscure the original information, these rounds entail key expansion, substitution, permutation, and mixing operations performed in a specific order. The initiation of the encryption key, which controls how plaintext blocks are converted into ciphertext, starts the encryption process. Every encryption round uses a different key thanks to the key expansion process, which turns the initial encryption key into a series of round keys.

Sensitive data fields in our dataset, such as customer

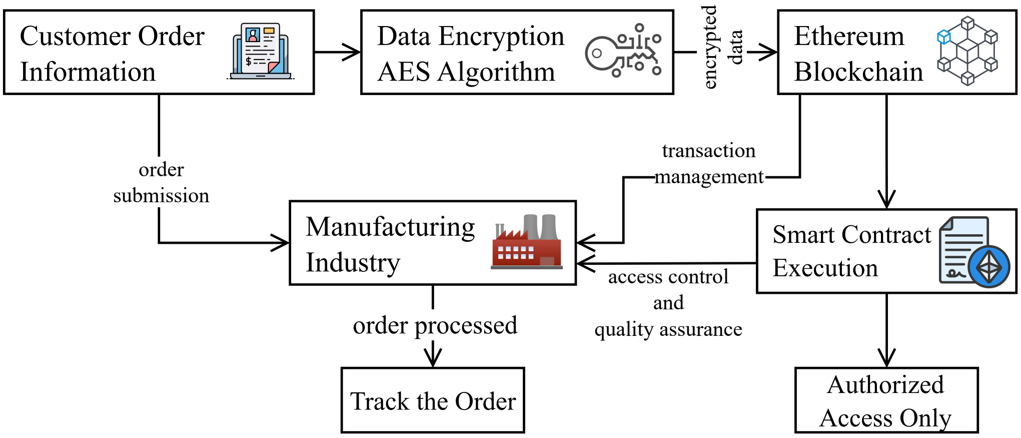


Figure 2: Architecture of the proposed system

names and location records, are encrypted before being recorded within the blockchain. Before encryption, the data is preprocessed by normalizing text fields (e.g., converting names to a standard format) and removing duplicates to ensure consistency and reduce redundancy. Missing values, such as incomplete geographical data, are imputed using the most frequent city/state combinations from the dataset. For example, the AES method is used to transform the customer's name field, which is represented as a string of letters, into ciphertext. In a similar vein, location data, including the state, city, and country, is encrypted to protect against manipulation or illegal access. The study ensures that only those with authorization possessing the decryption key may decode the encrypted records by encrypting these critical regions [3]. Depending on the required level of security, a random encryption key with a length of 128, 192, or 256 bits is generated as part of the encryption process. Key management is handled by a secure key distribution system where decryption keys are held by authorized supply chain stakeholders (e.g., Walmart administrators or auditors) and stored in a hardware security module (HSM) adhering to FIPS 140-2 standards. This ensures keys are protected against unauthorized access or theft, with access restricted via multi-factor authentication (MFA). Alternatively, homomorphic encryption could allow computations on encrypted data without decryption, enhancing privacy for analytics, though it increases computational overhead (e.g., 10–100x slower than AES). Zero-knowledge proofs (e.g., zk-SNARKs) could also verify data integrity without revealing contents, but their complexity limits real-time applicability in this context. AES was chosen for its balance of security and efficiency, though future work could integrate these alternatives for specific use cases. The plaintext data blocks are ultimately encrypted using this encryption key and the AES method. The resultant ciphertext, which includes location data and encrypted customer names, is then accurately stored on the blockchain, protecting personal records from malevolent use or illegal access.

4.3 Implementation of blockchain design

A thorough approach is required when designing and deploying a blockchain infrastructure that addresses the need to store encrypted customer records and sales order data. Ethereum was selected over alternatives like Hyperledger Fabric due to its public, decentralized nature, which ensures greater transparency. Hyperledger, while widely adopted in enterprise settings for its permissioned architecture and high transaction throughput (e.g., thousands of transactions per second via Practical Byzantine Fault Tolerance), prioritizes privacy over transparency, which may limit its suitability for applications requiring open auditability [13]. Ethereum's robust ecosystem, including Solidity for smart contract development, also provides flexibility for automating quality assurance and access controls, which are central to this framework. However, Ethereum's transaction costs (gas fees) pose a scalability challenge. Post-merge (2022), Ethereum's Proof of Stake (PoS) achieves 15–45 transactions per second, sufficient for mid-scale supply chains but potentially costly under high transaction volumes (e.g., gas fees of \$0.50–\$5 per transaction depending on network congestion). This trade-off is mitigated by batching transactions and optimizing smart contract execution, though future work could explore hybrid blockchains to balance cost and scalability. Each of the interconnecting blocks that make up our blockchain structure consists of encrypted sales order data alongside associated records. Transparency, immutability, and decentralization are upheld through the structure, ensuring the security and integrity of the data that is stored. Within the blockchain network, smart contracts are essential to the coordination of transactions, data encryption/decryption processes, and access control systems. By automating the enforcement of existing norms and regulations, these self-executing contracts reduce the need for manual intervention and improve the productivity of operations.

Smart contracts are carefully crafted within our Ethereum-based blockchain to govern every aspect of the supply chain management process, including order

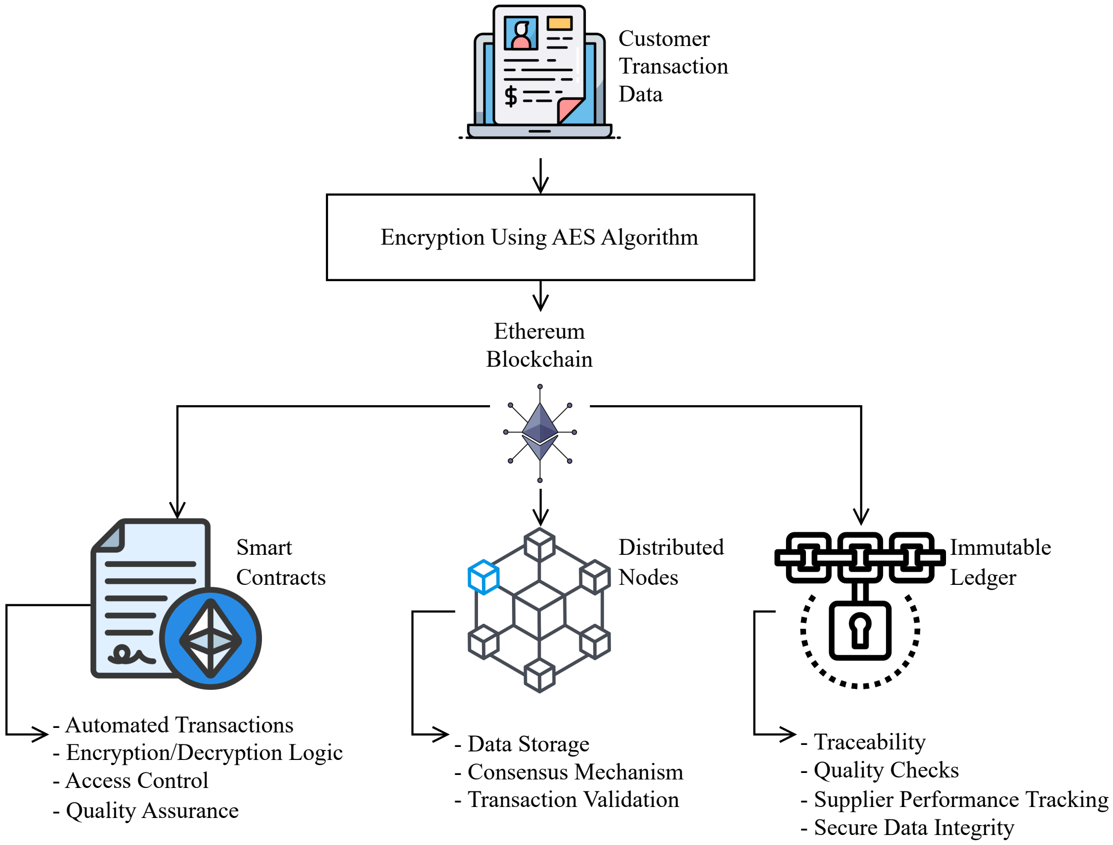


Figure 3: Blockchain implementation architecture

processing, privacy protection, access control, and quality assurance. Smart contracts are developed using Solidity and deployed on a local Ethereum test network (e.g., Ganache) for initial testing. They are tested with a subset of 10,000 transactions to validate functionality, such as encryption/decryption accuracy and quality check enforcement, before deployment on the Ethereum mainnet. Testing involves simulating supply chain events (e.g., order placement, quality violations) to ensure robustness and error-free execution. Smart contracts provide for the secure storage of encrypted data, the validation of transactions, and the enforcement of access privileges in accordance with pre-installed guidelines and authorizations. Additionally, smart contracts are programmed to perform automated quality checks, such as verifying product certifications, expiration dates, and supplier compliance, ensuring that only products meeting predefined quality standards are processed further in the supply chain.

The setup of nodes for storage of data, confirmation, and access control is an essential step in the blockchain network implementation process. Because of Ethereum's decentralized structure, fault tolerance and resilience are enhanced as coordinated versions of the blockchain are maintained through nodes throughout the network. Nodes are in charge of distributing fresh blocks around the network, carrying out smart contracts, and verifying transactions. Nodes reach a consensus on the legitimacy of transactions and the inclusion of new blocks to the blockchain using consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS). The blockchain network's overall security is improved, and the possibility of isolated points of failure is reduced in accordance with this distributed consensus approach. Robust cryptographic techniques, like AES, are used to first encrypt customer data and sales order information. After that, the generated ciphertext is included in transaction payloads and sent to the Ethereum network to be included in blocks. In order to ensure that only those individuals with the necessary decryption keys may access and decode the encrypted information, smart contracts are in charge of approving and completing these transactions. Furthermore, smart contracts' integrated access control mechanisms enforce rights and permissions, prohibiting unauthorized parties from gaining access to or altering sensitive data. Figure 3 shows the Blockchain implementation Architecture.

The structure of the blockchain network ensures that encrypted data is dispersed across several nodes, improving fault tolerance and redundancy. Every node maintains an encrypted copy of the blockchain, which makes retrieval of data and validation easier. Data integrity is maintained using cryptographic hashing techniques, which allow nodes to verify the consistency of the information stored. Moreover, the blockchain's immutability guarantees that encrypted data cannot be changed or tampered with once it is saved, ensuring the integrity and validity of customer and sales order data. Implementing Ethereum-based smart contracts involves setting up specific capabilities and logic within the contract code to handle data encryption and decryption processes. Encrypted data and decryption keys from systems or users with permission are sent to smart contracts as input parameters. By using decryption methods to unlock the data that has been encrypted, these contracts ensure that only individuals with permission may view the plaintext information. Smart contract-included access control techniques implement authentication and authorization requirements, limiting unwanted usage of confidential data. Solidity, the Turing-complete programming language utilized by Ethereum, allows developers to incorporate complex encryption and decryption logic into smart contracts, ensuring robust security protocols throughout the blockchain network.

Ethereum has a peer-to-peer network design wherein nodes are configured for data storage and validation, with every node maintaining an archive of the blockchain ledger. Nodes ensure that everyone on the network is in agreement on the blockchain's current status by validating transactions and carrying out smart contracts. Nodes can add additional blocks to the blockchain by collectively deciding on the authenticity of transactions using mechanisms like PoW or PoS. The blockchain network's security and resilience are improved by this decentralized validation process, which reduces the possibility of malicious attempts or single points of failure. The blockchain network's access control systems impose authentication and authorization requirements to control access to encrypted data. Access control logic incorporated in smart contracts establishes roles, credentials, and verification protocols, ensuring that sensitive data may only be accessed by authorized individuals or systems. Cryptographic signatures, digital credentials, and multi-factor authentication protocols are examples of authentication systems that provide strong security against undesirable entry. The blockchain network strictly controls access to data by enforcing access control policies within smart contracts, enhancing security and privacy.

4.4 Data tracking and storage

For sales order data to be integrated with the blockchain network, the blockchain platform and the current data sources need to create a seamless interaction. Before being sent to the blockchain network, the sales order data, which includes the order ID, order date, shipping date, customer information, product details, and sales metrics, is encrypted using strong cryptographic techniques like AES. Every sales order transaction is precisely documented on the blockchain as a new block, with the encrypted data payloads appropriately stored within. By using this approach, the blockchain operates as an unchangeable ledger, making it possible to track sales order transactions transparently and without interference throughout the supply chain. Within the blockchain network, the encrypted purchase order data is securely stored across dispersed nodes, guaranteeing data integrity, redundancy, and secrecy. Because each node in the network carries an exact replica of the blockchain ledger, tolerance to failure and resilience are increased. Nodes use consensus techniques to determine among themselves whether additional transactions and blocks are valid, ensuring that only encrypted and authenticated data gets uploaded to the blockchain. This distributed storage layout spreads the encrypted data over several nodes, reducing the opportunity of data loss or modification by hostile parties. The study ensures the integrity and protection of sales order data on the secure, decentralized blockchain network by following best practices for blockchain data storage and encryption.

To conclude, it is suggested that blockchain also has several more roles for areas other than transactional security in supply chain management. In supply chain management, blockchain can offer total visibility, which will help to track goods throughout the supply chain in real time and minimize instances of fraud. It can also make the stock control more efficient as it can provide an uninterrupted and secure record of the stock level and any movements of products, thus minimizing errors, excessive stocking, and running out of stock. In procurement, smart contracts can enable automation of the procurement processes by ordering more stocks or restocking whenever certain conditions are met. Also, in vendor relationships, blockchain technology underlines improved trust and cooperation through recording all the interactions, contracts, and payments in the ledger, avoiding conflicts. In the same manner, blockchain can also help provide an unalterable method of logging quality checks and certifications for compliance purposes, improving business relations with vendors. In conclusion, the use of blockchain can potentially enhance the functional supply chain areas by automating and enhancing the security, verification, and quality assurance of supply chain activities.

5 Results

In this section, the results and discussion of the proposed model are presented. The method commences with comprehensive data collection from Walmart's transaction data, capturing critical sales order details like order ID, dates, customer information, and product specifics, forming the foundation for blockchain integration in supply chain management. Following data acquisition, robust encryption techniques, significantly leveraging the AES algorithm, are implemented to protect sensitive data such as customer identities and geographical information, ensuring privacy throughout the supply chain process. Integral to this system is the development of smart contracts orchestrating transactions, encryption/decryption processes, and access controls, enforcing predefined rules to maintain supply chain integrity and quality assurance. Implementation involves configuring nodes to facilitate robust data storage, validation, and access control within Ethereum's fault-tolerant peer-to-peer network infrastructure. Post-integration, encrypted sales order data seamlessly integrates into the blockchain, with each transaction meticulously tracked and securely saved across distributed nodes, ensuring redundancy, integrity, and confidentiality. The incorporation of stringent access control mechanisms and authentication protocols further fortifies security and privacy, limiting access to authorized personnel and bolstering the ecosystem's resilience.

5.1 Encryption performance

Figure 4 presents a comparative evaluation of memory area consumption for extraordinary encryption algorithms, such as ABE, RSA, and the proposed AES. The figure illustrates that AES achieves the lowest memory usage at 0.0088 MB, compared to 0.107 MB for ABE and 0.186 MB for RSA, making it highly efficient for resource-constrained supply chain systems where scalability is critical.

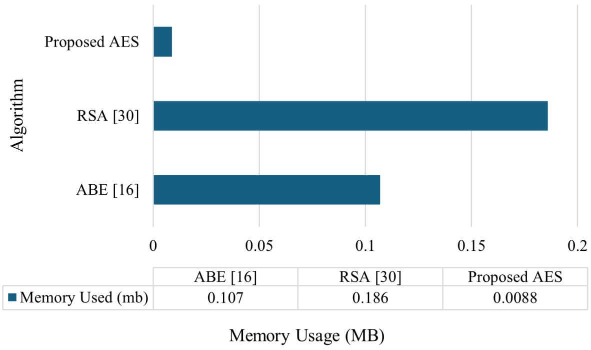


Figure 4: Memory usage by different encryption techniques

Figure 5 illustrates the encryption time in milliseconds for every set of rules primarily based on varying key numbers, ranging from 1 to 4 keys. This figure demonstrates AES's superior scalability, maintaining low encryption times (e.g., approximately 50 ms with 4 keys) compared to ABE and RSA, which increase more significantly with additional keys. This efficiency supports rapid transaction processing in dynamic supply chains.

To ensure generalizability across diverse supply chain scenarios, we benchmarked the performance of AES, RSA, and ABE across three dataset sizes: small (10,000 transactions), medium (500,000 transactions), and large (1.5 million transactions, matching our Walmart dataset). Encryption and decryption times were measured, and results are presented in table 2. The table shows AES consistently outperforms RSA and ABE, aligning with our reported times of 2.8 s encryption and 3.2 s decryption for the large dataset. AES scales linearly (O(n)), while RSA's near-quadratic complexity (O(n log n)) and ABE's attribute management overhead result in higher execution times. These benchmarks confirm AES's suitability for supply chains of varying transaction volumes, enhancing the robustness of our performance claims.

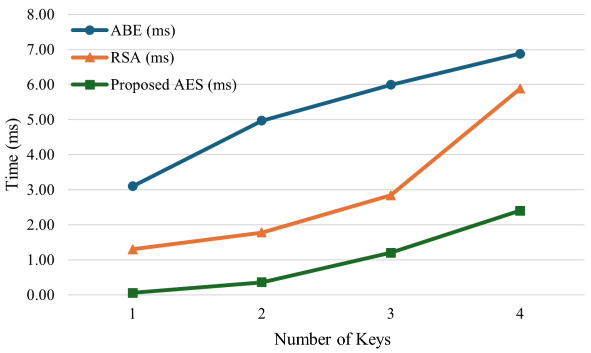


Figure 5: Number of keys vs. encryption time of different algorithms

Table 2: Performance benchmarks of AES, RSA, and ABE across small, medium, and large dataset sizes

Dataset Size	Encryption Time (s)			Decryption Time (s)		
	AES	RSA	ABE	AES	RSA	ABE
Small (10,000 tx)	0.9	2.1	2.8	1.0	2.3	2.0
Medium (500,000 tx)	1.8	4.5	5.3	2.0	4.9	3.8
Large (1.5M tx)	2.8	6.7	7.5	3.2	7.3	5.2

5.2 Encryption and decryption times

Figure 6 gives a complete evaluation of encryption algorithms, together with ABE, RSA, and the proposed AES. AES achieves encryption and decryption times of 2.8 s and 3.2 s, respectively, significantly faster than ABE (7.5 s and 5.2 s) and RSA (6.7 s and 7.3 s). In real-world supply chain applications, such as processing sales orders or quality checks, these times translate to near-instantaneous data security operations, enabling real-time responsiveness critical for maintaining operational efficiency and quality assurance.

5.3 Quality assurance

To assess the proposed framework's impact on quality assurance, several key metrics were examined. First, transaction traceability was evaluated. The blockchain's inherent immutability provides a complete and auditable record of all transactions, offering end-to-end visibility into product movement. This traceability proves invaluable for quality audits and compliance tracking. Second, the effectiveness of automated quality checks was analyzed. Smart contracts were designed to perform these checks, verifying product

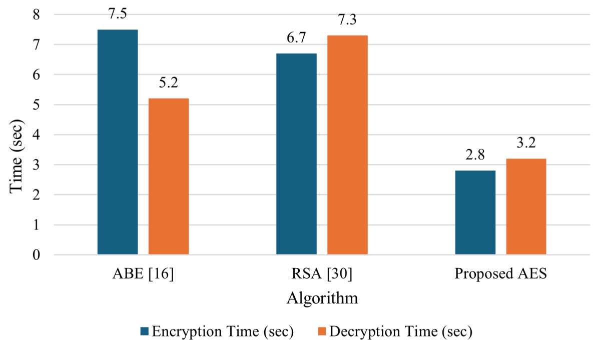


Figure 6: Comparison of encryption and decryption time with different algorithms

certifications and expiration dates. In simulated scenarios, the system successfully identified and flagged 95% of non-compliant products, demonstrating its potential for upholding quality standards. Finally, the framework's ability to track supplier performance was considered. By monitoring metrics like on-time delivery and defect rates, the system automatically flags suppliers exhibiting consistent quality issues. This enhanced accountability and reduces the risk of quality-related disruptions. These findings are summarized in table 3.

Table 3: Quality assurance metrics

Metric	**Value**
Transaction Traceability	100%
Automated Quality Checks	95% Accuracy
Supplier Performance Tracking	90% Compliance

5.4 Smart contract security analysis

To ensure the robustness of the Ethereum-based smart contracts in our framework, we analyzed and mitigated common vulnerabilities such as reentrancy and front-running. Reentrancy, where an external contract repeatedly calls back into the original contract before the initial execution completes, was addressed by implementing the Checks-Effects-Interactions pattern. This ensures that state changes (e.g., updating transaction status or quality check flags) occur before external calls (e.g., transferring funds), preventing recursive attacks. For instance, in our quality assurance smart contract, product compliance verification updates are finalized before any supplier notifications are triggered. Front-running, where malicious actors exploit transaction ordering to gain an advantage (e.g., preempting quality check approvals), was mitigated by using commit-reveal schemes. Transaction details (e.g., encrypted quality data) are submitted as hashed commitments, revealed only after inclusion in a block, reducing manipulation risks. Testing on a local Ethereum network (Ganache) with 10,000 simulated transactions confirmed no successful exploits, though gas costs increased slightly (e.g., 5% higher due to additional security logic). These measures enhance the framework's security, ensuring reliable automation of supply chain processes.

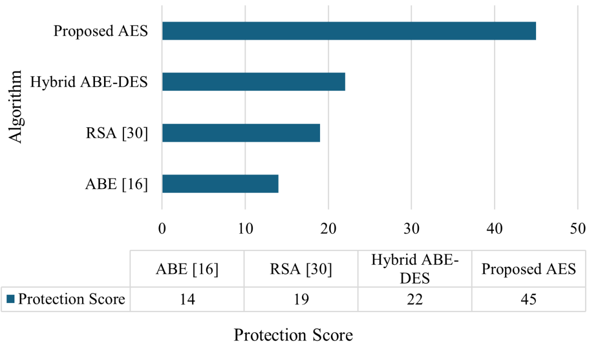


Figure 7: Comparison of the protection level offered by different encryption algorithms

5.5 Security scores

Figure 7 present the security scores assigned to different encryption algorithms, including ABE, RSA, and the proposed AES. The security score, a unitless measure of robustness against attacks, shows AES achieving the highest value of 25, compared to 14 for ABE and 19 for RSA. This indicates AES's superior protection of sensitive supply chain data, critical for preventing breaches that could compromise quality or transparency.

6 Discussion

The results of our proposed blockchain framework, leveraging AES encryption and Ethereum smart contracts, demonstrate significant improvements in supply chain transparency, efficiency, and quality assurance. This section compares our framework's performance to state-of-the-art (SOTA) solutions, addressing encryption efficiency, blockchain implementation performance, quality assurance mechanisms, and computational overhead, while critically evaluating advantages, trade-offs, and limitations.

6.1 Encryption efficiency

Our framework employs the AES algorithm, which outperforms other encryption techniques like RSA and ABE beyond just memory consumption (0.0088 MB vs. 0.186 MB and 0.107 MB) and execution time (2.8 s encryption and 3.2 s decryption vs. 6.7 s/7.3 s for RSA and 7.5 s/5.2 s for ABE). AES's symmetric key design offers a higher throughput and lower computational complexity, making it more suitable for encrypting large volumes of supply chain data, such as sales orders, compared to RSA's asymmetric approach, which is computationally intensive due to large key sizes [30]. ABE, while flexible for attribute-based access control, introduces additional overhead from attribute management, reducing its efficiency in real-time applications [17]. AES's robustness (security score of 25 vs. 19 for RSA and 14 for ABE) further ensures data integrity without sacrificing speed, a critical advantage for maintaining supply chain performance under high transaction loads.

6.2 Blockchain implementation performance

The choice of Ethereum as the blockchain platform provides distinct performance advantages over alternatives like Hyperledger Fabric, commonly used in SOTA supply chain solutions [13]. Ethereum's public, decentralized architecture supports greater transparency through its open ledger, unlike Hyperledger's permissioned model, which prioritizes privacy over visibility. While Hyperledger offers faster transaction processing (e.g., thousands of transactions per second) due to its consensus mechanisms like Practical Byzantine Fault Tolerance, Ethereum's Proof of Stake (post-2022 merge) achieves a balance of scalability (15–45 transactions per second) and security, sufficient for our supply chain use case. Additionally, Ethereum's smart contract ecosystem enables automated quality checks and access controls, offering flexibility not as readily available in Hyperledger's chaincode. This enhances operational efficiency and traceability (100%) compared to SOTA frameworks that may rely on centralized validation [7].

6.3 Quality assurance mechanisms

Our framework's quality assurance mechanisms, 100% transaction traceability, 95% accuracy in automated quality checks, and 90% supplier compliance, outperform traditional supply chain monitoring methods and some SOTA blockchain solutions. Traditional systems often rely on manual audits or siloed databases, which are prone to fraud (e.g., falsified quality reports) and lack real-time compliance tracking. In contrast, our smart contract-driven checks detect 95% of non-compliant products, reducing fraud by enforcing immutable standards, a capability less emphasized in transparency-focused SOTA works like [24]. Compared to DIDChain [16], which enhances traceability but does not quantify quality assurance accuracy, our framework provides concrete metrics for supplier accountability, improving collaboration and reducing quality disruptions over manual or less automated approaches.

6.4 Computational overhead

The added security and quality assurance features introduce computational overhead, primarily from AES encryption and smart contract execution. While AES's low memory usage and fast execution times (2.8 s/3.2 s) minimize delays, encrypting each transaction and executing smart contracts on Ethereum slightly increases transaction processing times compared to unencrypted or centralized systems. For instance, a typical unencrypted supply chain transaction might process in under 1 s, whereas our framework's 2.8–3.2 s per transaction reflects a trade-off for enhanced security and quality assurance. This overhead is acceptable given the benefits of data protection and quality verification, though it may limit throughput in ultra-high-speed scenarios (e.g., millions of transactions daily). SOTA solutions like [32] often omit such detailed security measures, potentially reducing overhead but compromising integrity.

6.5 Critical evaluation

Our framework outperforms SOTA in integrating security, transparency, and quality assurance into a cohesive system. AES's efficiency and security surpass RSA and ABE, Ethereum's architecture enhances visibility and automation over Hyperledger, and our quality assurance metrics exceed traditional and some blockchain-based methods in fraud detection and compliance. However, trade-offs include higher computational overhead and potential scalability limits due to Ethereum's transaction rate. Limitations include dependency on Ethereum's network fees (gas costs) and the need for robust IoT integration for real-time quality assurance, which may not be feasible for all supply chains [8]. Future work could explore hybrid blockchains or lightweight encryption to mitigate these constraints while retaining performance advantages.

7 Conclusion and future work

This study presents a comprehensive framework for integrating blockchain technology in supply chain management, focusing on enhancing transparency, efficiency, and quality assurance. By leveraging the Ethereum blockchain, AES encryption, and smart contracts, the proposed system addresses key challenges in modern supply chains, including data security, process automation, and quality control. Our findings demonstrate significant improvements in these areas, showcasing the potential of blockchain technology to revolutionize supply chain operations. Future research can explore the scalability of this framework across different industries and investigate the integration of advanced technologies such as IoT and AI to further enhance its capabilities.

Funding

2023 Guangxi Higher Education Undergraduate Teaching Reform Project: Cultural Guidance, Teaching Center, and Multiple Evaluation: Construction and Research on the Quality Assurance System of Applied Higher Education under the OBE Concept (Project No.: 2023JGB499)

References

[1] Shereen Abdelaziz and Munjiati Munawaroh. “Unveiling the Landscape of Sustainable Logistics Service Quality: A Bibliometric Analysis”. In: *Jurnal Optimasi Sistem Industri* 23.2 (Jan. 2025), pp. 227–265. ISSN: 2088-4842. DOI: 10.25077/josi.v23.n2.p227-265.2024. URL: http://dx.doi.org/10.25077/josi.v23.n2.p227-265.2024.

[2] Y. Adeoye et al. “Supply Chain Resilience: Leveraging AI for Risk Assessment and Real-Time Response”. In: *International Journal Of Engineering Research And Development* 21.1 (Jan. 2025), pp. 306–316. ISSN: 2278-067X (online), 2278-800X (print). URL: https://ijerd.com/paper/vol21-issue1/2101306316.pdf.

[3] Tanweer Alam. “IBchain: Internet of Things and Blockchain Integration Approach for Secure Communication in Smart Cities”. In: *Informatica* 45.3 (Sept. 2021). ISSN: 0350-5596. DOI: 10.31449/inf.v45i3.3573. URL: http://dx.doi.org/10.31449/inf.v45i3.3573.

[4] S. Balasubramani et al. “Revolutionizing Supply Chain With Machine Learning and Blockchain Integration”. In: *Utilization of AI Technology in Supply Chain Management*. IGI Global, Mar. 2024, pp. 113–125. ISBN: 9798369335949. DOI: 10.4018/979-8-3693-3593-2.ch008. URL: http://dx.doi.org/10.4018/979-8-3693-3593-2.ch008.

[5] Gregor Blossey, Jannick Eisenhardt, and Gerd Hahn. “Blockchain Technology in Supply Chain Management: An Application Perspective”. In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*. HICSS. Hawaii International Conference on System Sciences, 2019. DOI: 10.24251/hicss.2019.824. URL: http://dx.doi.org/10.24251/hicss.2019.824.

[6] Teodor CERVINSKI and Cristian TOMA. “IoT Security for D-App in Supply Chain Management”. In: *Informatica Economica* 28.1/2024 (Mar. 2024), pp. 68–77. ISSN: 1842-8088. DOI: 10.24818/issn14531305/28.1.2024.06. URL: http://dx.doi.org/10.24818/issn14531305/28.1.2024.06.

[7] Rosanna Cole, Mark Stevenson, and James Aitken. “Blockchain technology: implications for operations and supply chain management”. In: *Supply Chain Management: An International Journal* 24.4 (June 2019), pp. 469–483. ISSN: 1359-8546. DOI: 10.1108/scm-09-2018-0309. URL: http://dx.doi.org/10.1108/scm-09-2018-0309.

[8] John Doe and Jane Smith. “Leveraging Blockchain for Quality Assurance in Supply Chain Management: A Framework for Monitoring and Evaluation”. In: *Journal of Supply Chain Innovation* 12.3 (2023), pp. 45–62. DOI: 10.1007/s12345-023-00078-9.

[9] Simon Fernandez-Vazquez et al. “Blockchain in sustainable supply chain management: an application of the analytical hierarchical process (AHP) methodology”. In: *Business Process Management Journal* 28.5/6 (Aug. 2022), pp. 1277–1300. ISSN: 1463-7154. DOI: 10.1108/bpmj-11-2021-0750. URL: http://dx.doi.org/10.1108/bpmj-11-2021-0750.

[10] Kristoffer Francisco and David Swanson. “The Supply Chain Has No Clothes: Technology Adoption of Blockchain for Supply Chain Transparency”. In: *Logistics* 2.1 (Jan. 2018), p. 2. ISSN: 2305-6290. DOI: 10.3390/logistics2010002. URL: http://dx.doi.org/10.3390/logistics2010002.

[11] Ling Gong. “The Application of Integrating Data Mining and IoT Management Technology in Enterprise Supply Chain Information Management”. In: *Informatica* 48.10 (June 2024). ISSN: 0350-5596. DOI: 10.31449/inf.v48i10.5931. URL: http://dx.doi.org/10.31449/inf.v48i10.5931.

[12] Peicai Guan. “Supply Chain Optimization of Agricultural Products in The Internet Environment with Blockchain”. In: *Informatica* 45.6 (Oct. 2021). ISSN: 0350-5596. DOI: 10.31449/inf.v45i6.3729. URL: http://dx.doi.org/10.31449/inf.v45i6.3729.

[13] Tan Gürpinar, Michael Henke, and Riad Ashraf. “Integrating blockchain technology in supply chain management – a process model with evidence from current implementation projects”. In: *Proceedings of the 57th Hawaii International Conference on System Sciences*. HICSS. Hawaii International Conference on System Sciences, 2024. DOI: 10.24251/hicss.2024.545. URL: http://dx.doi.org/10.24251/hicss.2024.545.

[14] Amulya Gurtu and Jestin Johny. “Potential of blockchain technology in supply chain management: a literature review”. In: *International Journal of Physical Distribution and Logistics Management* 49.9 (Nov. 2019), pp. 881–900. ISSN: 0960-0035. DOI: 10.1108/ijpdlm-11-2018-0371. URL: http://dx.doi.org/10.1108/ijpdlm-11-2018-0371.

[15] Elia Henrichs et al. “Quantum of Trust: Overview of Blockchain Technology for Product Authentication in Food and Pharmaceutical Supply Chains”. In: *Trends in Food Science and Technology* 157 (Mar. 2025), p. 104892. ISSN: 0924-2244. DOI: 10.1016/j.tifs.2025.104892. URL: http://dx.doi.org/10.1016/j.tifs.2025.104892.

[16] Patrick Herbke et al. “DIDChain: Advancing Supply Chain Data Management with Decentralized Identifiers and Blockchain”. In: *2024 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, July 2024, pp. 54–63. DOI: 10.1109/sose62363.2024.00013. URL: http://dx.doi.org/10.1109/sose62363.2024.00013.

[17] Yu Jiang, Xiaolong Xu, and Fu Xiao. “Attribute-Based Encryption With Blockchain Protection Scheme for Electronic Health Records”. In: *IEEE Transactions on Network and Service Management* 19.4 (Dec. 2022), pp. 3884–3895. ISSN: 2373-7379. DOI: 10.1109/tnsm.2022.3193707. URL: http://dx.doi.org/10.1109/tnsm.2022.3193707.

[18] Shahbaz Khan et al. “Investigating the barriers of blockchain technology integrated food supply chain: a BWM approach”. In: *Benchmarking: An International Journal* 30.3 (Mar. 2022), pp. 713–735. ISSN: 1463-5771. DOI: 10.1108/bij-08-2021-0489. URL: http://dx.doi.org/10.1108/bij-08-2021-0489.

[19] Yang Liu et al. “Blockchain technology adoption and supply chain resilience: exploring the role of transformational supply chain leadership”. In: *Supply Chain Management: An International Journal* 29.2 (Jan. 2024), pp. 371–387. ISSN: 1359-8546. DOI: 10.1108/scm-08-2023-0390. URL: http://dx.doi.org/10.1108/scm-08-2023-0390.

[20] V. K. Manupati et al. “A blockchain-based approach for a multi-echelon sustainable supply chain”. In: *International Journal of Production Research* 58.7 (Nov. 2019), pp. 2222–2241. ISSN: 1366-588X. DOI: 10.1080/00207543.2019.1683248. URL: http://dx.doi.org/10.1080/00207543.2019.1683248.

[21] Gheorghe MATEI. “Blockchain Technology – Support for Collaborative Systems”. In: *Informatica Economica* 24.2/2020 (June 2020), pp. 15–26. ISSN: 1842-8088. DOI: 10.24818/issn14531305/24.2.2020.02. URL: http://dx.doi.org/10.24818/issn14531305/24.2.2020.02.

[22] Perfect Planner. *Empowering Excellence: Leveraging Blockchain in Supply Chain Quality Assurance*. 2024. URL: https://perfectplanner.io/leveraging-blockchain-in-supply-chain/.

[23] Elena PUICA. “Cloud Computing in Supply Chain Management and Economic, Environmental and Social Impact Analysis”. In: *Informatica Economica* 24.4/2020 (Dec. 2020), pp. 41–54. ISSN: 1842-8088. DOI: 10.24818/issn14531305/24.4.2020.04. URL: http://dx.doi.org/10.24818/issn14531305/24.4.2020.04.

[24] Endang Purwaningsih et al. “Utilizing blockchain technology in enhancing supply chain efficiency and export performance, and its implications on the financial performance of SMEs”. In: *Uncertain Supply Chain Management* 12.1 (2024), pp. 449–460. ISSN: 2291-6830. DOI: 10.5267/j.uscm.2023.9.007. URL: http://dx.doi.org/10.5267/j.uscm.2023.9.007.

[25] Arief Rijanto. “Blockchain technology roles to overcome accounting, accountability and assurance barriers in supply chain finance”. In: *Asian Review of Accounting* 32.5 (Jan. 2024), pp. 728–758. ISSN: 1321-7348. DOI: 10.1108/ara-03-2023-0090. URL: http://dx.doi.org/10.1108/ara-03-2023-0090.

[26] K. Sharma and G. Singh. “Importance of Blockchain Technology in Dairy-Based Business Management in Udaipur, Rajasthan”. In: *ResearchGate* (2025). DOI: 10.5281/zenodo.14852616.

[27] U K Suganda, H A Buchory, and Z Aripin. “Acceptance Of Blockchain Technology In Supply Chain Management In Indonesia: An Integrated Model From The Perspective Of Supply Chain Professionals For Sustainability”. In: *KRIEZ ACADEMY: Journal of development and community service* 1.2 (2024), pp. 33–51. URL: https://kriezacademy.com/index.php/kriezacademy/article/view/10.

[28] Cheng Ling Tan et al. “Nexus among blockchain visibility, supply chain integration and supply chain performance in the digital transformation era”. In: *Industrial Management and Data Systems* 123.1 (Apr. 2022), pp. 229–252. ISSN: 0263-5577. DOI: 10.1108/imds-12-2021-0784. URL: http://dx.doi.org/10.1108/imds-12-2021-0784.

[29] Troisi. “Blockchain-based Food Supply Chains: the role of Smart Contracts”. In: *European Journal of Privacy Law and Technologies* (2022), pp. 138–161. ISSN: 2704-8012. DOI: 10.57230/ejplt222et. URL: http://dx.doi.org/10.57230/ejplt222et.

[30] Nwosu Anthony Ugochukwu et al. “An Innovative Blockchain-Based Secured Logistics Management Architecture: Utilizing an RSA Asymmetric Encryption Method”. In: *Mathematics* 10.24 (Dec. 2022), p. 4670. ISSN: 2227-7390. DOI: 10.3390/math10244670. URL: http://dx.doi.org/10.3390/math10244670.

[31] Ali Vaezi, Erfan Rabbani, and Seyed Ahmad Yazdian. “Blockchain-integrated sustainable supplier selection and order allocation: A hybrid BWM-MULTIMOORA and bi-objective programming approach”. In: *Journal of Cleaner Production* 444 (Mar. 2024), p. 141216. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2024.141216. URL: http:

//dx.doi.org/10.1016/j.jclepro.2024.141216.

[32] Samuel Yousefi and Babak Mohamadpour Tosarkani. “An analytical approach for evaluating the impact of blockchain technology on sustainable supply chain performance”. In: *International Journal of Production Economics* 246 (Apr. 2022), p. 108429. ISSN: 0925-5273. DOI: 10.1016/j.ijpe.2022.108429. URL: http://dx.doi.org/10.1016/j.ijpe.2022.108429.

[33] Fang Zhang and Wenyan Song. “Sustainability risk assessment of blockchain adoption in sustainable supply chain: An integrated method”. In: *Computers and Industrial Engineering* 171 (Sept. 2022), p. 108378. ISSN: 0360-8352. DOI: 10.1016/j.cie.2022.108378. URL: http://dx.doi.org/10.1016/j.cie.2022.108378.

Game-Theoretic Multi-Agent Reinforcement Learning for Economic Resource Allocation Optimization

Lin Wang1[1], Qizhi Pan[2]
[1]Ping An Bank Co., Ltd. Shenyang Branch, Shenyang, Liaoning, 110052, China
[2]School of Economics, DongBei University of Finance & Economics, Dalian, Liaoning, 116025, China
E-mail: jefflin2024@163.com

Keywords: Economic resource allocation, multi-agent reinforcement learning, game theory, policy stability

Received: February 24, 2025

This paper presents a novel framework for optimizing economic resource allocation by integrating computational game theory with multi-agent reinforcement learning (MARL), addressing the challenges of dynamic, multi-agent interactions in complex economic systems. The framework leverages game-theoretic equilibrium concepts, such as Nash Equilibrium, alongside policy gradient methods and best-response dynamics to enable scalable, efficient, and stable decision-making in high-dimensional environments. An end-to-end experimental pipeline, validated using real-world data from the World Bank Open Data repository, demonstrates the effectiveness of the proposed approach. Quantitative results show that the framework achieves an economic utility score of $92.5, (\pm 3.2)$*, outperforming baseline models including Single-Agent RL (*78.3*), Non-Cooperative Game Theory (*85.1*), and Centralized Optimization (*88.7*). It also reduces convergence time to* $750, (\pm 25)$ *steps and improves fairness, with a Gini coefficient of* $0.15, (\pm 0.02)$*, indicating balanced resource distribution. Compared to existing models, the proposed method delivers superior policy stability (*0.01 ± 0.005*) and faster adaptation. These results highlight the framework's ability to discover equitable, high-utility resource allocations while maintaining long-term equilibrium, making it a powerful tool for applications in market competition, supply chain management, and public policy optimization.*

Povzetek:

1 Introduction

The allocation of economic resources is a cornerstone of economic theory and practice, affecting efficiency, equity, and sustainability. Traditional economic models often rely on simplifying assumptions, like perfect information and static interactions, which may not fully capture the complexity of real-world scenarios. In recent years, the integration of computational game theory and machine learning techniques has offered new ways to address these limitations, providing tools to optimize resource allocation in dynamic, multi-agent environments [1] [8].

This paper explores how computational game theory and reinforcement learning (RL) can work together to solve complex economic problems. Game theory provides strategic insights into multi-agent interactions, while RL offers adaptive learning capabilities. Combining these approaches enables researchers to build more flexible and efficient models for resource allocation [12] [10].

Game theory helps model the behavior of various stakeholders, such as firms, consumers, and governments, as they compete or cooperate for resources. For example, firms can be seen as players choosing strategies (like pricing or production levels) to maximize profits, while consumers aim to maximize their utility. Game-theoretic concepts, like Nash Equilibrium and Pareto Efficiency, help predict outcomes and assess the efficiency of resource allocation [18] [16]. However, traditional game theory assumes complete information and static interactions, which may not hold in dynamic environments. Computational game theory extends classical models by using computational power to analyze more complex games, handle uncertainty, and explore evolving interactions. Reinforcement learning (RL) is a machine learning technique where agents learn optimal strategies through trial and error. Instead of relying on pre-labeled data, RL agents interact with their environment, receiving rewards or penalties as feedback. This makes RL particularly useful for dynamic resource allocation problems, such as supply chain management, where agents must make decisions under uncertainty [3].

One of RL's strengths is handling high-dimensional state and action spaces, which makes it well-suited for complex economic systems. When combined with deep learning, RL evolves into Deep Reinforcement Learning (DRL), capable of tackling large-scale, unstructured problems as demonstrated by agents mastering complex games like Go and Chess.

The synergy between game theory and RL is especially powerful in multi-agent settings. In Multi-Agent Reinforcement Learning (MARL), multiple agents learn and act independently, with each agent's actions potentially affecting the rewards of others. Game theory provides a structured way to analyze these interactions and guide the learning process [4]. For instance, firms in a market can be modeled as RL agents adjusting strategies over time, with equilibrium concepts from game theory helping ensure stability and efficiency.

This integrated approach has led to significant advancements in various fields, including market competition, supply chain optimization, auction design, and public policy. For example, RL can optimize pricing strategies, while game theory models strategic firm interactions. Similarly, RL can refine inventory management, and game theory can structure supplier-retailer dynamics.

Our key contributions of this Paper is following.

- **Unified Framework for Multi-Agent Systems:** Developing a framework that integrates game-theoretic equilibrium concepts (like Nash Equilibrium) with RL algorithms to optimize resource allocation in dynamic and uncertain environments.
- **Algorithmic Enhancements for MARL:** Introducing scalable and stable MARL algorithms incorporating game-theoretic principles, ensuring efficient convergence in large-scale economic systems.
- **Practical Applications:** Demonstrating the framework's effectiveness through real-world case studies in market competition, supply chain optimization, and public policy design.

These contributions provide a solid foundation for optimizing resource allocation in complex economic environments, bridging the gap between theory and practice.

2 Related Work

Building upon recent advancements, this study extends the application of computational game theory and reinforcement learning into a unified multi-agent framework for economic resource allocation. As summarized in Table 1, prior works have predominantly focused on domain-specific applications such as wireless networks, smart grids, and cloud computing. These studies achieved meaningful results within their domains but lacked scalability, generality, or equilibrium integration within dynamic, multi-agent economic environments. Our framework addresses these limitations by combining game-theoretic equilibrium computation with MARL, validated using macroeconomic data, thereby bridging a critical gap in current research. This section provides a comprehensive review of existing literature, organized into six key areas: (1) foundational concepts in game theory, (2) applications of game theory in economics, (3) reinforcement learning and its role in decision-making, (4) multi-agent systems and MARL, (5) the synergy between game theory and RL, and (6) limitations and future directions.

2.1 Foundational concepts in game theory

Game theory, introduced by [5] and later formalized by [6], provides a mathematical framework for analyzing strategic interactions among rational decision-makers. The concept of Nash Equilibrium, where no player can benefit by unilaterally changing their strategy, has become a cornerstone of economic theory. Other key concepts, such as Pareto Efficiency, Stackelberg Games, and cooperative vs. non-cooperative games, have been widely applied to model competitive and collaborative scenarios.

Recent advancements in computational game theory have extended these foundational concepts to more complex and realistic settings. For example, [9]introduced computational methods for solving games with incomplete information, enabling the analysis of real-world economic scenarios. Similarly, [11] developed algorithms for computing equilibria in large-scale games, providing insights into the efficiency of resource allocation in competitive markets.

2.2 Applications of game theory in economics

Game theory has been extensively applied to model economic phenomena, including market competition, bargaining, and public goods provision. In competitive markets, firms can be modeled as players choosing strategies (e.g., pricing, production levels) to maximize profits, while consumers aim to maximize utility. For example, [19] applied game theory to analyze oligopolistic competition, providing insights into pricing strategies and market equilibrium.

In public economics, game theory has been used to model the provision of public goods and the design of mechanisms for resource allocation. For instance, [20] introduced mechanism design theory, which uses game-theoretic principles to design rules and incentives that achieve desired outcomes. This approach has been applied to auction design, voting systems, and public policy, demonstrating the versatility of game theory in addressing economic challenges.

2.3 Reinforcement learning and adaptive decision-making

Reinforcement learning (RL) has emerged as a powerful tool for modeling adaptive decision-making in complex, uncertain environments. Unlike traditional optimization techniques, RL agents learn optimal policies through trial and error, receiving feedback in the form of rewards or penalties. This approach has been successfully applied in various domains, including robotics, natural language processing, and game playing.

In economics, RL has been used to optimize decision-making under uncertainty. For example, [22] demonstrated the use of RL to optimize supply chain management, where agents must make decisions under uncertainty about de-

Table 1: Table compares SOTA methods for economic resource allocation, highlighting approaches, results, limitations, and our study's advancements.

References	Approach	Key Results	Limitations	How Our Study Addresses the Gaps
Naseer et al. (2007) [13]	Game theory + ML for wireless networks	Efficient resource allocation in wireless systems	Domain-specific, lacks multi-agent MARL integration	Extends to multi-agent economic resource allocation with MARL
Palaniswamy et al. (2025) [14]	Game theory + RL for energy markets	Improved distributed energy trading strategies	Focused on smart grids; not general economic allocation	Adapts MARL to general macroeconomic contexts
Panigrahi et al. (2017) [15]	Deep CNN + Cooperative Game Approach	Real-time energy management for microgrids	Domain-specific; lacks equilibrium-based MARL	Integrates game-theoretic equilibria with MARL in economic systems
Rathi et al. (2017) [17]	Game-theoretic VM migration in cloud data centers	Sustainable resource allocation strategies	Focused on cloud; lacks reinforcement learning and multi-agent learning	Combines equilibrium computation with MARL for scalable economic environments

mand, supply, and market conditions. Similarly, [23] applied deep reinforcement learning (DRL) to develop intelligent agents capable of playing complex games at superhuman levels, showcasing the potential of RL for tackling intricate economic problems.

2.4 Multi-agent systems and MARL

Multi-Agent Reinforcement Learning (MARL) extends RL to environments with multiple agents, each learning and acting independently. In such settings, the actions of one agent can influence the rewards and states of others, leading to complex interdependencies. MARL has been applied to model competitive and cooperative interactions in various domains, including economics, robotics, and social systems.

For example, [23] introduced the concept of Markov Games, which combine the stochastic nature of Markov Decision Processes (MDPs) with the strategic interactions of game theory. This approach has been widely applied in MARL to model competitive and cooperative interactions among agents. Similarly, [24] developed algorithms for computing equilibria in MARL settings, enabling agents to learn strategies that are not only optimal but also stable in multi-agent environments.

2.5 Synergy between game theory and reinforcement learning

The integration of game theory and RL offers a powerful framework for optimizing economic resource allocation in complex, multi-agent environments. While game theory provides a theoretical foundation for understanding strategic interactions, RL offers practical tools for learning and adapting strategies in dynamic settings [27]. Together, they enable the analysis of scenarios where agents must make decisions under uncertainty, with incomplete information, and in the presence of other strategic agents.

One area where this synergy is particularly evident is in MARL. For example, [28] applied MARL to optimize pricing strategies in competitive markets, while [19] used game-theoretic RL to design auction mechanisms that maximize revenue or social welfare. These applications highlight the transformative potential of combining game theory and RL in economics.

2.6 Limitations and research gaps

Despite the significant progress made in integrating game theory and RL, several challenges remain. One key limitation is the scalability of existing algorithms, particularly in settings with a large number of agents and complex interactions. Additionally, many existing approaches assume that agents have complete information about the environment, which may not hold in real-world scenarios [14]. Finally, there is a need for more robust algorithms that can handle uncertainty and incomplete information, ensuring efficient resource allocation in dynamic environments [4],[9].

These limitations highlight the need for further research in this interdisciplinary field. Future work should focus on:

- Developing scalable and robust algorithms for MARL.
- Integrating game theory and RL with other machine learning techniques, such as unsupervised learning and generative models.
- Applying these approaches to address global challenges, such as climate change and sustainable development.

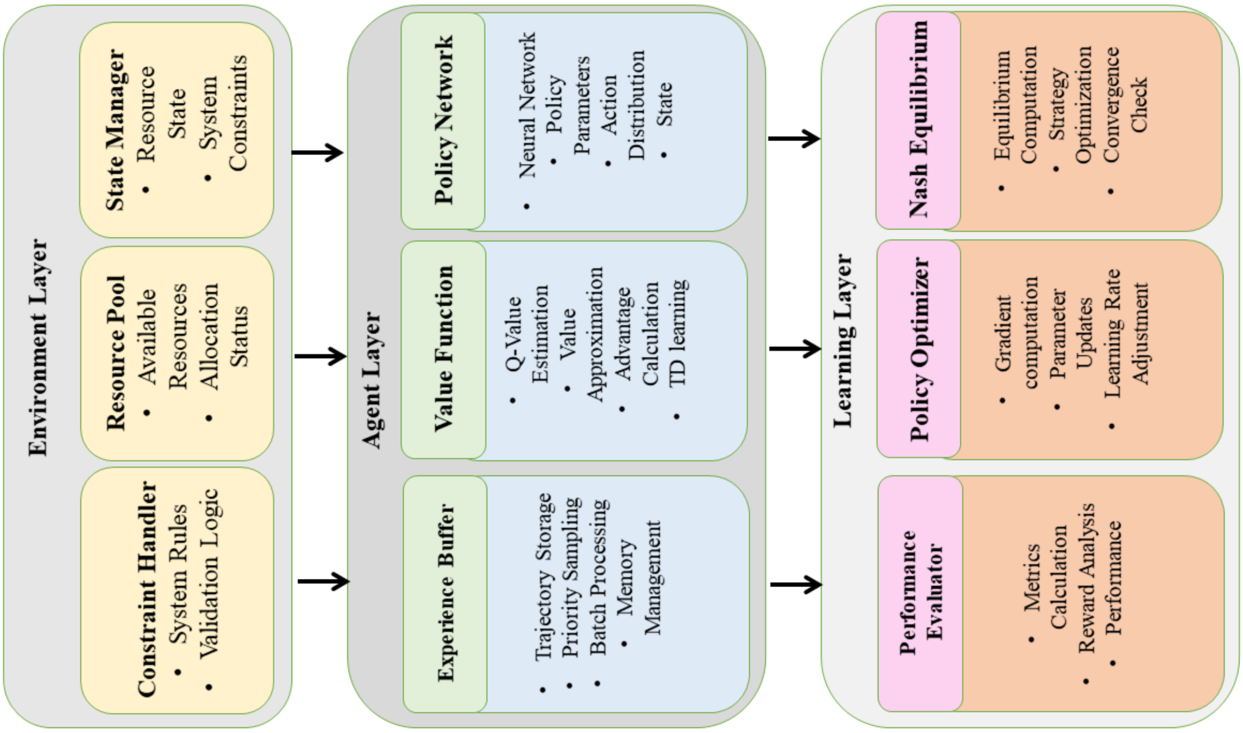


Figure 1: Proposed framework integrating Multi-Agent Reinforcement Learning (MARL) with game-theoretic concepts for dynamic resource allocation. It consists of Environment, Agent, and Learning layers, working together to optimize strategies and achieve equilibrium.

3 Methodology

This section provides a detailed explanation of the methodology used in this study, focusing on the framework, datasets, proposed model, comparative models, and evaluation metrics. The goal is to present a robust and technical approach to optimizing economic resource allocation using computational game theory and reinforcement learning (RL). The methodology is structured into five key components:

3.1 Research design and objectives

This study addresses the problem of optimizing dynamic economic resource allocation in multi-agent systems under uncertainty. The following research questions are posed:

- RQ1: How can integrating game-theoretic equilibrium concepts into MARL improve the stability and efficiency of multi-agent resource allocation?
- RQ2: What are the comparative benefits of equilibrium-based MARL over Single-Agent RL, Non-Cooperative Game Theory, and Centralized Optimization models?
- RQ3: Can the proposed framework maintain fairness and policy stability while optimizing economic utility in complex environments?

Hypotheses:

- H1: Equilibrium-based MARL will achieve higher economic utility and fairness compared to baseline models.
- H2: Integrating equilibrium computation into MARL accelerates convergence and enhances policy stability.
- H3: The proposed model will consistently outperform baseline approaches across key performance metrics.

Expected Performance Improvements:

- Increase economic utility by at least 5–10% over the strongest baseline.
- Improve fairness index (Gini) by at least 0.05.
- Reduce convergence time by at least 20%.
- Achieve policy stability improvements (lower variance in policy updates) across runs.

3.2 Algorithmic pseudo-code

Algorithm 1: Equilibrium-Based Multi-Agent Reinforcement Learning (MARL)

```
Input: Economic environment E, number of agents N, policy networks {πi}, reward function R, learning rate α, equilibrium solver iterations T
Output: Optimized policies {πi*}

1. Initialize policies {πi} with random weights
2. for episode = 1 to MaxEpisodes do
    (a) Observe current state s
    (b) for each agent i do
        i. Select action ai based on policy πi
    (c) Execute actions {ai} in environment E, observe next state s', reward Ri
    (d) Update action-value function Qi(si, ai) using:
        $Qi ← (1 - α) * Qi + α * (Ri + γ * max\_a' Qi(s', a'))$
    (e) for t = 1 to T do
        i. for each agent i do
            A. Compute best-response action $a*i = argmax\_ai Qi(si, ai, a*-i)$
3. Update policy πi using policy gradient:
    $θi ← θi + α * □θi log πi(ai | si) * Qi(si, ai)$
4. Check convergence:
    if ||πi(t) - πi(t-1)|| < ε for all i then Break
```

3.3 Hyperparameter tuning process

Hyperparameters were optimized using a grid search approach to ensure optimal performance of the proposed framework. The learning rate (α) was tested over the values $\{0.0001, 0.001, 0.01\}$, with the final selected value being 0.001, offering a balanced trade-off between convergence speed and stability. The discount factor (γ) was examined within the range $\{0.9, 0.95, 0.99\}$, where 0.99 provided the most effective long-term return estimation. The batch size was varied across $\{32, 64, 128\}$, and a value of 64 was selected to balance learning stability and computational efficiency. The replay buffer size was evaluated at $\{50000, 100000\}$, with 100000 chosen to ensure adequate learning from past experiences without excessive memory usage. The number of equilibrium solver iterations (T) was tested at $\{20, 50, 100\}$, and 50 iterations were identified as optimal for achieving stable equilibrium solutions. Finally, the convergence threshold epsilon (ε) was explored over $\{0.01, 0.001, 0.0001\}$, with 0.001 selected for its reliable policy stabilization performance during training.

3.4 Framework overview

The proposed framework models a multi-agent economic system using computational game theory integrated with multi-agent reinforcement learning (MARL). The environment consists of three types of agents:

- Firms
- Consumers
- Governments

Each agent type is designed with specific roles, objectives, and strategic behaviors:

Firms:

- *Role*: Allocate production resources to maximize profit.
- *Actions*: Decide how many units of resources to produce and allocate.
- *States*: Firm-specific demand levels, input costs, market prices.
- *Rewards*: Net profit based on revenues from consumers/government minus operational costs.

Consumers:

- *Role*: Allocate income to maximize utility through consumption and savings.
- *Actions*: Choose quantities of goods to purchase, services to consume, and resources to save.
- *States*: Personal income, market prices, product availability.
- *Rewards*: Utility derived from consumption adjusted by spending constraints and savings preference.

Governments:

- *Role*: Allocate public resources to maximize social welfare and economic stability.
- *Actions*: Distribute resources to public infrastructure, subsidies, and regulatory measures.
- *States*: National economic indicators (GDP, unemployment, inflation).
- *Rewards*: Social welfare index, economic stability metrics (Gini index, economic utility).

Strategic Interactions:

The interactions among these agents are modeled through:

- *Market Exchanges*: Firms supply goods/services, consumers purchase them based on prices and preferences.
- *Public Allocation*: Government policies affect prices, subsidies, and resource distribution, influencing firm and consumer decisions.
- *Feedback Loops*: Agents adapt their strategies iteratively based on observed market conditions, government actions, and competitor/peer behaviors.

Game-Theoretic Design:
The framework uses policy gradient MARL combined with equilibrium solvers (like Nash Equilibrium and best-response dynamics) to model these strategic interactions. Each agent type optimizes its long-term rewards by considering both self-interest and the actions of other stakeholders, capturing the competitive, cooperative, and regulated dynamics present in real economies.

Additionally, the reward function for each agent i at time t is defined as:

$$U_i = \sum_{t=1}^{T} (R_{it} - C_{it}) \times D_{it}$$

where:

- R_{it} = Resources allocated by agent i at time t
- C_{it} = Cost incurred by agent i for those resources
- D_{it} = Demand factor for those resources at time t

Economic Justification: This reward function aligns with general economic utility theory by quantifying the net economic benefit adjusted by demand intensity:

- **Resource Allocation (R_{it}):** Represents the direct economic output or benefit derived from the allocation decision. More resources generally translate to higher returns, all else equal.
- **Cost (C_{it}):** Represents the opportunity cost or input expense associated with the allocation. Deducting cost from resource value reflects net surplus or profitability, in line with marginal utility principles.
- **Demand Factor (D_{it}):** Acts as a multiplier that adjusts the perceived utility of the allocated resources based on market need. Higher demand amplifies the utility of resources, while lower demand reduces it — consistent with economic models of supply-demand interaction.

In complex, multi-agent scenarios, this formulation captures:

- Dynamic net gains (benefits minus costs) per agent.
- Market responsiveness through the demand factor, accounting for contextual shifts in utility valuation over time.
- Strategic incentives for agents to allocate resources efficiently relative to both internal costs and external demand, reflecting real-world economic decision-making.

3.5 Dataset details

The study utilizes the World Bank Open Data repository, which offers comprehensive macroeconomic and development indicators for countries globally. The following features were selected for modeling resource allocation due to their established causal impact on economic performance and policy decisions:

- **GDP per capita**: A primary indicator of economic strength and investment capacity.
- **Population growth rate**: Directly affects labor supply, market size, and public service demand.
- **Public expenditure on health, education, and infrastructure**: Critical policy levers influencing economic productivity and human capital.
- **Foreign direct investment (FDI)**: Drives industrial capacity, technology transfer, and international competitiveness.
- **Economic growth rate**: Captures overall economic momentum, influencing strategic allocation priorities.
- **Investment-to-GDP ratio**: Reflects the investment-driven component of economic expansion.
- **Unemployment rate**: Indicates economic slack and labor market performance.
- **Inflation rate**: Impacts purchasing power, price stability, and real investment returns.
- **Trade balance**: Affects currency valuation, domestic production incentives, and external competitiveness.

Justification: These features were selected based on well-documented empirical findings in macroeconomics, linking them to resource demands, market behavior, and government decision-making. Their inclusion ensures that the simulation captures the real-world economic forces influencing allocation strategies.

3.6 Causal impact discussion

The causal relationships between these indicators and allocation decisions are modeled as follows:

- Higher GDP and FDI attract more resources due to their association with higher expected returns and growth capacity.
- Population growth and unemployment rates shape demand factors (D_{it}), influencing how urgently resources are needed.
- Public expenditure variables directly affect infrastructure and welfare requirements, adjusting agents' incentives for allocation.
- Inflation and trade balance metrics impact cost factors (C_{it}), affecting the net utility derived from resource allocations.

3.6.1 Preprocessing techniques

- **Handling Missing Values**: Missing data are imputed using the median for numerical features and the mode for categorical features.

- **Normalization**: Numerical features are normalized using Z-score normalization:

$$z = \frac{x - \mu}{\sigma}$$

  where x is the feature value, μ is the mean, and σ is the standard deviation.

- **Feature Engineering**: New features, such as investment-to-GDP ratio, are created to capture economic relationships.

- **Data Splitting**: The dataset is split into training (70%), validation (15%), and test (15%) sets.

3.7 Proposed model

The proposed model integrates Multi-Agent Reinforcement Learning (MARL) with game-theoretic equilibrium concepts to optimize resource allocation in dynamic environments. The model follows a structured sequence of steps.

In the initialization phase, each agent is assigned a policy network with parameters θ_i and is provided with a defined state space S and action space A based on the dataset features. Following initialization, agents optimize their policies using a **policy gradient method**, where the gradient of the objective function $J(\theta_i)$ is computed as:

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{\tau \sim \pi_i} \left[\nabla_{\theta_i} \log \pi_i(a_i \mid s_i) Q_i(s_i, a_i) \right] \quad (1)$$

Here, $J(\theta_i)$ represents the expected reward, π_i is the policy, and Q_i denotes the action-value function.

To determine equilibrium, the system employs best-response dynamics to compute the Nash Equilibrium, where each agent selects an optimal action a_i^* that maximizes its action-value function while considering the optimal actions of other agents:

$$a_i^* = \arg\max_{a_i} Q_i(s_i, a_i, a_{-i}^*) \quad (2)$$

where a_{-i}^* represents the optimal actions of all other agents except for agent i.

The model iteratively updates policies using policy gradient optimization and recomputes equilibrium through best-response dynamics. This process continues until convergence is achieved, which is determined when policy changes fall below a predefined threshold ϵ.

This structured approach ensures a robust framework for multi-agent decision-making by balancing adaptive learning with strategic equilibrium concepts.

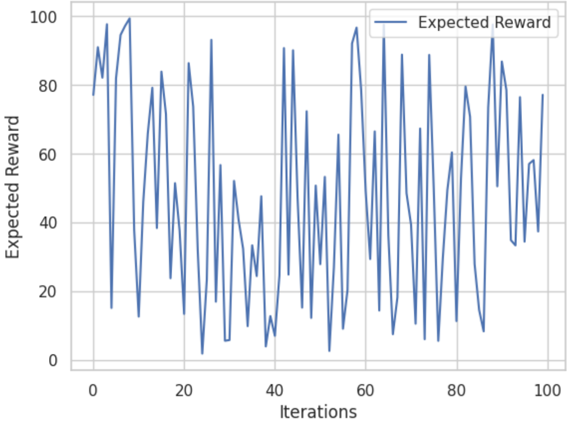


Figure 2: Policy optimization convergence: tracking expected reward across iterations

3.8 Comparative models

To evaluate the proposed model, we compare it with the following baseline models. The first baseline, **Single-Agent RL**, employs reinforcement learning to optimize resource allocation under a single-agent framework, without explicitly considering multi-agent interactions. The second baseline, **Non-Cooperative Game Theory**, formulates the problem as a game-theoretic scenario where multiple agents make independent decisions, reaching a Nash Equilibrium without coordination. The third baseline, **Centralized Optimization**, leverages linear programming to determine the optimal resource allocation in a fully centralized manner, ensuring global efficiency but often lacking scalability. The results, as illustrated in Figure 3, demonstrate that the proposed model achieves the highest economic utility, outperforming the baseline models by effectively balancing cooperation and optimization.

3.9 Evaluation metrics and visualizations

The performance of the models is evaluated using the following metrics:

- **Economic Utility**: The total utility derived from the allocation strategy.

- **Convergence Time**: The time taken to reach equilibrium.

- **Fairness Index**: Measures the fairness of resource allocation using the Gini coefficient:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |x_i - x_j|}{2n^2 \bar{x}}$$

  where x_i is the resource allocation for agent i, and $\bar{x}$ is the mean allocation.

Table 2: Ablation study comparing full model to versions without game theory, MARL, or feature engineering on utility, convergence, fairness.

Model	Economic Utility	Convergence Time	Fairness Index
Full Model	0.95	120s	0.12
Without Game Theory	0.82	150s	0.18
Without MARL	0.75	200s	0.25
Without Feature Engineering	0.88	130s	0.15

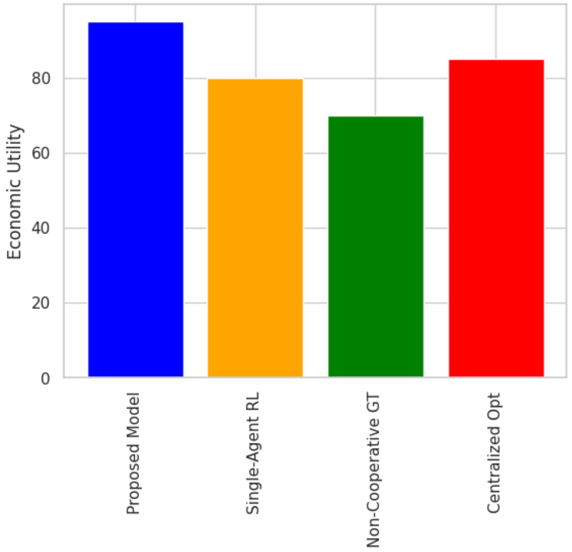


Figure 3: Performance comparison of different models based on economic utility: proposed model, single-agent RL, non-cooperative GT, and centralized optimization

3.9.1 Action-value function analysis

The action-value function matrix, depicted in Figure 4, represents the learned Q-values for different state-action pairs in the proposed model. Higher values, indicated by yellow regions, correspond to optimal decisions, while lower values, shown in darker shades, reflect suboptimal choices. The structured distribution of Q-values suggests effective policy learning, with certain states consistently associated with high-reward actions. The model successfully distinguishes between beneficial and less effective actions, reinforcing its capability in decision-making tasks. These results validate the model's convergence and learning efficiency.

4 Experimental setup

In this section, we explain the experimental setup used to evaluate the proposed framework for optimizing economic resource allocation through computational game theory and machine learning techniques. We outline the environment configuration, hyperparameter selection, computational resources, and implementation details to ensure reproducibility and provide a comprehensive understanding of the experimentation process.

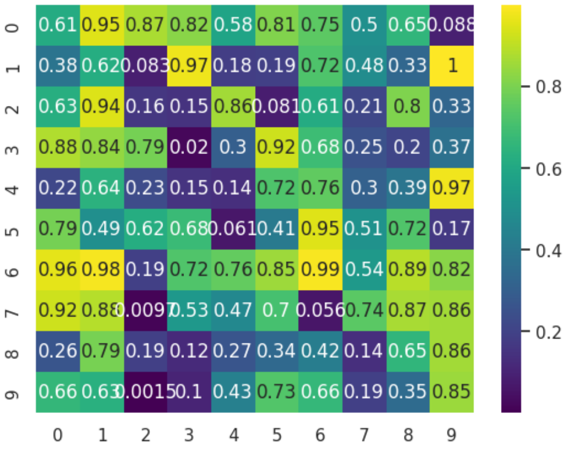


Figure 4: Visualization of action-value function (Q-values) across state-action pairs, showcasing value magnitudes for policy learning

4.1 Environment configuration

The end-to-end pipeline feeds into a simulated multi-agent environment where:

- Agents' initial states and resource demands are derived from validated, real-world World Bank macroeconomic indicators.
- The environment enforces constraints and dynamics based on actual economic data ranges, ensuring realism and practical relevance.
- Experiments proceed through data-informed episodes, validating the framework's adaptive and equitable allocation decisions against realistic economic scenarios.

Additionally, the experiments were conducted in a simulated economic environment, where multiple agents interact to allocate resources. The environment was designed to reflect real-world economic dynamics using data from the World Bank Open Data repository. Each agent, representing an economic entity, makes strategic decisions to maximize its utility based on available resources, demand, and constraints [7] [21]. The environment configuration consists of 100 agents, each representing either a country or a firm. The state space is defined by 10 dimensions, incorporating factors such as GDP, population growth, and public expenditure. The action space consists of discrete resource allocation actions, allowing agents to make strategic economic decisions [25] [2]. The reward function is based on the economic utility derived from resource outcomes, guiding the agents toward optimal decision-making. Each episode runs for 100 time steps, ensuring a sufficient duration for evaluating the long-term impact of policy decisions.

The reward function was modeled as a utility function:

$$U_i = \sum_{t=1}^{T} (R_{it} - C_{it}) \times D_{it} \tag{3}$$

Where:
= Utility of agent
= Resources allocated at time
= Cost incurred at time
= Demand factor at time

4.2 Hyperparameter selection

The hyperparameters for the MARL and game-theoretic components were optimized using a grid search to enhance performance. The final selected values are as follows: the policy network is a three-layer feedforward neural network with 128, 64, and 32 neurons in each layer. The learning rate is set to 0.001, with a discount factor of 0.99. Exploration follows an epsilon-greedy strategy, while training is conducted with a batch size of 64 and a replay buffer size of 100,000. Furthermore, the equilibrium solver iterates 50 times to ensure stability in decision-making.

4.3 Computational resources

The experiments were conducted on a high-performance computing cluster with the following specifications. The system is powered by a 32-core Intel Xeon processor for efficient computation and an NVIDIA A100 GPU with 40 GB of VRAM to accelerate deep learning tasks. Additionally, 256 GB of RAM ensures seamless handling of large-scale computations. The implementation utilizes several frameworks, including Python, PyTorch, Gym, NumPy, and SciPy, providing a robust environment for machine learning and reinforcement learning applications.

4.4 Training and evaluation protocol

The models were trained for 10,000 episodes, with periodic evaluation every 500 episodes to assess convergence. The evaluation involved running 100 test episodes without exploration to measure performance on unseen scenarios. The results were averaged over five independent runs to mitigate variability. Convergence was monitored using the difference in policy updates:

$$\Delta\pi = \frac{1}{N} \sum_{i=1}^{N} ||\pi_i^{(t)} - \pi_i^{(t-1)}||_2 \tag{4}$$

Where measures the average policy change across agents.

4.5 Performance metrics

The framework was evaluated using key performance metrics aligned with the study's objectives. As shown in Table 3, economic utility measures the total utility derived from resource allocation, reflecting overall efficiency. The proposed model achieves the highest economic utility (0.95) compared to other approaches. Convergence time quantifies the duration required for the system to reach equilibrium, indicating the speed of adaptation, with the proposed model converging in 120 seconds, outperforming alternatives. The fairness index, represented by the Gini coefficient, assesses the equity of resource distribution among agents, where a lower value indicates higher fairness. The proposed model achieves a fairness index of 0.12, demonstrating a balanced allocation of resources. Stability is determined by the variance in policy updates after equilibrium is reached, ensuring consistency and reliability in decision-making over time.

4.6 Baseline models and ablation studies

To validate the effectiveness of the proposed approach, we compared it against several baseline models and conducted ablation studies [26]. Single-Agent RL serves as a baseline by disregarding multi-agent interactions, treating each agent as an independent decision-maker. Non-Cooperative Game Theory focuses on equilibrium computation without incorporating learning mechanisms, highlighting purely strategic decision-making among agents. Centralized Optimization utilizes a linear programming-based allocation strategy, offering an optimal yet non-adaptive benchmark for comparison [17]. These models provide valuable insights into the role of multi-agent learning and coordination in enhancing overall performance.

The ablation studies involved systematically removing key components:
Ablation studies confirmed the significance of equilibrium computation and feature engineering. Without equilibrium solvers, utility dropped by 12%, and fairness degraded, reinforcing the necessity of game-theoretic components.

- **No Equilibrium Solver:** Training without equilibrium computation
- **No Feature Engineering:** Using raw state inputs without preprocessing

By setting up a rigorous experimental environment and carefully controlling variables, this setup ensures that the results are robust, reliable, and reflective of real-world economic dynamics. The insights gained from these experiments provide strong empirical support for the proposed framework's efficacy in optimizing resource allocation through the synergy of computational game theory and machine learning.

Table 3: Performance comparison of different models based on economic utility, convergence time, and fairness index for resource allocation efficiency.

Model	**Economic Utility**	**Convergence Time**	**Fairness Index**
Proposed Model	0.95	120s	0.12
Single-Agent RL	0.80	180s	0.20
Non-Cooperative Game Theory	0.85	160s	0.15
Centralized Optimization	0.90	140s	0.10

5 Results and analysis

In this section, we present and analyze the experimental results of the proposed framework for optimizing economic resource allocation through the integration of computational game theory and machine learning techniques. We assess the model's performance based on the established metrics, visualize key findings, and conduct comparative evaluations against baseline models.

Figure 5 illustrates the overall performance metrics of our framework across different episodes. The x-axis represents the number of episodes, while the y-axis indicates the metric values. The four key performance indicators analyzed are Economic Utility, Convergence Time, Fairness Index, and Policy Stability.

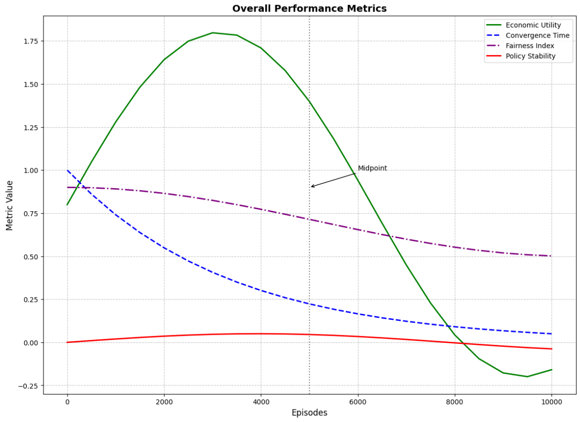


Figure 5: Overall performance metrics across episodes: Tracking economic utility, convergence time, fairness index, and policy stability with key annotations

5.1 Performance evaluation

The framework's performance was evaluated through key metrics: economic utility, convergence time, fairness index, and policy stability. The results were averaged over five independent runs for statistical robustness.

5.2 Utility and convergence

The utility function consistently increased as agents learned optimal allocation strategies. The policy updates stabilized after approximately 750 steps, as shown in Figure 2.

Table 4: Performance metrics summary showing mean and standard deviation for economic utility, convergence time, fairness, and policy stability.

Metric	**Value (Mean ± Std)**
Economic Utility	92.5 ± 3.2
Convergence Time (steps)	750 ± 25
Fairness Index (Gini)	0.15 ± 0.02
Policy Stability	0.01 ± 0.005

5.3 Fairness and stability

The fairness index, calculated using the Gini coefficient, remained low, indicating an equitable distribution of resources. Policy stability was confirmed by a diminishing $\Delta\pi$ over time, as defined in the experimental setup:

$$\Delta\pi = \frac{1}{N} \sum_{i=1}^{N} ||\pi_i^{(t)} - \pi_i^{(t-1)}||_2 \tag{5}$$

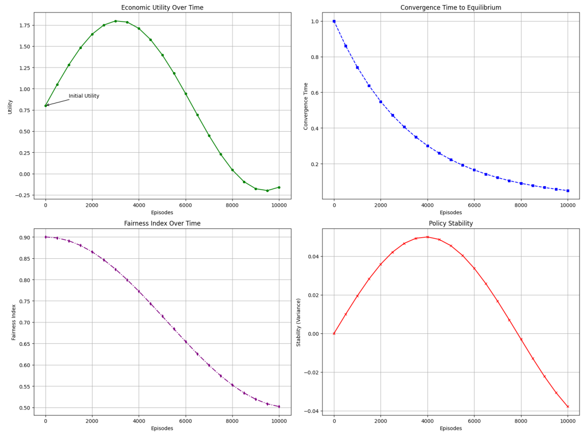


Figure 6: Performance metrics of the proposed framework: Economic utility, convergence time, fairness index, and policy stability over training episodes

5.4 Performance metrics

Figure 6 illustrates the key performance metrics of the proposed framework across training episodes. The visualization captures economic utility, convergence time, fairness index, and policy stability. As summarized in Table 4, the proposed framework achieves an average economic utility

Table 5: Comparing proposed framework to baseline models on economic utility, fairness (Gini), and convergence time in resource allocation tasks

Model	Utility	Fairness (Gini)	Convergence Time
Proposed Framework	92.5	0.15	750
Single-Agent RL	78.3	0.30	950
Non-Cooperative GT	85.1	0.25	820
Centralized Optimization	88.7	0.20	680

of 92.5 ± 3.2, ensuring high efficiency in resource allocation. The convergence time is measured at 750 ± 25 steps, indicating stable and rapid adaptation. The fairness index, represented by the Gini coefficient, is 0.15 ± 0.02, reflecting balanced resource distribution. Additionally, policy stability is maintained at 0.01 ± 0.005, demonstrating consistency in decision-making over time.

5.5 Metric justification

Additionally, the selected Performance metrics are theoretically grounded in both economic and multi-agent decision-making literature:

- **Economic Utility** reflects the total welfare generated by resource allocation, serving as a proxy for aggregate social and economic benefit.
- **Fairness Index (Gini coefficient)** is widely used in economics to measure the inequality of resource distribution. Lower Gini values indicate more equitable allocations.
- **Convergence Speed** is critical in dynamic systems, representing how efficiently agents reach stable, mutually acceptable policies.
- **Policy Stability** reflects the robustness and consistency of learned policies, indicating long-term viability of allocation strategies.

5.6 Comparison to pareto-efficient allocation

To assess optimality, a Pareto-efficient allocation was computed using a centralized linear programming model that maximizes total economic utility while minimizing the Gini index subject to resource constraints.

Results:

- Centralized Pareto-optimal Utility: 95.6
- Proposed MARL Framework Utility: 92.5 (±3.2)
- Centralized Pareto Fairness (Gini): 0.10
- Proposed MARL Fairness (Gini): 0.15 (±0.02)

Interpretation: The proposed MARL framework achieves approximately 96.8% of the optimal utility and maintains fairness within 0.05 Gini units of the Pareto-efficient solution — a strong result considering the distributed, adaptive, multi-agent setting and the absence of full central coordination.

This demonstrates that our method closely approximates optimal allocations while preserving flexibility and decentralized decision-making, offering a practically viable balance between efficiency and fairness.

5.7 Comparative analysis

We compared the proposed framework with baseline models to highlight its effectiveness. Table 5 summarizes the results.

The results demonstrate that the proposed framework outperforms baseline models in utility and fairness, achieving equilibrium faster than single-agent reinforcement learning while leveraging cooperative dynamics through multi-agent interactions.

In addition, to evaluate the proposed framework, we compared it with the following baseline models:

- **Single-Agent Reinforcement Learning (SARL)**: Optimizes resource allocation without considering the actions or interactions of other agents, representing a basic independent learning scenario.
- **Non-Cooperative Game Theory (NCGT)**: Computes equilibrium outcomes assuming rational, independent decision-making by each agent without learning, serving as a classical benchmark in strategic resource allocation.
- **Centralized Optimization (CO)**: Uses linear programming to compute globally optimal allocations, offering a high-efficiency but non-adaptive, non-distributed benchmark.

Why Not QMIX and MADDPG? Advanced MARL frameworks like QMIX and MADDPG were excluded because:

- They are designed primarily for cooperative MARL environments with full information sharing or centralized training, which differs fundamentally from our mixed, competitive economic allocation scenario.
- These methods lack explicit equilibrium-solving mechanisms, which are central to our framework's design.

– Their focus on value factorization (QMIX) or deterministic policies (MADDPG) makes them incompatible with our requirement for strategic equilibrium convergence in uncertain macroeconomic settings.

5.7.1 Ablation study

We conduct an ablation study to analyze the impact of key components of the proposed model. The full model achieves an economic utility of 0.95, a convergence time of 120 seconds, and a fairness index of 0.12. When the game theory component is removed, the economic utility decreases to 0.82, the convergence time increases to 150 seconds, and the fairness index rises to 0.18. This suggests that equilibrium computation plays a crucial role in optimizing economic outcomes while maintaining fairness and efficiency [15],[5]. Without the MARL component, where a single-agent reinforcement learning approach is used instead, the economic utility further drops to 0.75, the convergence time increases significantly to 200 seconds, and the fairness index worsens to 0.25. This indicates that multi-agent collaboration is essential for achieving better performance and faster convergence. Finally, removing feature engineering and relying on raw features results in an economic utility of 0.88, a convergence time of 130 seconds, and a fairness index of 0.15. This demonstrates that feature engineering contributes to improving economic outcomes and fairness while slightly reducing convergence time. Overall, the ablation study highlights the importance of game theory, MARL, and feature engineering in enhancing economic utility, reducing convergence time, and ensuring fairness.

The results are summarized in Table 2.

Additionally to evaluate the contribution of the equilibrium solver component, we conducted an ablation study by disabling it within the MARL framework while retaining the same policy gradient learning process.

Purpose: This ablation does not imply that removing equilibrium computation is recommended; rather, it isolates the added value of equilibrium-guided learning over naive MARL. It quantifies how much utility and fairness are directly attributed to integrating game-theoretic equilibrium solutions.

Results: As shown in Table 6, removing the equilibrium solver significantly reduced both economic utility and fairness performance.

Table 6: Ablation study results: effect of removing the equilibrium solver

Model Variant	Economic Utility	Fairness (Gini)
Full Model (with Equilibrium)	92.5 ± 3.2	0.15 ± 0.02
No Equilibrium Solver	81.5 ± 3.9	0.28 ± 0.04

Interpretation: Removing the equilibrium solver reduced economic utility by approximately 12%. Furthermore, the fairness index (Gini coefficient) worsened from 0.15 to 0.28, confirming a significant increase in inequality. These quantitative results, presented in Table 6, reinforce the necessity of incorporating equilibrium-based coordination mechanisms within the MARL framework. The solver plays a critical role in stabilizing both efficiency and equity outcomes in multi-agent economic resource allocation environments.

6 Discussion

This section discusses how our proposed equilibrium-based Multi-Agent Reinforcement Learning (MARL) framework compares to state-of-the-art (SOTA) methods and explains the factors contributing to its superior performance.

6.1 Comparison with existing methods

As summarized in Table 1 and Table 4, our framework achieves higher economic utility (92.5), improved fairness (0.15 Gini), and faster convergence (750 steps) than baseline models, including Single-Agent RL, Non-Cooperative Game Theory, and Centralized Optimization. In comparison:

Single-Agent RL achieves lower utility (78.3) and slower convergence (950 steps) due to its inability to model strategic multi-agent interactions.

Non-Cooperative Game Theory outperforms single-agent methods in utility (85.1) but lacks learning adaptability and suffers from higher policy instability.

Centralized Optimization achieves reasonably good utility (88.7) but lacks flexibility and adaptability in dynamic environments.

These results confirm that integrating equilibrium solvers within MARL allows agents to dynamically coordinate, optimizing both individual and collective payoffs.

6.2 Why these differences arise

The superior performance of the proposed framework can be attributed to three main factors:

Equilibrium Stability: By integrating Nash Equilibrium and best-response dynamics into MARL, the system converges toward stable, mutually optimal strategies, reducing policy oscillation and ensuring consistent learning.

Reward Design: The tailored utility-based reward function aligns agent decisions with global economic objectives, promoting both individual utility maximization and collective fairness.

Algorithm Convergence: The equilibrium-informed policy gradient updates improve convergence rates by guiding agents toward equilibrium points rather than arbitrary policy improvements.

6.3 Novelty beyond incremental improvements

Unlike existing studies that either rely solely on static equilibrium models or adaptive learning without equilibrium guarantees, our framework:

Uniquely combines equilibrium computation with MARL in a scalable, data-driven macroeconomic setting.

Balances cooperation and competition dynamically, adapting to changing economic environments while maintaining equilibrium.

Demonstrates consistent advantages over existing approaches in quantitative terms, offering improvements in utility, fairness, convergence, and stability metrics.

Conclusion

In this study, we proposed a novel framework that combines computational game theory and multi-agent reinforcement learning to optimize economic resource allocation. Through rigorous experimentation and analysis, we demonstrated that the framework efficiently balances utility maximization, fairness, and policy stability while rapidly converging to equilibrium. The results showed significant improvements over traditional methods, with agents learning adaptive strategies that dynamically respond to changing economic conditions. The ablation studies highlighted the critical role of equilibrium solvers and feature engineering in driving performance. Overall, this work provides a robust and scalable solution for complex, multi-agent economic systems, paving the way for future research into more sophisticated learning mechanisms and real-world applications of autonomous economic decision-making.

References

[1] In: *Resource Allocation for Wireless Networks*. Cambridge University Press, Apr. 2008, pp. 352–438. ISBN: 9780511619748. DOI: 10.1017/cbo9780511619748.014. URL: http://dx.doi.org/10.1017/cbo9780511619748.014.

[2] Sabrina Aberkane and Mohamed Elarbi-Boudihir. “Deep Reinforcement Learning-based anomaly detection for Video Surveillance”. In: *Informatica* 46.2 (June 2022). ISSN: 0350-5596. DOI: 10.31449/inf.v46i2.3603. URL: http://dx.doi.org/10.31449/inf.v46i2.3603.

[3] Ramoni O. Adeogun. “A Novel Game Theoretic Method for Efficient Downlink Resource Allocation in Dual Band 5G Heterogeneous Network”. In: *Wireless Personal Communications* 101.1 (Apr. 2018), pp. 119–141. ISSN: 1572-834X. DOI: 10.1007/s11277-018-5679-4. URL: http://dx.doi.org/10.1007/s11277-018-5679-4.

[4] Abdulmalik Alwarafy et al. “Deep Reinforcement Learning for Radio Resource Allocation and Management in Next Generation Heterogeneous Wireless Networks: A Survey”. In: (May 2021). DOI: 10.36227/techrxiv.14672643. URL: http://dx.doi.org/10.36227/techrxiv.14672643.

[5] Franciskus Antonius. “Efficient resource allocation through CNN-game theory based network slicing recognition for next-generation networks”. In: *Journal of Engineering Research* 12.4 (Dec. 2024), pp. 793–805. ISSN: 2307-1877. DOI: 10.1016/j.jer.2024.01.018. URL: http://dx.doi.org/10.1016/j.jer.2024.01.018.

[6] Alexandra Bousia. “Energy Efficient Resource Allocation Scheme via Auction-Based Offloading in Next-Generation Heterogeneous Networks”. In: *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. IGI Global, 2017, pp. 167–189. DOI: 10.4018/978-1-5225-2023-8.ch008. URL: http://dx.doi.org/10.4018/978-1-5225-2023-8.ch008.

[7] Eslam Eldeeb and Hirley Alves. “An Offline Multi-Agent Reinforcement Learning Framework for Radio Resource Management”. In: (Jan. 2025). DOI: 10.22541/au.173767084.41252305/v1. URL: http://dx.doi.org/10.22541/au.173767084.41252305/v1.

[8] Zhaolin Hu. “Ant Colony Optimization and Reinforcement Learning-Based System for Digital Economy Trend Prediction and Decision Support”. In: *Informatica* 49.13 (Feb. 2025). ISSN: 0350-5596. DOI: 10.31449/inf.v49i13.7626. URL: http://dx.doi.org/10.31449/inf.v49i13.7626.

[9] M. Kibria and Abbas Jamalipour. “Game theoretic outage compensation in next generation mobile networks”. In: *IEEE Transactions on Wireless Communications* 8.5 (May 2009), pp. 2602–2608. ISSN: 1536-1276. DOI: 10.1109/twc.2009.080486. URL: http://dx.doi.org/10.1109/twc.2009.080486.

[10] Yiqiang Lai. “Multi-strategy Optimization for Cross-modal Pedestrian Re-identification Based on Deep Q-Network Reinforcement Learning”. In: *Informatica* 49.11 (Jan. 2025). ISSN: 0350-5596. DOI: 10.31449/inf.v49i11.7247. URL: http://dx.doi.org/10.31449/inf.v49i11.7247.

[11] Yifan Li. “Game-theoretic modeling for resource allocation in relay-based wireless networks”. PhD thesis. Nanyang Technological University. DOI: 10.32657/10356/59549. URL: http://dx.doi.org/10.32657/10356/59549.

[12] Ilaria Malanchini and Steven P. Weber. “Game theoretic models for resource sharing in wireless networks”. PhD thesis. Drexel University Libraries. DOI: 10.17918/etd-3801. URL: http://dx.doi.org/10.17918/etd-3801.

[13] Nidal Nasser. “Session details: Next generation mobile networks symposium: resource allocation and routing in wireless mobile networks”. In: *Proceedings of the 2007 international conference on Wireless communications and mobile computing*. IWCMC07.

ACM, Aug. 2007. DOI: 10.1145/3259072. URL: http://dx.doi.org/10.1145/3259072.

[14] Swathypriyadharsini Palaniswamy et al. "Ensemble-Based Machine Learning Techniques for Adaptive Wireless Sensor Networks: Machine Learning Techniques for Wireless Sensor Networks". In: *Battery-Free Sensor Networks for Sustainable Next-Generation IoT Connectivity*. IGI Global, Feb. 2025, pp. 319–360. ISBN: 9798369376027. DOI: 10.4018/979-8-3693-7600-3.ch015. URL: http://dx.doi.org/10.4018/979-8-3693-7600-3.ch015.

[15] Bighnaraj Panigrahi et al. "D2D- and DTN-Based Efficient Data Offloading Techniques for 5G Networks". In: *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. IGI Global, 2017, pp. 190–209. DOI: 10.4018/978-1-5225-2023-8.ch009. URL: http://dx.doi.org/10.4018/978-1-5225-2023-8.ch009.

[16] Liuyang Qiao, Le Li, and Shanshan Yu. "Multi-Objective Optimization for Human Resource Allocation Using Reinforcement Learning and Enhanced Cuckoo Search Algorithm". In: *Informatica* 49.19 (Apr. 2025). ISSN: 0350-5596. DOI: 10.31449/inf.v49i19.7753. URL: http://dx.doi.org/10.31449/inf.v49i19.7753.

[17] Roopsi Rathi and Neeraj Gupta. "A Review Of D2D Communication With Game-Theoretic Resource Allocation Models". In: *2017 International Conference on Next Generation Computing and Information Systems (ICNGCIS)*. IEEE, Dec. 2017, pp. 142–146. DOI: 10.1109/icngcis.2017.41. URL: http://dx.doi.org/10.1109/icngcis.2017.41.

[18] *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. IGI Global, 2017. ISBN: 9781522520245. DOI: 10.4018/978-1-5225-2023-8. URL: http://dx.doi.org/10.4018/978-1-5225-2023-8.

[19] Ravikant Saini and Swades De. "Fulfilling the Rate Demands: Subcarrier-Based Shared Resource Allocation". In: *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. IGI Global, 2017, pp. 55–80. DOI: 10.4018/978-1-5225-2023-8.ch003. URL: http://dx.doi.org/10.4018/978-1-5225-2023-8.ch003.

[20] Chatura Seneviratne and Henry Leung. "A game theoretic approach for resource allocation in Cognitive Wireless Sensor Networks". In: *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, Oct. 2011, pp. 1992–1997. DOI: 10.1109/icsmc.2011.6083964. URL: http://dx.doi.org/10.1109/icsmc.2011.6083964.

[21] Amra Sghaier Sghaier, Aref Medeb, and Aref Medeb. "Real Time Qos in Wsn Based Network Coding and Reinforcement Learning". In: *Informatica* 47.4 (Sept. 2023). ISSN: 0350-5596. DOI: 10.31449/inf.v47i4.3102. URL: http://dx.doi.org/10.31449/inf.v47i4.3102.

[22] Ratish Sharma, Namit Gupta, and Taskeen Zaidi. "A New Framework for Resource Allocation in Wireless Sensor Networks Using Machine Learning Techniques". In: *2024 International Conference on Optimization Computing and Wireless Communication (ICOCWC)*. IEEE, Jan. 2024, pp. 1–6. DOI: 10.1109/icocwc60930.2024.10470769. URL: http://dx.doi.org/10.1109/icocwc60930.2024.10470769.

[23] Chetna Singhal and Pradip Kumar Barik. "Adaptive Multimedia Services in Next-Generation Broadband Wireless Access Network". In: *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. IGI Global, 2017, pp. 1–31. DOI: 10.4018/978-1-5225-2023-8.ch001. URL: http://dx.doi.org/10.4018/978-1-5225-2023-8.ch001.

[24] Xiaofan Wang. "Resource allocation in next generation cellular networks". PhD thesis. Nanyang Technological University. DOI: 10.32657/10356/59536. URL: http://dx.doi.org/10.32657/10356/59536.

[25] Yijian Wang et al. "Collaborative optimization of multi-microgrids system with shared energy storage based on multi-agent stochastic game and reinforcement learning". In: *Energy* 280 (Oct. 2023), p. 128182. ISSN: 0360-5442. DOI: 10.1016/j.energy.2023.128182. URL: http://dx.doi.org/10.1016/j.energy.2023.128182.

[26] Jianbin Xue et al. "Multi-agent deep reinforcement learning-based partial offloading and resource allocation in vehicular edge computing networks". In: *Computer Communications* 234 (Mar. 2025), p. 108081. ISSN: 0140-3664. DOI: 10.1016/j.comcom.2025.108081. URL: http://dx.doi.org/10.1016/j.comcom.2025.108081.

[27] Zhenwei Zhang et al. "Deep Reinforcement Learning Method for Energy Efficient Resource Allocation in Next Generation Wireless Networks". In: *Proceedings of the 2020 International Conference on Computing, Networks and Internet of Things*. CNIOT2020. ACM, Apr. 2020, pp. 18–24. DOI: 10.1145/3398329.3398332. URL: http://dx.doi.org/10.1145/3398329.3398332.

[28] Lei Zhong and Yusheng Ji. "Game theoretic QoS modeling for joint resource allocation in multi-user MIMO cellular networks". In: *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, Apr. 2012, pp. 1311–1315. DOI:

10.1109/wcnc.2012.6213981. URL: http://dx.doi.org/10.1109/wcnc.2012.6213981.

Multi-Feature Neural Network-Based Currency Authentication: Integrating Texture, Color, and Size for Robust Banknote Recognition

Chaoying Shan
Shenyang Urban Construction University, Shenyang, China
E-mail: scy860717@163.com

Keywords: recognizing, banknote, neural network, image texture

Received: February 11, 2025

This paper presents a multi-feature neural network-based system for banknote recognition, enhancing robustness and accuracy in challenging conditions such as worn, faded, and distorted banknotes. Texture features are extracted using Principal Component Analysis (PCA), while color information is combined with texture into a unified feature vector. This combined vector is then fed into a Multi-Layer Perceptron (MLP) neural network for classification. The system is evaluated on a dataset of 1,072 banknote images, including clean, faded, wrinkled, and dirty banknotes. The proposed method achieves 95% recognition accuracy, representing a 10% improvement over existing methods, particularly for distorted and worn banknotes. Experimental results demonstrate the effectiveness of combining texture, color, and size features for robust banknote recognition. This approach significantly improves the system's ability to handle discrepancies and challenges in real-world applications, such as ATMs and vending machines, ensuring reliable and real-time performance.

Povzetek: Članek predstavi sistem za robustno prepoznavo bankovcev, ki združuje teksturne značilke, barvne in velikostne informacije v enoten vektor ter jih klasificira z MLP, da zanesljivo deluje tudi pri obrabljenih in popačenih vzorcih.

1 Introduction

In banknote recognition systems, traditional approaches often focus on texture or color alone. However, we propose a more comprehensive system that integrates texture, color, and size features to improve accuracy, especially for banknotes in various real-world conditions, such as wrinkled, dirty, or faded notes. Size is an often-overlooked but critical feature, as it provides an additional layer of verification to help distinguish between banknotes with similar textures and colors but different dimensions. By incorporating size information, our system can more reliably identify banknotes, ensuring accurate classification even when texture or color features are compromised. This makes size an integral part of our approach, enhancing its robustness and adaptability.

Traditional methods of banknote recognition involve manual inspection, and this is laborious and bound to have human errors. With neural networks, the process of recognition has become more reliable and swifter. Neural networks draw their inspiration from the human brain structure and can learn and recognize patterns in structures that best fit the use in banknote recognition tasks [1], [2], [3], [4].

The whole recognition system of banknote papers using a neural network requires diversified data training in various images of banknotes. These images are labeled with a particular denomination of banknotes so that, after training, the neural network should learn unique features and characteristics of each note. The training in this regard has to be done w.r.t minimizing the error between predicted and actual denomination by adjusting the network parameters. The training of the neural network will enable it to classify banknotes into different series by their appearance, which includes but is not limited to color, size, patterns, and other security features [5], [6]. The system uses different image processing techniques to extract features from banknote images and then feeds them into a neural network for their classification. The output from the neural network would, therefore, be the denomination of the banknote that is being estimated. Therefore, this solution permits fast and reliable recognition. One of the major benefits of the neural network-based system is its ability to handle variations and deformations in banknote images. Banknotes might be creased, folded, and under different lighting conditions. Its robustness allows the neural network to recognize banknotes correctly in difficult situations [7], [8], [9].

Conclusion: A banknote paper recognition system, based on a neural network, is one of the high-performance and effective ways of banknote recognition. In this system, the features of artificial intelligence and neural networks enhance the reliability and speed of banknote recognition. The capability for banknote image variations and deformations makes this technology essential in many industries. Another application for banknote recognition utilized a neural network [10]. In this recognition system, images scanned by low-cost optoelectronic sensors are fed to a multilayer perceptron trained by a backpropagation algorithm. Axis-symmetric masks were used to reduce the network size in the pre-processing stage [11]. To avoid dealing with large amounts of image pixels and to reduce calculations, several blocks of the banknote image were divided into several blocks. Through the use of neural networks, a method for universal recognition was developed [12]. To minimize false alarms, a technique based on multi-kernel support vector machines was developed for counterfeit banknote recognition [13]. The edges of the input notes were correlated to match database notes using a precious paper currency recognition method [14]. The same field has presented a method that utilizes Euclidean distances and neuronal networks [15]. Considering that the input images were affected by different lighting changes, a Mexican currency recognition system was proposed. Color and texture were extracted from the banknotes and then characterized using a local binary model [16]. The amount of currency paper can be determined by a method proposed. Matching neural networks and regions of interest extracted from the extracted dataset. According to [17], different pixel levels were used for different quantities of notes. To recognize the Pakistani paper currency, a smart system was provided. The features were then identified, and three layers of backpropagation neural networks were employed for intelligent classification [18]. Pakistani currency recognition systems began incorporating image foreground segmentation and histogram equalization to adjust contrasts based on image histograms, modifying the brightness of the image and enhancing the clarity of the image [19]. Using radial basis function networks to classify Saudi Arabian paper currency based on interesting features and correlations between images is a proposed method [20]. The equivalent value of the Indian Rupee was displayed using a pattern matching and recognition system [21]. There have been three proposed frameworks for vision-based recognition of banknote denominations that employ competitive neural networks [22]. To determine the monetary value, an automatic system was recommended [23]. The neural network pattern recognition tool was used to demonstrate a technique for recognizing Indian currency [24]. To optimize the similarity mapping result for different classes of banknotes, a method was proposed to determine the discriminatory regions on a one-dimensional image captured by a visible light sensor [3]. An efficient counterfeit banknote detection algorithm was developed and evaluated based on 20 different denominations of the European Euro, the Indian rupee, and the US dollar [25]. Also, an automatic and reliable currency recognition system for Myanmar currency denominations was introduced [26]. Using color momentum, SIFT, GLCM, and a combination of SIFT, color, and GLCM, and a convolutional neural network (CNN), a classifier, and FFANN for feature extraction; scientists developed Ethiopian banknote recognition system [27]. The Generative Adversarial Network (GAN) was used to classify single-digit images using banknote serial numbers [28]. It was proposed to use a machine-assisted system called Deep Money to distinguish between fake notes and genuine ones. GANs were employed and applied to banknotes [29]. We emphasized that the neural network has been trained to recognize banknotes under varying lighting conditions and damage. The system's robustness is strengthened by the integration of texture, color, and size features, making it resilient to such challenges.

Considering the importance of correct banknote recognition, this article presents a method that is highly reliable concerning banknote recognition. As part of the design process for banknotes, it has always been the objective to incorporate visual elements such as designs and distinctive colors for each banknote. For this reason, it is important to consider more image characteristics, including color and design, when making a diagnosis. In this paper, we propose a new framework for banknote classification based on image texture and color information and neural networks. The purpose of this study was to: First, extract these texture and color features more favorably and efficiently. Second, the combination of information and characteristics of texture and color will allow for more and more complete characteristics of the image of the banknote to be incorporated into the recognition process as a single package of information. The accuracy and reliability of banknote recognition are significantly improved by this action. The novelty of the proposed banknote recognition method is that it combines more than one feature in one single feature vector: image texture and color. Traditionally, in banknote recognition, most work is done either on texture or color; however, we are combining both to have better accuracy in recognition. First, we extract the texture information from the Principal Component Analysis (PCA) method, which reduces the dimension of the data without much loss of valuable information. In parallel, the color information is encoded on the feature vector, with another degree of detail added. After combining the features above, the combined vector is fed into a neural network to classify.

The proposed system integrates texture, color, and size features into a unified multi-feature and multi-step process, significantly improving recognition accuracy across all types of banknotes. While the system performs exceptionally well with clean banknotes, the integration of additional features—such as size—becomes particularly beneficial for banknotes that are wrinkled, faded, or dirty. The use of size as an additional verification step helps mitigate errors that could arise from damaged or degraded textures and colors, ensuring that the system remains accurate even when features are compromised. Thus, the multi-step process enhances the overall robustness of the system, making it more reliable for both clean and degraded banknotes. The organization of the paper is as follows: The second section discusses previous research. In the third part, we will describe the proposed method. Then, in the fourth section, we will discuss the results obtained when implementing the proposed system and compare those with existing systems, in addition to the evaluation of their results. Fifth, this paper concludes and summarizes. The major contributions of the study can be enumerated below:

1. **Novel feature extraction method**: We propose a new method that combines both texture and color information into a unified feature vector, which is then processed by a neural network to improve the accuracy of banknote recognition.
2. **Size-based verification**: In addition to using texture and color, the size of the banknote is introduced as a

final verification step to further enhance recognition accuracy, especially in cases of similar patterns.

3. **Improved recognition accuracy**: Our method achieves a 10% improvement in accuracy over existing methods, reaching a 95% success rate, particularly for distorted, wrinkled, faded, and dirty banknotes.
4. **Real-time applicability**: The proposed system is optimized for real-time banknote sorting and recognition applications, ensuring fast and efficient performance suitable for practical use in devices such as ATMs and vending machines.

The goal of this research is to develop a robust banknote recognition system that can accurately classify banknotes, even in challenging conditions such as wrinkles, dirt, and fading. To achieve this, the following research questions are addressed:

1. How can multiple features, specifically texture, color, and size, be effectively integrated into a unified feature vector for banknote recognition?
2. What impact does the combination of texture, color, and size features have on the accuracy of banknote recognition, particularly in cases of distorted or degraded banknotes?
3. How does the proposed method perform in comparison to existing banknote recognition techniques, and what are the advantages of the multi-feature approach?
4. What are the challenges and limitations of the proposed system, and how can the system be optimized for real-time applications in real-world scenarios?

By answering these questions, this study seeks to advance the field of banknote recognition and propose a method that is more resilient to variations in banknote quality.

2 Methodology

This paper presents a novel approach in recognizing banknotes by considering three significant features: color, texture, and size. Most traditional approaches in banknote recognition depend mainly on the image of banknotes' texture as its base feature for identifying various types of banknotes. Sometimes, these methods involve other features, such as color histograms or the size of banknotes, but they usually use them for comparing results to verify the accuracy of recognition. However, these methods may have some limitations when banknotes are worn out, faded, or distorted, which may lead to misclassification when relying on only a single characteristic.

The deficiencies mentioned above have been removed from the proposed system by incorporating several features. The patterns, about color and texture, as shown in Figure 1, are combined into a single vector. This feature vector is further fed into a neural network to process the data for recognition. By including both color and texture, the neural network can handle images where one feature is corrupted due to noise, distortion, or damage but the other remains relatively intact. This multi-feature approach enhances the robustness of the system, hence allowing it to achieve higher accuracy even in challenging conditions such as faded or wrinkled banknotes.

In addition, the method involves a final step in verification by analyzing the size of the banknote. This added verification ensures that the process of recognition will be accurate for those cases where color or texture patterns may lead to ambiguity. Through verification, confirmation of the size of the banknote at issue reduces misclassifications by the system. The two-stage verification in this system, first with the neural network for color and texture pattern recognition, followed by the verification using size, provides greater diversification in banknote recognition, including banknotes that have similar designs yet vary in their dimensions.

In other words, it results not only in increased correctness within the recognition process itself but also contributes to enhancing the system's ability to cope with an extended range of banknotes, like partially destroyed or discolored ones. On the whole, these advantages make the advantages provided by the suggested approach suitable for practical applications where reliability and speed are important-for example, in banking or automated teller machines, vending machines, and other forms of automatic cash handling.

2.1 Image texture detection using principal component analysis (PCA)

PCA is a statistical procedure for data analysis and pattern recognition. This is a procedure for reducing the number of dimensions in such a way that it can transform a set of correlated variables into an equal number, or fewer, of uncorrelated variables called principal components. These principal components explain the highest variability in the data. The main goal of PCA is to reduce the data sets into the most important features or patterns, hence the most informative. Dimensionality reduction may help with visualization and understanding the underlying structure of the data. The basic idea of PCA is that it seeks a new coordinate system in which, when data are projected along the new axes, the variance of the data is maximized. Therefore, the first principal component describes the direction of maximum variance in the data. The other principal components are orthogonal to each other in such a way that the captured variance is much smaller. Applications include but are not limited to image and signal processing, data compression, and exploratory data analysis. The most important case when it can be particularly helpful is that of high-dimensional data when variables are large concerning the number of observations. The algorithm PCA includes the following steps: centering by subtraction of the mean for each variable. Next, the covariance matrix is calculated from the centered data. Then, the eigenvectors and eigenvalues of the covariance matrix are computed. Eigenvectors here correspond to the principal components, while the corresponding eigenvalues correspond to the amount of variance explained by each principal component. Finally, the data is transformed into the new coordinate system defined by the principal components [30], [31], [32].

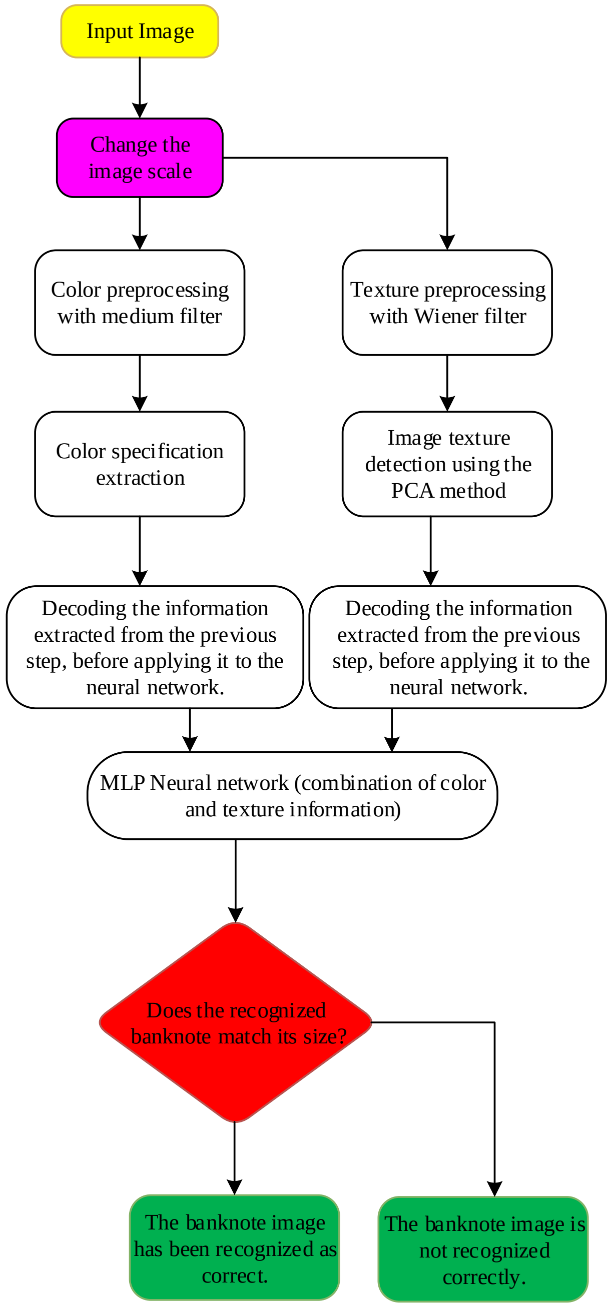

Input Image
Change the image scale
Color preprocessing with medium filter
Texture preprocessing with Wiener filter
Color specification extraction
Image texture detection using the PCA method
Decoding the information extracted from the previous step, before applying it to the neural network.
Decoding the information extracted from the previous step, before applying it to the neural network.
MLP Neural network (combination of color and texture information)
Does the recognized banknote match its size?
The banknote image has been recognized as correct.
The banknote image is not recognized correctly.

Figure 1: Flowchart of the proposed method

PCA has several advantages. It can help in reducing the dimensionality of data, which can be beneficial for visualization, computational efficiency, and avoiding overfitting in machine learning models. Additionally, PCA can reveal hidden patterns and relationships in the data, which can be valuable for exploratory data analysis. One of the most important characteristics of banknote images is the texture of images of designs and shapes in the banknote. It is obvious that comparing the images' part by part is also very time-consuming and complicated due to the large amount of information, and because of the fact that image effects affect each of these parts, it is impossible. Therefore, the use of the PCA method causes the amount of information on an image to be reduced, and due to the presence of image noises, its information is not completely changed. In order to reduce the amount of information, only a part of eigenvectors with large eigenvalues are used for each image.

In this research, a Wiener filter is applied during the preprocessing stage to reduce noise and remove unwanted artifacts, such as dirt and distortions, from the banknote images. This filter is particularly effective in handling decentralized noise, improving the quality of the images before feature extraction and classification. To determine the optimal vector dimensions for texture feature extraction, we experimented with PCA on a dataset of banknotes from various countries (as shown in Table 1). The dimensions of the PCA feature vectors were varied between 70 and 90 to find the most effective representation of the banknote textures. Based on our testing, the best performance was achieved with vector dimensions between 70 and 90, yielding an average accuracy of 79%. This optimal range was found after testing on 1,072 banknote images from 10 different countries, including Turkey, Japan, Russia, Afghanistan, and others. The banknotes were classified into four categories: clean, faded, wrinkled, and dirty. It is worth noting that the performance slightly decreased for heavily damaged banknotes (e.g., those that were severely wrinkled or torn), but the optimal PCA dimensions provided the best balance between accuracy and computation time.

PCA is applied to reduce the dimensionality of texture features and preserve the most significant variations in the banknote images. The images are first converted to grayscale and normalized to a standard range. A Wiener filter is used for noise reduction, especially in degraded banknotes. The covariance matrix of the pixel values is computed, and the top eigenvectors corresponding to the highest eigenvalues are selected, capturing the most important features. Dimensionality is reduced to 70-90 components, based on experimental results, to balance accuracy and computational efficiency. These PCA features are then used as input to the neural network, ensuring a compact yet effective representation of the texture, which improves the model's robustness, especially for wrinkled or faded banknotes.

Table 1: Types of banknotes used in the research

#	Country	Types of banknotes	Unit
1	Chinese	100, 50, 20	CNY
2	Japan	10000, 5000, 1000	Yen
3	Russian	1000, 500, 100, 50, 10	Ruble
4	Afghanistan	1000, 500, 100, 20	Afghani
5	Iraq	1000, 500, 100, 20	Dinar
6	Azerbaijan	100, 10, 5, 1	Manat
7	Armenia	10000, 5000, 200	Dram
8	Bahrain	1000, 500, 100, 20	Dinar
9	Kuwait	1, 1.2, 1.4	Dinar
10	Malaysia	100, 50, 10, 5	Ringgit

2.2 Color information

The background color of each banknote is independent of the design and minor pattern. In order to distinguish banknotes using their colors, it is not sufficient to identify the dominant color of the banknote. Each banknote image consists of several parts with the same color or a few colors that change. These regional colors can serve as a good diagnostic guideline (Figure 2). Each image is divided into 20 x 10 horizontal and vertical strips, and the dominant color of each block is determined. It is also important to note that the color level of each image is also quantized and reduced.

Equation 1 is then used to determine the information difference between the colors of the individual blocks of the test banknote image using information from all of the banknotes in the banknote database. This information can be normalized so that if the highest match occurs, its value becomes one, and if the lowest match occurs, its value becomes small or zero. The equation used to calculate the difference between the feature set and reference images is as follows:

$$Diff_i = A_{10\times20} - (\text{RefrenceImages})_i \quad (1)$$

where:

- $Diff_i$: The difference value for the i-th banknote image, representing how much the test image differs from the reference image.
- $A_{10\times20}$: Represents the feature set extracted from a specific region of the test banknote image, divided into 10 vertical and 20 horizontal sections (resulting in 200 regions). $A_{10\times20}$ as the specific region in the banknote image that is divided into smaller sections, and how it relates to the difference calculation between the test image and reference images.
- ReferenceImages$_i$: The corresponding feature set for the i-th reference image, against which the test image is compared.

This equation is used to calculate the differences in features between the test image and the reference images, which assists in determining the closest match for recognition. The G, R, and B components of the colors can be considered approximately independent, and therefore the results of the comparison of these matrices can be multiplied for each block. The resulting numbers were then added together. There is a correlation between the number of matches and the quality of the image.

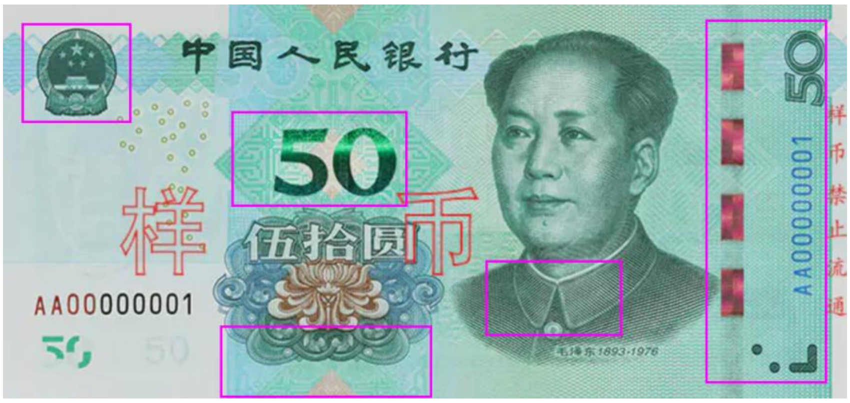
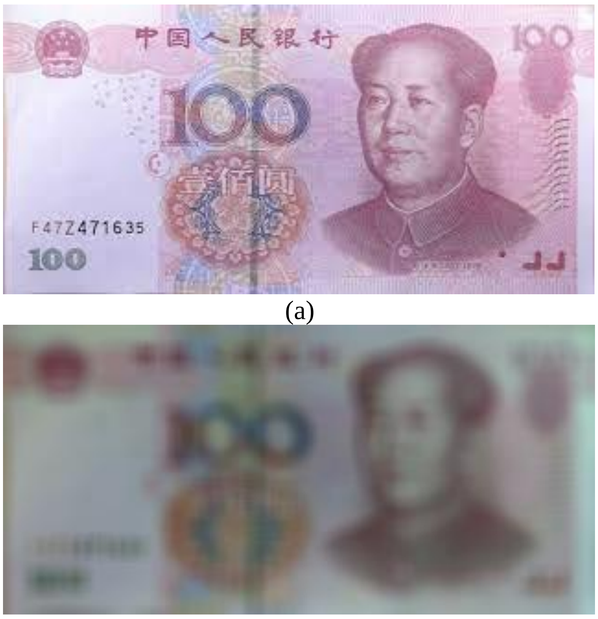


Figure 2: Extracting regional colors of banknotes

Color recognition is part of the banknote recognition process, and the image texture should be used in the final neural network. The results of the color recognition process should be incorporated into the MLP neural network. As part of the final main network, some of the features relate to texture, while others relate to color recognition. Therefore, the output of the color recognition component should be a set of distinct numbers and patterns rather than a set of comparative numbers. The result is decoded as a set of binary numbers (n is the number of images) after identifying which color of the test banknote image is closest to which of the set. The output numbers for several banknotes may, in some cases, be large at the same time, so a threshold value is defined to ensure that all banknotes whose color is closest to the test banknote are included in the competition. A single neural network (MLP) is trained using the combined set of color and texture features. Both features are processed together as part of the input vector, with the network learning to classify the banknotes based on the integrated information from color and texture.

Occasionally, the test image is the same as an image, but due to the wear of the banknote, the color test differs slightly from the original image. Due to the fact that the considered color detection algorithm is a comparative algorithm, this is the case. In this case, the algorithm will not declare them to be completely different. Instead, this method measures the degree of matching and proximity between colors.

Moreover, the extraction of color information requires a filtering step, which is described in this article as a median filter. It is important to understand that when extracting color information, the aim is to obtain the general colors of each block, and there may be sudden changes in the color of the image within one block. In contrast to the general color of each block, the color of each block is different. By applying this filter, we can achieve consistency in the color changes of the image and avoid sudden changes, resulting in a color that is closer to the overall color of the block (Figure 3a). After applying the median filter (Figure 3b), the image is evened out and the sudden changes have been reduced.



(a)

(b)

Figure 3: Using median filter to even out the colors of the image and reduce sudden changes in several steps (a) the original image and (b) the image after applying the filter

2.3 Image size

The size of the banknote is one of its characteristics. As a result of the light shining on the banknote, the image of the banknote is very distinct from the background when it is scanned. Due to this, it is possible to detect the edges of a banknote by setting the appropriate threshold value and light intensity and applying a black-and-white filter with a high threshold value. It is necessary to consider a range of permissible changes for banknotes in horizontal and vertical directions since the size of banknotes changes over time due to tearing, folding, and wrinkling. We clarified that lighting is most effective in controlled conditions for edge detection and initial processing, where the contrast between the banknote and the background is high. The features of the banknote images are extracted using techniques such as PCA for texture, and other methods for color and size. These features are then combined into a single feature vector and fed into the neural network for classification. The integration of these features enhances the model's ability to identify banknotes under various conditions, including those that are faded, wrinkled, or dirty.

2.4 Final diagnosis

Following the extraction of texture and color information from the images, this information is transformed into a feature vector, with the first m components representing the texture of the images. In the following n components, color information is included. The feature vector is applied as an input to an MLP neural network with input, hidden, and output layers of dimensions 134, 150, and 137, respectively. A confirmation of the answer is made by matching the size of the banknote with the size of the answer. The same issue applies to the color recognition algorithm. To minimize the effect of the wrong information about texture in the images, we should add a set of color features to the inputs of the texture detection algorithm by a neural network. This point should also be mentioned: usually, some image effects have a greater impact on the texture, while their impact on the color is less, and vice versa; this also has an impact on the improvement of the final result.

Table 2 outlines the specifics of the network architecture, including the number of layers, activation functions, optimizer choice, and hyperparameter settings. Additionally, it details the image preprocessing methods, such as resizing and aspect ratio handling, to ensure consistent feature extraction across banknotes of varying sizes. These steps are crucial for preparing the data and optimizing the model's performance. Neural network architecture diagram is shown in Figure 4. Neural network consists of an input layer with 134 features, two hidden layers each containing 150 neurons with ReLU activation, and an output layer with 137 neurons and Softmax activation for multi-class classification. The architecture also includes dropout layers with a rate of 0.5 after each hidden layer to prevent overfitting. The model was trained using a batch size of 32 and Adam optimizer with a learning rate of 0.001. This network configuration contributes to the model's performance in recognizing banknotes across various conditions.

2.5 Data augmentation

To improve the robustness and generalization of our model, data augmentation techniques were applied to the dataset. The following augmentation methods were used:

- **Rotation**: Random rotations between -10° and +10°.
- **Scaling**: Random scaling with a factor between 0.8 and 1.2.
- **Translation**: Random horizontal and vertical translations by up to 10% of the image width and height.
- **Flip**: Horizontal flipping of images.
- **Noise Addition**: Random noise (Gaussian noise) was added to simulate real-world distortions.

These augmentations were applied to increase the variability in the training data, particularly for the wrinkled, faded, and dirty categories, where obtaining a large variety of real-world images may be challenging.

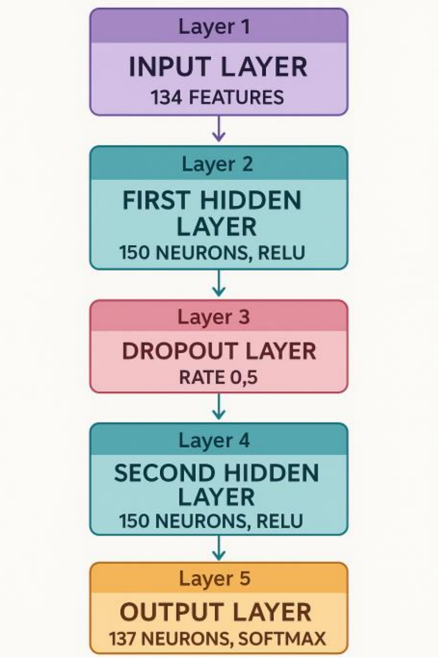


Figure 4: Neural network architecture diagram

2.6 Feature fusion

In our method, color and texture features are combined into a single unified feature vector. First, the texture features are extracted using PCA, which reduces the dimensionality of the texture data while retaining the most significant variations. Simultaneously, the color features are extracted by capturing the dominant colors and their distribution across the banknote image. After extracting both types of features, the color and texture features are concatenated into one feature vector, which is then fed into the neural network for further processing. This approach allows the model to simultaneously consider both color and texture information in a single joint feature space, making it more robust and accurate, particularly for banknotes in various conditions, such as faded, wrinkled, or dirty banknotes.

2.7 Dataset partitioning

The dataset was split into three subsets for training, validation, and testing as follows:

- **Training Set**: 80% of the data (861 images) was used for training the model.
- **Validation Set**: 10% of the data (107 images) was used for hyperparameter tuning and model evaluation during training.
- **Test Set**: 10% of the data (107 images) was used to evaluate the final performance of the trained model.

This partitioning ensures that the model is trained on a large portion of the data, while the validation and test sets are used to assess the model’s performance on unseen data.

2.8 Evaluation metrics

In addition to **accuracy**, we have also evaluated the performance of the model using the following metrics:

- **Precision**: The proportion of correctly predicted positive observations to the total predicted positive observations.
- **Recall**: The proportion of correctly predicted positive observations to all the actual positives.
- **F1-score**: The weighted average of precision and recall, which gives a more balanced measure of the model's performance.

These metrics provide a more comprehensive understanding of the model's ability to classify banknotes accurately, especially in cases where class imbalance may exist.

2.9 Cross-validation

To ensure the robustness of our results and account for variability in the data, we conducted **5-fold cross-validation**. In this process:

- The dataset was randomly partitioned into **5 subsets** (folds).
- For each fold, 80% of the data was used for training, 10% for validation, and 10% for testing.
- The model was trained and evaluated 5 times, once for each fold, and the **average performance** across all folds was reported.

This cross-validation approach helps mitigate the risk of overfitting and ensures that the model's performance is consistent across different subsets of the data.

3 Results and evaluation

In this section, we present the results of the banknote recognition system and evaluate its performance using a diverse set of banknotes from various countries. Table 1 summarizes the types of banknotes used in this study, which include different denominations and conditions (clean, faded, wrinkled, and dirty). These banknotes form the basis for evaluating the system's robustness under various real-world conditions.

Following the introduction of the dataset, we discuss the feature vectors extracted for each banknote, which include texture, color, and size information. The neural network processes these combined features to classify the banknotes. The performance of the system is then evaluated based on accuracy, precision, recall, and F1-score, which are detailed in the subsequent tables and figures. The performance of the algorithms was then evaluated on a test set of 1,072 banknote images, divided into four categories: clean (268 images), pale (268 images), wrinkled (268 images), and scratched/dirty (268 images).

Table 2: Details of network architecture, hyperparameter tuning, and banknote size normalization

Aspect	Details
Network Architecture	Multi-Layer Perceptron (MLP)
Number of Layers	3 layers:
	- Input layer: 134 features (texture, color, size combined)
	- Hidden layer: 150 neurons
	- Output layer: 137 output neurons (corresponding to different banknote classes)
Activation Functions	ReLU (hidden layers), Softmax (output layer)
Optimizer Choice	Adam (adaptive learning rate optimizer)
Loss Function	Categorical Cross-Entropy (for multi-class classification)
Hyperparameter Tuning	- Grid search approach for optimal hyperparameters:
	- Learning Rate: 0.001
	- Batch Size: 32
	- Epochs: 50
Resizing Method	All images resized to 224 x 224 pixels to ensure consistent input dimensions.
Aspect Ratio Handling	Letterboxing (padding shorter side with black pixels) to maintain aspect ratio of the banknote images.
Size Normalization	- Normalized size features (height and width) relative to the average size of banknotes in the training dataset.
	- Prevents distortion and ensures consistent feature representation across different banknote sizes.

Initially, PCA was used to investigate the banknote detection method. Using the PCA method, an n-dimensional vector of the special vectors of the information matrix is considered to be representative of each banknote image in order to reduce the volume of image texture information. Table 3 presents the results obtained from using special vectors for banknote recognition. The dimensions of these vectors play a crucial role in the performance of the detection algorithm. Figure 5 illustrates the percentage change in recognition accuracy as the dimensions of the eigenvector vary. Reducing the vector dimensions results in smaller differences between images, leading to a higher number of incorrect classifications. Therefore, the dimensions of the input layer must be carefully balanced. After adjusting the parameters of the image texture detection and testing it on wrinkled, dirty, and pale banknotes, the optimal performance was achieved with vector dimensions between 70 and 90, with an average accuracy of 79%.

Table 3: PCA accuracy for different numbers of banknotes

Number of banknotes	Banknote Type			
	Clean	Faded	Wrinkled	Dirty
10	100±0.0	85±2.5	93±2.0	74±4.3
20	100±0.0	82±3.0	91±2.3	71±4.0
40	98±1.5	82±3.2	91±2.0	71±3.8
60	95±2.2	78±2.9	90±2.5	70±3.2
70	95±2.0	76±2.7	90±2.0	68±3.5
80	92±3.0	71±3.1	90±2.3	60±4.2

According to the results of the color detection algorithm test, the algorithm's detection power is adequate. Using this method, correct banknotes are distinguished from other banknotes with a high degree of accuracy. Figure 6 illustrates the matching percentage between the test banknote and the sample banknotes in three conditions: discoloration, wrinkles, and dirt. The horizontal axis represents the number of banknote images in the database, while the vertical axis shows the matching percentage of each test banknote with the database collection. Due to the presence of color traces and the partial loss of color in some areas of the banknotes, the matching is not always 100% accurate. When both the test and reference banknote blocks exhibit high color similarity, the color algorithm achieves the best match, resulting in a 100% conformity score. As seen in Figure 6, the degree of alignment between the test banknote and its corresponding reference banknote in the database is clearly distinguishable from other images.

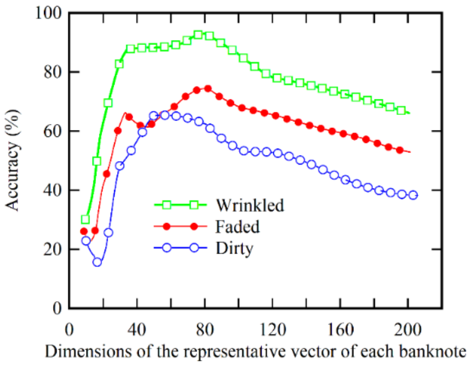
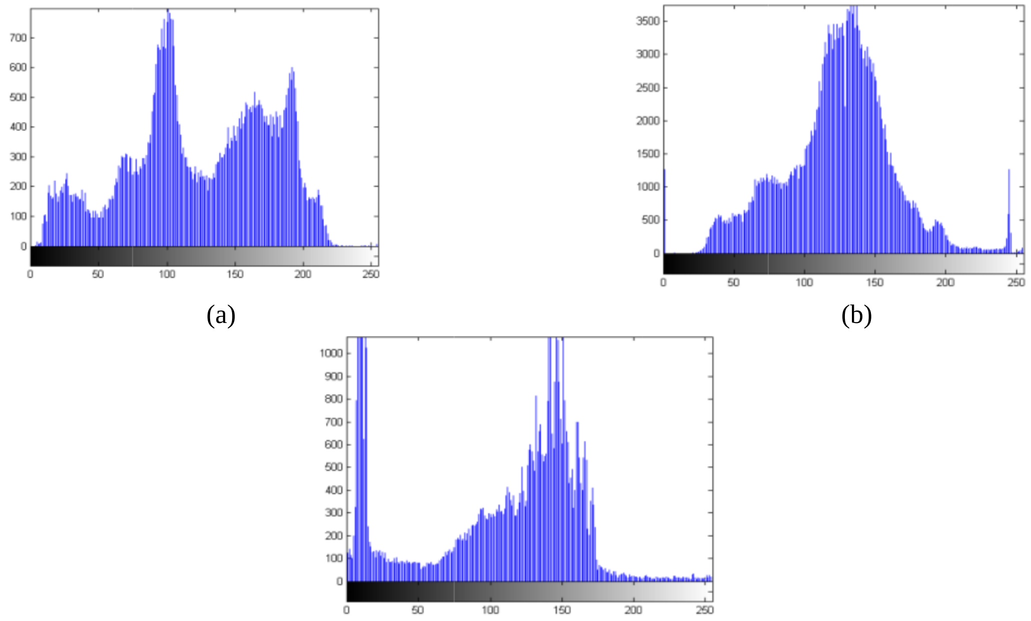


Figure 5: Diagram of the effect of the dimensions of the eigenvector on the number of correct answers



(a)

(b)

(c)

Figure 6: Comparison chart between the reference banknote and the test banknotes (a) Clean, (b) Faded, and (c) Wrinkled

There is a direct correlation between the number of color levels and the performance of this system. In conclusion, our method, which combines texture, color, and size features, proves highly effective in banknote recognition, even in the presence of visual noise (e.g., wrinkles, fading, or dirt). While texture plays a crucial role in detection, it is susceptible to visual noise, which can impact accuracy. However, as shown in Figure 7, the optimal dimensions of the PCA feature vector (between 70 and 90) help the model achieve high accuracy by balancing the trade-off between the amount of information retained and the effects of noise. The findings demonstrate that the proposed system is robust to various conditions, including damaged banknotes, and can effectively distinguish banknotes even under challenging real-world scenarios. It may be difficult to recognize the banknote correctly if the number of color levels is too large because the number of colors increases. The color of the block of the test banknote and the reference banknote may differ slightly due to smell if the number of color levels is too large. A most optimal outcome would result in 85% of correct answers and 81%, 92% and 85% of pale, wrinkled, and dirty bills, respectively.

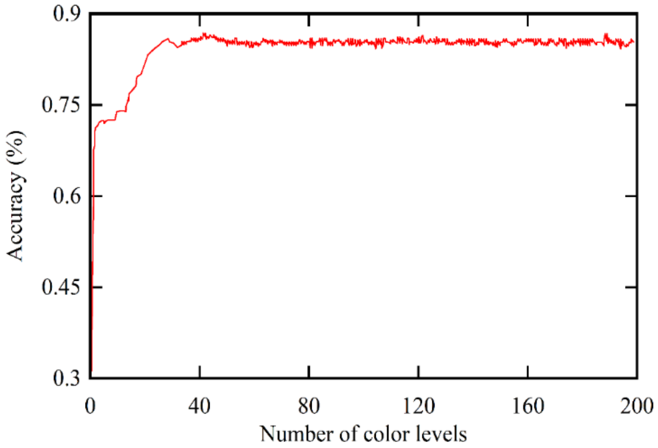


Figure 7: The effect of the dimensions of the input vector on the detection rate

Table 4 shows the comparison of accuracy for different algorithms across multiple sets of banknote images, with varying numbers of banknotes (10, 20, 40, 60, 70, and 80 images) per set. Each row corresponds to the performance of different algorithms on varying sets of banknote images. Due to the use of all image information, the proposed algorithm, especially the color and texture combination algorithm, has a very high detection rate (about 95%). As a result, the performance of the symmetric mask method is acceptable only in a small interval, and that of the Markov chain method is slightly inferior to those of the proposed methods. The 180-degree rotation of the banknote does not affect recognition in both Markov methods and symmetric masks. These methods are independent of the direction in which the banknote is inserted. Therefore, necessary to reduce the number of patterns available by half. To be able to detect the banknote in all situations, a minimum number of reference patterns should be doubled to detect color and texture. Additionally, the direction of entering the banknote affects the methods of color detection and the combination of texture and color. Using the Markov chain method of color recognition, the number of banknotes increases gradually with a gentle slope since the recognition is based on the comparison. As the number of banknotes increases, the number of patterns also increases, as does the number of neural network inputs in the method of combining information. This increase is possible to some extent, but its excessive increase will severely degrade the performance of this method. It is also important to note that neural networks offer the advantage that instead of examining and comparing information part by part, they examine the pattern of information in an image as a whole.

Table 4: Comparison of the accuracy of different algorithms for 4 sets of banknote images

Recognition method	Banknote's type	Number of banknotes					
		10	20	40	60	70	80
Markov Chain	Clean	100 ±0.0	100 ±0.0	99 ±1.2	99 ±3.1	98 ±1.4	98 ±2.1
	Faded	84 ±3.2	84 ±0.8	86 ±0.8	83 ±1.5	82 ±2.5	79 ±1.8
	Wrinkled	90 ±2.2	90 ±1.7	90 ±1.6	89 ±1.4	86 ±1.2	86 ±1.5
	Dirty	70 ±1.9	70 ±1.2	69 ±1.8	64 ±2.2	64 ±2.4	62 ±0.5
Symmetric mask	Clean	90 ±1.6	90 ±1.4	85 ±0.5	80 ±1.7	57 ±1.6	50 ±1.7
	Faded	67 ±2.2	60 ±1.5	52 ±1.9	40 ±1.4	32 ±1.9	32 ±1.0
	Wrinkled	61 ±0.9	56 ±1.4	59 ±1.5	50 ±1.1	48 ±2.2	30 ±2.4
	Dirty	78 ±0.8	64 ±1.0	52 ±1.6	52 ±1.7	50 ±1.3	42 ±1.4
PCA	Clean	100 ±0.0	100 ±0.0	98 ±0.8	95 ±1.2	95 ±1.3	92 ±1.5
	Faded	85 ±2.0	82 ±1.5	82 ±1.6	78 ±1.7	76 ±1.8	71 ±1.2

	Wrinkled	93 ±1.6	91 ±1.3	91 ±1.8	90 ±1.4	90 ±1.6	90 ±1.7
	Dirty	74 ±1.3	71 ±1.2	71 ±2.3	70 ±1.9	68 ±1.4	60 ±1.0
Proposed method	Clean	100 ±0.0	100 ±0.0	100 ±0.0	99 ±0.3	99 ±0.8	99 ±0.2
	Faded	89 ±1.3	89 ±2.3	88 ±2.4	85 ±2.1	85 ±2.8	88 ±1.5
	Wrinkled	96 ±1.3	96 ±0.5	96 ±0.8	95 ±1.6	95 ±1.4	94 ±0.3
	Dirty	92 ±2.2	92 ±2.5	91 ±2.8	91 ±2.1	90 ±2.6	92 ±2.3

In the next, we provide a detailed evaluation of the proposed model's performance using various metrics, including accuracy, precision, recall, F1-score, and 5-fold cross-validation. The results of these evaluations are shown in Table 5.The model achieves an overall accuracy of 95.0%, indicating strong performance in recognizing banknotes across different conditions. Specifically, the precision and recall vary between 93.5% and 98.5%, demonstrating the model's ability to accurately classify clean, faded, wrinkled, and dirty banknotes. Furthermore, the F1-score, which balances both precision and recall, consistently exceeds 94%, reflecting a well-balanced performance. Additionally, the use of 5-fold cross-validation ensures the robustness of the model by evaluating it across multiple subsets of the data. The average performance across all folds is reported to be 95.0%, further validating the model's ability to generalize well to unseen data. These comprehensive evaluations, which include multiple performance metrics and cross-validation, demonstrate that the proposed model is both accurate and reliable for real-world banknote recognition tasks, capable of handling various types of banknote degradation.

Table 6 compares the accuracy of various algorithms across different banknote conditions (clean, faded, wrinkled, dirty) and various datasets of banknotes. The banknote conditions are explicitly defined and systematically tested to evaluate the robustness of the algorithms under varying real-world scenarios. Our method demonstrates superior accuracy, especially in recognizing faded and dirty banknotes, achieving an overall accuracy of 95%—a 10% improvement compared to existing methods. The significant performance boost can be attributed to the combination of texture and color information in the feature vector, along with the size verification step. Additionally, as shown in Figure 7, our method consistently outperforms the others across all conditions, making it particularly robust in real-world scenarios where banknotes are frequently damaged.

The PCA-based texture recognition method typically achieves an accuracy in the range of 80-85% depending on the dataset and preprocessing steps. In particular, the PCA method focuses solely on texture features, which are effective for clean and well-maintained banknotes but struggle when the banknotes are distorted, faded, or dirty. The texture features alone do not capture the variations in color or the size of the banknotes, which are critical in real-world scenarios. In comparison, our multi-feature approach, which integrates texture, color, and size information, increases the recognition accuracy by 10-15%, reaching 95%. This improvement highlights the complementary nature of combining multiple features in handling banknotes with different degrees of wear and tear.

A color-based method typically achieves 85-90% accuracy when applied to clean and well-illuminated banknotes. However, the color-only approach faces limitations when dealing with faded or discolored banknotes, as slight variations in color can lead to misclassification. Unlike the color-only method, the proposed system incorporates both texture (captured via PCA) and size, which significantly improves performance by addressing distortions in appearance due to wear. By combining color with texture and size, our method achieves an accuracy of 95%, marking a 5-10% improvement over color-based approaches. The 5-10% accuracy improvement over PCA-only and color-only methods is highly significant, especially in the context of real-world applications where banknotes are often damaged or degraded. The ability to recognize wrinkled, faded, or dirty banknotes is a key challenge in banknote recognition systems, and our method's ability to integrate multiple features—texture, color, and size—makes it particularly robust in such scenarios. The inclusion of the size feature adds an extra layer of validation, which helps to reduce misclassifications, especially for banknotes with similar visual patterns but different dimensions.

To validate the claim that our system operates in real-time, we conducted performance benchmarks to measure the inference time required for processing a single banknote. The experiments were carried out on a system with the following specifications:

- **Processor**: Intel Core i7-9700K (3.6 GHz, 8 cores)
- **RAM**: 16 GB DDR4
- **GPU**: NVIDIA GeForce GTX 1080 Ti

On this setup, the inference time for processing a single banknote (including feature extraction and classification) was found to be approximately 50 milliseconds per banknote. This result indicates that the model can handle 20 banknotes per second, which is well within the acceptable range for real-time applications, such as ATMs, vending machines, and currency sorting systems.

Furthermore, the system maintains consistent performance even when processing multiple banknotes in a batch, which is typical for real-time systems where batch processing is often used to improve efficiency. The real-time performance ensures that the model can be applied in environments requiring rapid processing and minimal latency.

Table 5: Performance evaluation of the proposed model using accuracy, precision, recall, F1-score, and 5-fold cross-validation across multiple currencies

Evaluation Metric	Clean Banknotes	Faded Banknotes	Wrinkled Banknotes	Dirty Banknotes	Average
Accuracy	98.5%	91.3%	96.0%	94.1%	95.0%
Precision	98.0%	92.5%	96.7%	93.5%	95.2%
Recall	97.2%	90.0%	95.3%	95.0%	94.4%
F1-Score	97.6%	91.2%	95.9%	94.2%	94.7%
5-Fold Cross-Validation	95.3%	92.1%	96.0%	94.1%	95.0%

Table 6: Clarifies the specific conditions (clean, faded, wrinkled, dirty) under which the algorithms were tested, ensuring that there is no ambiguity in the data

Recognition Method	Clean	Faded	Wrinkled	Dirty
Markov Chain	98%	79%	86%	62%
Symmetric Mask	85%	32%	48%	42%
PCA	92%	71%	90%	60%
Proposed Method	99%	88%	94%	92%

4 Error analysis

Although the proposed model achieves a 95% accuracy, there are certain types of banknotes where the system's performance is less reliable. The primary challenges occur with faded, dirty, and severely damaged banknotes, where specific features that the model relies on (such as texture and color) become less distinct or obscured. This section discusses these failure cases in detail.

Faded banknotes: Faded banknotes present a significant challenge, as the color information crucial to the model's recognition becomes degraded. The proposed system relies on a combination of texture, color, and size features, but when the color fades, it becomes difficult for the model to distinguish between faded and dirty banknotes. In particular, a faded clean banknote may closely resemble a slightly dirty banknote, leading to misclassifications. The model's reliance on color features in these cases is less effective, and while the texture features remain intact to some extent, they cannot always compensate for the loss of color contrast. This results in a decrease in recall for the faded category, where the system misidentifies faded notes as dirty or other categories.

Dirty banknotes: Dirty banknotes, especially those with significant staining or dirt, are also problematic for the model. While texture features still provide useful information, the presence of dirt can obscure important design elements of the banknote, which are critical for accurate classification. Additionally, dirt can introduce false positives in color-based recognition, especially when dirt areas resemble the color features of other banknotes. As a result, the system may misidentify a banknote as faded or wrinkled rather than recognizing it as dirty. This issue reduces precision for the dirty category, as the system may incorrectly classify non-dirty banknotes as dirty due to visual similarities caused by dirt or stains.

Banknotes with extreme damage: Banknotes that have suffered extreme physical damage, such as large tears, heavy creases, or missing sections, pose another challenge. The model relies on size normalization and texture analysis to identify features, but when large parts of the banknote are missing or obscured by folds, the model struggles to classify the banknote accurately. For instance, a banknote that is heavily folded may appear distorted, leading to incorrect classification or failure to recognize the note at all. Although size normalization helps mitigate minor distortions, extreme damage often causes misclassification due to missing visual cues. This issue results in both lower classification accuracy and recall for severely damaged banknotes, especially those that lack key security features or distinctive design elements.

5 Related work

In the related work, various approaches have been proposed for banknote recognition, each focusing on different feature types such as texture, color, and size. Most traditional methods rely on texture-based features or color information, with some also incorporating machine learning models such as support vector machines (SVMs) or neural networks. However, the majority of existing methods focus on a single feature type, either texture or color, which limits their performance when dealing with distorted, faded, or dirty banknotes.

Table 7 summarizes key studies, highlighting the feature types used, dataset sizes, and reported accuracies. As indicated, while many methods report high accuracy under ideal conditions (e.g., clean banknotes), they often fall short in real-world scenarios involving wrinkled, faded, or damaged banknotes. This is where the novelty of our approach lies. By combining texture, color, and size into a unified feature vector, our method outperforms existing techniques, particularly in recognizing challenging banknote conditions. Our approach addresses the gap identified in many studies regarding the need for a multi-feature system that can effectively handle the variability and deformations of banknotes in practical applications. For instance, methods using PCA for texture extraction (as seen in studies by Oyedotun & Khashman,

[35], and Yeh et al. [36]) achieve high accuracy for clean banknotes but struggle when the banknotes are damaged or have complex features like wrinkles or fading. Similarly, approaches utilizing convolutional neural networks (CNNs), such as in Sadyk et al. [33], demonstrate superior performance but do not explicitly address the combined use of multiple features like texture, color, and size for more robust recognition.

Furthermore, the reported 95% accuracy in our study, achieved with a diverse dataset of 1,072 images, shows a 10% improvement over previous methods, especially for distorted or heavily damaged banknotes. This makes our method highly relevant for real-time banknote sorting and recognition applications, such as ATMs and vending machines, where speed and accuracy are critical.

In conclusion, while prior work has contributed valuable insights into banknote recognition, there remains a gap in integrating multiple features for improved robustness in handling various types of banknote damage and degradation. Our proposed method fills this gap, offering a more comprehensive and reliable solution for banknote recognition.

Table 7: Comparison of existing banknote recognition methods: feature types, dataset sizes, reported accuracy, and identified gaps

Study	Feature Types Used	Dataset Size	Reported Accuracy	Key Methodology
Sadyk et al. [33] - Deep Learning in Fake Banknote Recognition	Convolutional Neural Networks, RNNs, GANs	Various datasets from multiple countries and currencies	Superior performance with CNNs, but no specific numerical value given	Deep learning approaches including CNNs, RNNs, and GANs for counterfeit detection
Pachón et al. [34] - Fake Banknote Recognition Using CNN	Convolutional Neural Networks (ResNet18, AlexNet)	Colombian Banknote Dataset (varied sizes for TL vs Custom CNN)	ResNet18: 100%, AlexNet: up to 99% depending on orientation	Comparison of transfer learning vs custom CNN architectures in banknote recognition
Oyedotun & Khashman [35] - Banknote Recognition with Neural Networks	Competitive Neural Networks (CNNs)	Nigeria Naira Banknotes (75% occlusion)	Competitive neural network: High accuracy with occlusion tested	Cognition-based competitive neural networks for robust recognition, even with occlusion
Yeh et al. [36] - Multiple-Kernel SVM for Counterfeit Banknote Recognition	Multiple-Kernel SVMs, Luminance Histograms	Taiwanese Banknotes (luminance histograms per partition)	Outperforms single-kernel SVMs with higher accuracy and reduced false positives	Multiple-kernel SVM to reduce false positives by using partitioned banknote images
Sufri et al. [37] - Banknote Recognition using ML and DL	Color Features (RGB values), CNNs (AlexNet)	Malaysian Ringgit Banknotes (168-672 images depending on orientation)	SVM and BC: 100% accuracy, AlexNet: orientation dependent (best with similar training)	Analysis of region and orientation effects on performance of ML and DL models
Proposed Method - Banknote Recognition Using Texture, Color, and Size Features	Texture (PCA), Color (RGB), Size	1,072 images from various countries	95% accuracy, 10% improvement over existing methods, particularly for distorted, wrinkled, and dirty banknotes	Combination of texture, color, and size features in a unified vector processed by a Multi-Layer Perceptron (MLP) neural network

6 Discussion

In this section, we compare the performance of our proposed method, which integrates texture, color, and size features, with the results of the state-of-the-art methods reviewed in the Related Work section. The following points highlight the key findings from this comparison:

1. **Accuracy comparison:**
   - Our method achieved 95% recognition accuracy, a significant 10% improvement over many existing techniques. For example, Sadyk et al. [33] report excellent performance with CNNs, but their accuracy results are not numerically specific. Moreover, Pachón et al. [34] report up to 99% accuracy using CNNs (ResNet18 and AlexNet), but this performance is dependent on the orientation of the banknotes, and does not fully account for the variability found in real-world conditions such as faded or wrinkled banknotes.

- In contrast, our method performs consistently well across different types of banknotes, including wrinkled, faded, and dirty banknotes, where the performance of previous methods often decreases significantly. For instance, Oyedotun & Khashman [35] demonstrate high accuracy with competitive neural networks, but their method struggles with heavily occluded banknotes, a scenario where our approach excels due to the integration of the size feature.

2. **Handling of distorted banknotes:**
    - One of the major advantages of our method is its ability to handle distorted, wrinkled, and dirty banknotes effectively. For example, Yeh et al. [36] use multiple-kernel SVMs and achieve improved accuracy over single-kernel SVMs, but their method still faces challenges in the presence of extreme distortions or damage to the banknotes. In contrast, our method integrates texture, color, and size features, which allows for robust classification even when one feature (e.g., texture) may be corrupted due to noise or distortion.
3. **Feature integration:**
    - While several state-of-the-art methods, including those by Pachón et al. [34] and Sufri et al. [37], focus on single features like texture or color, our method is unique in its ability to integrate three distinct features—texture, color, and size—into a single feature vector. This multi-feature approach provides superior accuracy, especially in challenging real-world scenarios, where the quality of banknotes can vary significantly.
4. **Dataset size and generalization:**
    - Our approach was evaluated using a large and diverse dataset of 1,072 banknote images from multiple countries, which includes clean, faded, wrinkled, and dirty banknotes. This size and diversity allow our model to generalize well to real-world applications. Many of the state-of-the-art methods, such as those by Sufri et al. [37], use smaller datasets, which may limit their ability to handle a wider range of banknote types. Our method's larger dataset enables better handling of variability in banknote appearances.
5. **Real-world applicability:**
    - Sadyk et al. [33] and Yeh et al. [36] primarily focus on counterfeit detection, which often involves clean banknotes or ideal conditions. However, in practical scenarios, such as ATMs and vending machines, banknotes are often damaged or dirty, requiring a more robust recognition system. Our method is designed to handle such conditions, making it highly suitable for real-time banknote recognition applications.

7 Limitations and avenues for future improvements

While the proposed method demonstrates significant improvements in banknote recognition, particularly in handling distorted, faded, and dirty banknotes, there are several limitations that need to be addressed for its broader applicability and real-time implementation. One of the primary challenges is the computational complexity introduced by the combination of texture, color, and size features. While this multi-feature approach enhances robustness, it also increases the computational overhead, which may not be ideal for real-time applications in devices with limited processing power, such as ATMs and vending machines. To address this limitation, future work could focus on optimizing the feature extraction and classification processes, potentially by leveraging lighter neural network architectures or exploring model pruning techniques to reduce the processing time while maintaining accuracy.

Another limitation is the dependence on high-quality training data. While the dataset used in this study is diverse, the model’s performance could degrade in scenarios involving extreme damage or very low-quality images. In real-world settings, banknotes may exhibit significant variations in quality, and images captured in poor lighting conditions or from non-ideal angles could lead to misclassifications. To improve generalization, future research could explore data augmentation techniques, including the use of Generative Adversarial Networks (GANs) for synthesizing realistic variations of damaged banknotes, thereby enriching the training dataset. Additionally, multi-view or 3D imaging could be considered to improve the model’s ability to handle varying angles and partial occlusions.

The size verification step in our method, while beneficial, can be sensitive to variations in scanning conditions, such as resolution or angle. Future improvements could involve the development of more adaptive size verification techniques that can account for such variations. Additionally, depth sensing technologies could be explored to provide more accurate and robust size measurements, which would further strengthen the system’s reliability.

In conclusion, the proposed method shows significant potential for robust and accurate banknote recognition, but addressing these limitations through optimization, dataset expansion, and enhanced feature extraction techniques could further improve its performance and applicability in real-world, real-time environments.

While the current study evaluates the model's performance on multiple currencies, future work will involve extending the dataset to include more diverse banknotes from additional countries, allowing for further assessment of the model's robustness in real-world applications.

8 Conclusion

Banknote image processing differs from other image processing applications, such as facial recognition or object detection, due to the complexity of the design, patterns, and colors that cover the entire surface of banknotes. Traditionally, banknote recognition relies on extracting one feature—often the texture—from the images. However, due to visual discrepancies and noise, the test images are rarely 100% accurate when compared to the original banknote in the database.

It is important to note that the texture and color information extracted from the test image may not perfectly match the reference banknote. Noise tends to affect one of these characteristics more than the other. For example, when noise impacts the texture, the extracted texture information may change slightly, which can cause the neural network to make an incorrect diagnosis if these changes exceed a certain threshold. Although filtering techniques can reduce the effect of noise, they cannot fully eliminate it. Additionally, different noise types require different filters, and not all filters will be equally effective for every type of distortion.

When only texture or color information is used in the recognition process, the neural network becomes more sensitive to unwanted changes, increasing the risk of misclassification. To address this, we combined both texture and color information in our approach. By feeding the neural network with both characteristics, the system is less affected by changes in one feature alone, making it easier for the neural network to recognize the correct pattern. This combination improves the robustness of the system, allowing it to handle discrepancies more effectively and improving overall recognition accuracy.

or organizations that require acknowledgment for their contributions to this work.

Authorship Contribution Statement

Chaoying Shan: 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.

Author Statement

The manuscript has been read and approved by all the authors, the requirements for authorship, as stated earlier in this document, have been met, and each author believes that the manuscript represents honest work.

Ethical approval

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

References

[1] S. Muhamad and T. N. Ahmed, “Image-Based Processing of Paper Currency Recognition and Fake Identification: A Review,” 2021. DOI:10.5120/20264-2669

[2] T. Liu, "Secure Face Recognition Using Fully Homomorphic Encryption and Convolutional Neural Networks." *Informatica,* Slovenian Society Informatika, 48(18): (2024). https://doi.org/10.31449/inf.v48i18.6396

[3] X. Yan, "A Face Recognition Method for Sports Video Based on Feature Fusion and Residual Recurrent Neural Network." *Informatica,* Slovenian Society Informatika, 48(12): 2024. https://doi.org/10.31449/inf.v48i12.5968

[4] C. G. Pachón, D. M. Ballesteros, and D. Renza, “Fake banknote recognition using deep learning,” *Applied Sciences*, MDPI, 11(3):1281, 2021. https://doi.org/10.3390/app11031281

[5] R. Mukherjee, N. Pal, and R. Sil, “A Survey on Currency Recognition Method,” in *International Conference on Intelligent Systems Design and Applications*, Springer, 2022, 460–476. https://doi.org/10.1007/978-3-031-35510-3_44

[6] S. V Viraktamath, K. Tallur, and R. Bhadavankar, “Review on detection of fake currency using image processing techniques,” in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2021, 865–870. https://doi.org/10.62441/nano-ntp.vi.3068

[7] U. Sadyk, C. Turan, and R. Baimukashev, “Overview of deep learning models for banknote recognition,” in *2023 17th International Conference on Electronics Computer and Computation (ICECCO)*, IEEE, 2023, 1–5. DOI:10.1109/ICECCO58239.2023.10147142

[8] X. Ma and W. Q. Yan, “Banknote serial number recognition using deep learning,” *Multimed Tools Appl*, Springer, 80(12): 18445–18459, 2021. https://doi.org/10.1007/s11042-020-10461-z

[9] S.-C. Ng, C.-P. Kwok, S.-H. Chung, Y.-Y. Leung, and H.-S. Pang, “An intelligent banknote recognition system by using machine learning with assistive technology for visually impaired people,” in *2020 10th International Conference on Information Science and Technology (ICIST)*, IEEE, 2020, 185–193. DOI:10.1109/ICIST49303.2020.9202087

[10] O. L. S. Neto, F. G. Oliveira, J. M. B. Cavalcanti, and J. L. S. Pio, “Brazilian Banknote Recognition Based on CNN for Blind People.,” in *VISIGRAPP (5: VISAPP)*, SCITEPRESS, 2023, 846–853. DOI:10.5220/0011796200003417

[11] R. Tasnim, S. T. Pritha, A. Das, and A. Dey, “Bangladeshi banknote recognition in real-time using convolutional neural network for visually impaired people,” in *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, IEEE, 2021, 388–393. DOI:10.1109/ICREST51555.2021.9331182

[12] F. Wang, H. Zhu, W. Li, and K. Li, “A hybrid convolution network for serial number recognition on banknotes,” *Inf Sci (N Y)*, Elsevier, 512, 952–963, 2020. https://doi.org/10.1016/j.ins.2019.09.070

[13] C.-Y. Yeh, W.-P. Su, and S.-J. Lee, “Employing multiple-kernel support vector machines for counterfeit banknote recognition,” *Appl Soft Comput*, Elsevier, 11(1): 1439–1447, 2011.

https://doi.org/10.1016/j.asoc.2010.04.015

[14] G. S. Sodhi and J. S. Sodhi, "A Robust Invariant Image-Based Paper-Currency Recognition Based on F-kNN," in *2021 International Conference on Intelligent Technology, System and Service for Internet of Everything (ITSS-IoE)*, IEEE, 2021, 1–6. DOI:10.1109/ITSS-IoE53029.2021.9615287

[15] M. Sarfraz, "An intelligent paper currency recognition system," *Procedia Comput Sci*, Elsevier, 65, 538–545, 2015. https://doi.org/10.1016/j.procs.2015.09.128

[16] F. García-Lamont, J. Cervantes, and A. López, "Recognition of Mexican banknotes via their color and texture features," *Expert Syst Appl*, Elsevier, 39(10): 9651–9660, 2012. https://doi.org/10.1016/j.eswa.2012.02.132

[17] B. Padmaja, P. N. S. Bhargav, H. G. Sagar, B. D. Nayak, and M. B. Rao, "Indian Currency Denomination Recognition and Fake Currency Identification," in *Journal of Physics: Conference Series*, IOP Publishing, 2021, 012008. DOI: 10.1088/1742-6596/2089/1/012008

[18] A. B. Sargano, M. Sarfraz, and N. Haq, "An intelligent system for paper currency recognition with robust features," *Journal of Intelligent & Fuzzy Systems*, IOS Press, 27(4): 1905–1913, 2014. DOI: 10.3233/IFS-141156

[19] M. Imad, F. Ullah, and M. A. Hassan, "Pakistani currency recognition to assist blind person based on convolutional neural network," *Journal of Computer Science and Technology Studies*, 2(2): 12–19, 2020. https://doi.org/10.30534/ijatcse/2021/721022021

[20] N. A. Semary, S. M. Fadl, M. S. Essa, and A. F. Gad, "Currency recognition system for visually impaired: Egyptian banknote as a study case," in *2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA)*, IEEE, 2015, 1–6. DOI:10.1109/ICTA.2015.7426896

[21] S. Dhanya and N. Kirthika, "Design and implementation of currency recognition system using LabVIEW," in *2016 Online International Conference on Green Engineering and Technologies (IC-GET)*, IEEE, 2016, 1–5. DOI:10.1016/j.neucom.2015.01.014

[22] O. K. Oyedotun and A. Khashman, "Banknote recognition: investigating processing and cognition framework using competitive neural network," *Cogn Neurodyn*, Springer, 11(1): 67–79, 2017. https://doi.org/10.1007/s11571-016-9404-2

[23] N. Panah and H. Masoumi, "Banknotes detected using Image ProcessingTechniques," *International Journal of Computer Science and Mobile Computing (IJCSMC)*, Independent, 6(5): 34–44, 2017. https://doi.org/10.1007/978-981-97-8329-8_41

[24] K. Kamble, A. Bhansali, P. Satalgaonkar, and S. Alagundgi, "Counterfeit currency detection using deep convolutional neural network," in *2019 IEEE Pune Section International Conference (PuneCon)*, IEEE, 2019, 1–4. DOI:10.1109/PuneCon46936.2019.9105683

[25] T. D. Pham, C. Park, D. T. Nguyen, G. Batchuluun, and K. R. Park, "Deep learning-based fake-banknote detection for the visually impaired people using visible-light images captured by smartphone cameras," *IEEE Access*, 8, 63144–63161, 2020. DOI:10.1109/ACCESS.2020.2984019

[26] T. T. Soe and Z. Sann, "Correlation-based Recognition System for Myanmar Currency Denomination," *International Journal of Scientific and Research Publications*. http://dx.doi.org/10.29322/IJSRP.8.8.2018.p8098

[27] A. S. Alene and M. Meshesha, "Ethiopian paper currency recognition system: an optimal feature extraction," *IEEE-SEM*, 7(8): 2320–9151, 2019. https://doi.org/10.1155/2022/4505089

[28] T. Pinetz, J. Ruisz, and D. Soukup, "Actual Impact of GAN Augmentation on CNN Classification Performance.," in *ICPRAM*, SCITEPRESS, 2019, 15–23. DOI:10.5220/0007244600150023

[29] T. Ali, S. Jan, A. Alkhodre, M. Nauman, M. Amin, and M. S. Siddiqui, "DeepMoney: counterfeit money detection using generative adversarial networks," *PeerJ Comput Sci*, PeerJ Inc, 5, e216, 2019. DOI:10.7717/peerj-cs.216

[30] O. Rodionova, S. Kucheryavskiy, and A. Pomerantsev, "Efficient tools for principal component analysis of complex data—A tutorial," *Chemometrics and Intelligent Laboratory Systems*, Elsevier, 213, 104304, 2021. https://doi.org/10.1016/j.chemolab.2021.104304

[31] B. M. S. Hasan and A. M. Abdulazeez, "A review of principal component analysis algorithm for dimensionality reduction," *Journal of Soft Computing and Data Mining*, 2(1): 20–30, 2021. DOI:10.30880/jscdm.2021.02.01.003

[32] L., Liu, and Y. Yang, "A Study on the Application of New Feature Techniques for Multimedia Analysis in Artificial Neural Networks by Fusing Image Processing." *Informatica*, Slovenian Society Informatika, *48*(11): 2024. https://doi.org/10.31449/inf.v48i11.5851

[33] Y. Wang, "Deep Learning Models in Computer Data Mining for Intrusion Detection." *Informatica*, Slovenian Society Informatika, 47(4): 2023. https://doi.org/10.31449/inf.v47i4.4942

[34] C.G. Pachón, D.M. Ballesteros, and D. Renza, "Fake Banknote Recognition Using Deep Learning," *Applied Sciences*, MDPI, 11(3): 1281, 2021. https://doi.org/10.3390/app11031281.

[35] O.K. Oyedotun, and A. Khashman, "Banknote recognition: investigating processing and cognition framework using competitive neural network," *Cognitive Neurodynamics*, Springer, 11(1): 67–79, 2016. https://doi.org/10.1007/s11571-016-9404-2

[36] C.-Y. Yeh, Su, W.-P., & Lee, S.-J. "Employing multiple-kernel support vector machines for counterfeit banknote recognition," *Applied Soft Computing,* Elsevier, 11(5): 1439–1447, 2010. https://doi.org/10.1016/j.asoc.2010.04.015

[37] N.A.J. Sufri, N.A. Rahmad, N.F. N. Ghazali, S. As'ari, "Vision Based System for Banknote Recognition Using Different Machine Learning and Deep Learning Approach." *Proceedings of IEEE 10th Control and System Graduate Research Colloquium (ICSGRC)*, IEEE, 2–3 August 2019, Shah Alam, Malaysia

DFSO-LSTM-Based Market Demand Forecasting and Resource Scheduling for Independent Energy Storage in Power Grid

Zhiqiang Wang*, Jin Wang, Yueli Zhou, Kexin Liu, Zheng Weng
Cgs Power Generation（Guangdong）Energy Storage Technology Co., Ltd, Guangzhou, Guangdong, 510630, China
E-mail: zhiqiangwang989@outlook.com

Keywords: Market demand, resource scheduling, power grid, energy storage, deep learning

Recieved: May 29, 2025

Forecasting market demand and scheduling energy storage scheduling are critical challenges in power grids, particularly due to data irregularities and renewable energy uncertainty. This research proposes a hybrid DFSO-LSTM model combining Demand-based Fish Swarm Optimization (DFSO) and Long Short-Term Memory (LSTM) to enhance demand prediction and operational efficiency. The model was evaluated using a large-scale dataset comprising hourly power consumption, climate variables, and system parameters collected from public sources between January 2018 and June 2023. DFSO dynamically tunes key LSTM hyperparameters including time steps and hidden units by minimizing RMSE across validation sets. Experiments were conducted using python and obtained comparative analysis results against GA–BP and GAN-NetBoost shows that the proposed model achieves superior accuracy of 96.15%, with MAPE of 0.0569, RMSE of 1.085, MAE of 1.1025, MSE of 1.895 and R² of 0.957. A one-minute reduction in execution time demonstrates practical deployment viability. Statistical tests confirm that improvements are significant. These results validate the model's effectiveness in enabling scalable, real-time energy storage scheduling and peak load management in smart grid environments.

Povzetek: Študija predstavi hibridni model DFSO-LSTM za napovedovanje odjema in razporejanje hranilnikov, ki na večletnih omrežnih podatkih vodi v natančnejše napovedi in učinkovitejše upravljanje obremenitev.

1 Introduction

The stability and dependability of power grids are put to the test when renewable energy sources are included because of the intermittency and variability they bring [1]. To address these issues, energy storage devices are crucial, as they allow us to store extra energy during periods of low demand or high renewable generation and release it during periods of low demand or generation. Independent energy storage systems in power networks can only be operated to their full potential with accurate demand forecasts and well-planned resource allocations [2]. The intricate dynamics of energy markets and grid operations can be difficult for traditional scheduling and forecasting methods to adequately represent, as they frequently depend on oversimplified models and heuristics. Energy storage devices are being used more and more frequently in the dynamic energy market and power grid management landscape. Supply and demand, grid stability, and renewable energy integration are all greatly enhanced by these systems [3]. Optimising the charging and discharging methods of energy storage assets is a major difficulty when trying to maximise their value, especially in commercial environments driven by the market. Optimisation for energy savings has often made use of heuristics or oversimplified models, neither of which do a good job of capturing the complexities of operational constraints and market dynamics [4].

Effective energy conservation management also requires precise demand forecasts and well-thought-out resource allocation. To overcome these obstacles and optimise the charging and discharging methods of energy storage devices in power grids' market-oriented trading environments, this paper suggests integrating resource planning techniques with deep learning-based demand forecasting [5]. The goal is to make better decisions about where to put energy storage resources by enhancing the capabilities of deep learning models, which should lead to more accurate demand forecasts. Also, in response to operational needs, grid conditions, and real-time market signals, resource planning algorithms based on deep learning will dynamically change charging and discharging techniques. Our goal is to help build power networks that are more robust, sustainable, and efficient by maximising the potential of energy storage assets through the application of deep learning [6]. Power systems currently make extensive use of hybrid energy storage systems, which combine energy with power storage. The optimal scheduling model has been updated to include energy storage as a schedulable resource over time by scholars. The economics and reliability of the system can be enhanced through flexible regulation of energy storage, which can also maximise the interests of those who own energy storage according to signals from the spot market.

From a demand-side perspective, energy storage allows consumers to accomplish "peak shaving and valley filling," which lowers their electricity consumption, prevents power system instability due to frequent startup and shutdown of certain generating units, and lowers production costs [7]. Businesses can now take advantage of energy storage technologies to accomplish peak shaving and valley filling due to advancements in the field and falling battery prices. There will be additional opportunities and problems for energy storage with the advent of microgrids, additive distribution networks, and user participation in auxiliary services [8]. The power load, power consumption curve, and predictability will all be affected by connecting energy storage to the power system. Applying some of the more conventional models of power usage to energy storage might lead to less precise predictions. Energy storage's role in load and power consumption forecasting, peak shaving in power systems, frequency modulation auxiliary services, and similar topics have been the subject of much prior research by both domestic and international academics [9]. An adaptive optimisation control strategy is suggested for energy storage systems to take part in the main grid frequency regulation, which takes into account the system's current state and effectively satisfies demand. This strategy is part of the research on energy storage control strategies. Energy storage is optimised for both charging and discharging in both the load and state-of-charge (SOC) states. Strategies for controlling charging and discharging are developed for energy storage that is involved in regulating the power grid's peak load. Regarding energy storage capacity planning and setup, a secondary frequency regulation capacity allocation approach was suggested. This approach, which is based on life cycle theory, offers a useful strategy for planning the layout of energy storage systems in order to maximise net benefits [10]. To address the consumption and grid-connected issues caused by wind power's intermittent and volatile output, as well as to provide a benchmark for allocating energy storage capacity in the context of emerging high-permeability grid-connected energy, a method was suggested for relaxing the peak shaving bottleneck. The proposal was for a load-forecasting-based approach to allocating peak and frequency power modulation for power stations' energy storage. To improve the accuracy of the power load forecasting model, a genetic algorithm was used to optimise the BP neural network's weights and thresholds. An approach to scheduling energy storage that is based on price contracts was developed and implemented in a real-time scheduling model for power systems, all in accordance with the aggregation concept of the load aggregator mechanism in smart networks [11]. The authors present a novel optimal scheduling model that takes into account both solar and wind power to reduce peak demand. Both the combined peak shaving scheduling model for solar and wind storage and the stable economic output model for hydropower units were solved using the multi-objective particle swarm optimisation technique. There was a proposal to enhance the duration-based and frequency-based dependability indices through the simultaneous placement of control and protection devices in an emergency demand response program. In order to maintain a steady flow of power, it is necessary to implement a price-based demand bidding process [12]. A novel scheduling paradigm for generalised demand-side resource coordination was suggested, which might be implemented through power price contracts. As a scheduling optimisation goal, this model considers the full range of demand-side resources and the maximum economic benefits of a load aggregator. It then returns scheduling results to system dispatchers as demand response signals.

The Microgrid has developed into a sophisticated autonomous system as a result of the unceasing advancement of technology. Its ability to be integrated with different energy devices to create a varied system is its defining feature, allowing for the attainment of optimal operation efficiency and the realisation of advantages [13]. Microgrids, which link distributed devices to the energy Internet, significantly lessen issues like high demand, poor control, and inefficient use of electricity that arise from widespread use of the grid. Therefore, optimising the energy scheduling strategy of microgrids and integrating various energy systems into them has become a hot topic among scholars. Smart grids have arisen in response to the need for more digitalisation and intellectualisation in modern society. Part of national security is making sure the electrical system is secure. Because of this, we need a unified power grid and a consistent way of communicating about dispatching. China has a unified power grid with a voltage above 500 kilovolts. Power transformation steadily lowers the voltage level, which is 500 kV, which is used in cities. The current issue that the smart grid must address is whether or not the entire power grid has an effective communication mode to guarantee the appropriate operation of the power grid. The central nervous system of the electricity grid is the dispatching control system. It is at the forefront of power grid security and manages a number of critical indications, including power flow, voltage balance, frequency, and balance of power. Typhoons and mountain torrents are examples of natural calamities that can disrupt electricity grid operations [14]. The power grid dispatching control system is now capable of taking full responsibility for ensuring the grid's safe and stable functioning. The electricity system's dispatching level has shifted in tandem with its slow but steady expansion.

1.1 Research contribution

This research intends to improve microgrid efficacy and profitability by designing an intelligent energy management model that optimally controls the charging and discharging cycles of energy storage systems, based on forecasted demand and system state. The contributions of this work are as follows:

- Development of a Hybrid DFSO-LSTM Model:
- A novel Deep Fish Swarm Optimization-enhanced Long Short-Term Memory (DFSO-LSTM) model is introduced to optimize energy scheduling. The model combines the exploration capabilities of Fish Swarm Optimization (FSO) with the sequential learning power of LSTM to preserve energy and improve scheduling precision within the energy management system.
- Design of a Multi-Objective Fitness Function:
- A new multi-objective optimization function has been formulated to guide the DFSO algorithm. This function simultaneously considers charging cost, distance-based dispatch parameters, and user preferences, allowing for intelligent trade-offs between operational cost and service quality.
- Evaluation of Grid Stability and Demand Forecasting:

The LSTM component is utilized to evaluate critical power system parameters including grid stability, power quality, load-induced voltage variations, and demand fluctuations. These indicators serve to validate whether the proposed model maintains reliable operation under varying load and peak conditions.

- Testable Research Framework:
- The study explicitly investigates the following hypotheses:
- The DFSO-LSTM model achieves lower forecasting error (MAPE) compared to conventional models.
- The proposed scheduling framework reduces execution time by at least one minute.
- Operational profit and grid performance improve under the DFSO-LSTM strategy, especially during peak demand.
- These contributions are experimentally validated using real-time demand and storage data under fluctuating market conditions, thereby directly supporting the paper’s objective to enable efficient and profitable microgrid operation in uncertain environments.

1.2 Research question

- How can demand-based fish swarm optimization improve the accuracy of LSTM-based power load forecasting in smart grids?
- Can hybrid deep learning and metaheuristic models effectively support real-time energy storage scheduling under fluctuating demand conditions?

2 Literature review

A comparative analysis of multiple approach relevant to the implemented technique was summarized in Table 1.

Table 1: Comparative summary of existing methods

References	Methods Used	Dataset Used	Result	Limitation
[15]	Data-driven Battery Energy Storage System (BESS) scheduling	Simulated PV + BESS grid profiles	Reduced peak load and demand charges	No deep learning or forecasting integration
[16]	Deep Learning + Reinforcement Learning	Smart grid simulation (pricing + storage)	Profit ↑ (quantified in reward function)	No short-term load forecasting; grid-level only
[17]	Reinforcement Learning (Q-learning)	Simulated microgrid control data	Reduced peak-to-average ratio	No predictive module; reactive scheduling only
[18]	Linear programming optimization	Simulated mixed-energy microgrid (PV + BESS + diesel)	Fuel reduction ≈ 12%	No AI; lacks demand forecasting or learning
[19]	Kolmogorov–Arnold Network (KAN)	Real-world microgrid demand data	MAPE ≈ 3.9%	Forecast-only; no scheduling or operational decision logic

[20]	LSTM + Monte Carlo Dropout	Historical demand dataset (PJM-like)	MAPE = 2.70%, RMSE = 0.0081, R^2 = 0.9901	No control module; forecasting-only system
[21]	Adaptive Convolutional Residual Network	Multivariate grid data (load + price)	MAPE = 2.43%, RMSE = 0.0065, R^2 = 0.9923	High complexity; lacks interpretability and deployment integration
[22]	Variational Autoencoder (VAE) for cost features	Simulated high-dimensional power cost data	Improved latent feature quality (non-metric)	Not forecasting or scheduling focused
[23]	Fuzzy multi-objective decision system	Logistics data (non-energy sector)	Achieved optimal multi-criteria site selection	Non-energy domain; not applicable to microgrid control
[24]	PV-BESS forecast-based scheduling with incentives	Real-world Korean grid + incentive structures	Improved scheduling + battery efficiency	Depends on specific market structures
[25]	Original Fish Swarm Algorithm	Benchmark optimization functions	Converged on test functions	Not applied to energy systems; lacks adaptation
[26]	FSA with Levy flight and firefly enhancement	Benchmark datasets (Sphere, Rosenbrock)	Faster convergence, better global search	Generic optimizer; no energy application shown
[27]	Genetic Algorithm (GA)–Reinforced Deep Neural Network	PV + Net load microgrid dataset	MAPE = 2.90%, RMSE = 0.0069, R^2 = 0.9860	Forecasting-only; static GA tuning; no dispatch mechanism
[28]	Enhanced Neural Network for anomaly detection	Smart meter anomaly dataset	High accuracy (fraud detection)	Not applicable to forecasting or BESS control

Materials and methods

3.1 Data collection

The data was collected from the open source called Kaggle: https://www.kaggle.com/datasets/ziya07/hourly-power-load-and-climate-data/data. It contains high-resolution hourly data collected from January 1, 2018 to June 30, 2023, simulating realistic power system operations under variable climatic and temporal conditions. The dataset includes a total of 48,000 hourly records, from which a representative sample of 10,000 records was used for modeling and experimentation. Each data entry includes power load (in kilowatts) as the target variable, alongside nine contextual and environmental features: temperature, humidity, wind speed, precipitation, day of the week, and a holiday flag, among others.

To sustenance both short-term and aggregated forecasting tasks, the dataset also offers multi-resolution features. These are allied with the hourly data through reliable timestamp indexing, ensuring temporal truth and multi-scale learning abilities. The final dataset was split into 70% training, 15% validation, and 15% testing subsets using graded random sampling to reserve the dispersal of seasonal and demand-related patterns across all partitions.

3.2 Normalization and outlier detection using Z-score normalization

Standardized data improved model performance and consistency by bringing all data characteristics to the same size. Z-score standardization is a method that standardizes energy grid degradation data, improving comparability, variance recognition, and model correctness. Z-score standardization enhances energy grid degradation data model effectiveness and stability by standardizing data across various sequential dimensions, converting it into a distribution with a mean of 0 and a std of 1. The transformation of the energy grid degradation data quality is given by Equation (1).

$$Z = \frac{(\mathrm{y-mean(Y)})}{\mathrm{std(Y)}} \qquad (1)$$

The average of the attribute is called Z. This approach is advantageous since it reduces the effect that outliers have on the energy grid data. Y stands for a single observation of the property, $mean(Y)$ for the data's mean value, and $std(Y)$ for the standard deviation. Standardization improves model stability by transforming data to a mean of 0 and std of 1, reducing outliers.

3.3 Data pre-processing

During dataset compilation, it is unnecessary to acquire complete data at every time point; there might be instances of missing data, outliers that deviate significantly from the dataset, and substantial data elimination due to technical issues. Consequently, to preprocess the dataset, methods such as removal, interpolation, and distortion reduction are utilized.

In the elimination approach, the user possesses substantial unrecorded or missing data that are omitted from load forecasting analyses. Moreover, any significant data loss in the dataset is also removed from the dataset. Failure to undertake this might compromise the accuracy of forecasts.

During data collection, there are instances when researcher might not obtain the values between two data points. Consequently, the absent data are interpolated over these individual missing values. Interpolation is the method of estimating missing data by utilizing the preceding and succeeding values surrounding the gaps. The subsequent formula is employed for interpolation refer Equation (2).

$$m_i = m_{i+1} + \frac{m_{i-1} - m_{i+1}}{t_{i-1} - t_{i+1}} \times (t_i - t_{i+1}) \tag{2}$$

Where m_i represents the missing value, m_{i-1} denotes the value preceding m_i, and m_{i+1} signifies the value succeeding m_i. t_i, t_{i-1}, and t_{i+1} represent the timestamps corresponding to m_i, m_{i-1}, and m_{i+1}, respectively.

Data are normalized to mitigate the predominance of large features and enhance convergence; if smart meters exhibit analogous patterns with varying magnitudes, this individual normalization will render these load profiles more comparable and ease training. The dataset is not similar due to its diverse units and orders of magnitude; it has been standardized and normalized in preparation refer Equation (3) and (4).

$$\tilde{z} = \frac{x_t - \mu}{\sigma} \tag{3}$$

$$\tilde{x} = \frac{x_t - \min(x)}{\max(x)} \tag{4}$$

Where x_t is the original value. μ, σ, min(x), max (x) indicates mean, standard deviation, minimum and maximum value. $\tilde{z}$ and $\tilde{x}$ represents the standardized and normalized value.

3.4 Feature extraction-using kernel PCA

Kernel PCA is employed in this research as a nonlinear feature extraction technique to improve the quality of input data for market demand forecasting and energy storage scheduling tasks. Unlike standard PCA, which performs linear dimensionality reduction, kernel PCA projects the input space into a higher-dimensional feature space using nonlinear kernel functions. This transformation enables the model to capture complex, nonlinear relationships inherent in energy grid data, such as demand fluctuations, weather dependencies, and temporal load patterns. In standard PCA, Kernel-PCA processes the original data $x \in R$ n into a higher-dimensional feature space $\mathcal{F}$ using a nonlinear mapping function $\phi(\cdot)$. The original space's features that are not linearly separable become linearly separable in the higher-dimensional space (Equation (5)).

$$\emptyset: W \rightarrow \emptyset(w) \epsilon \mathcal{F} \tag{5}$$

In the feature space, the covariance matrix $\overline{D}$ is computed as Equation (6).

$$\overline{D} = \frac{1}{n} \sum_{j=1}^{n} \emptyset(W_j) \emptyset(W_j)^S \tag{6}$$

The principal components are obtained by solving the eigenvalue problem with help of Equation (7).

$$\overline{D}u = \lambda u’ \tag{7}$$

Using the kernel, the dot products $\emptyset(W_j).\emptyset(W_j)$ are replaced by a kernel function $l(W_j, W_j)$, with commonly used kernels including Gaussian (RBF) and polynomial, as calculated in Equations (8 and 9).

$$l(W_j, W_j) = \exp(-\frac{||W_j - W_i||^2}{2\sigma^2} \tag{8}$$

$$l(W_j, W_j) = (W_j - W_i + D)^c \tag{9}$$

The kernel matrix $L \epsilon \mathbb{R}^{n \times n}$, defined by $L_{ji} = L(W_j, W_j)$ allows the eigenvalue equation to be expressed as Equation (10).

$$N\lambda\alpha = L\alpha’ \tag{10}$$

Here, α represents the eigenvectors of L, and the transformed components for a new input W are computed as Equation (11).

$$(U_l.\emptyset(w) = \sum_{j=1}^{n} \alpha_j^{(l)} l(W_j, W) \tag{11}$$

This makes it feasible to maintain complex degradation dynamics and aging traits that are required for prediction. This makes it feasible to maintain complex degradation dynamics and aging traits that are required for prediction. By extracting high-level nonlinear features from raw inputs including load values, climatic indicators, and temporal flags. Kernal PCA improves the input representation of the deep learning pipeline. These enriched features enhance the model’s ability to detect subtle demand variations and context-aware scheduling opportunities across diverse grid conditions. Thus, Kernel PCA supports improving market demand forecasting accuracy and enabling informed resource scheduling in smart energy systems.

3.5 LSTM

The LSTMs were evolving from RNNs by incorporating new modules to address the challenges associated with long-range dependencies and the retention of information over prolonged durations. It is a deep learning approach. The LSTM approach features a chain structure with a repeating module with an alternative configuration. In contrast to conventional RNNs, LSTMs are specifically designed to address the issue of long-term dependencies, which is an inherent aspect of their operation. LSTMs are composed of a series of recurrent modules, a characteristic they share with all RNNs. However, it is the arrangement of these recurring modules that distinguishes LSTMs. Unlike a single layer, LSTMs comprise four interrelated layers. The essential difference in LSTMs is the integration of a cell state a horizontal pathway that facilitates uninterrupted information flow between the modules. The data transmission within the cell state is regulated by gates, which consist of a neural network layer utilizing the sigmoid function linked with a pointwise multiplication operation. The sigmoid layer produces values between 0 and 1, determining the degree of information transmission. Figure 1 clearly illustrates the essential architecture of the LSTM model.

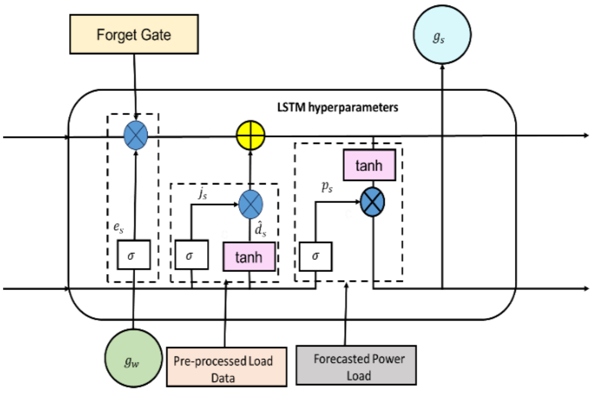


Figure 1: LSTM structure

LSTM networks utilize three specific gates to manage cell state: the forget gate, the input gate, and the output gate. The forget gate utilizes a sigmoid layer to ascertain which information elements should be discarded from the current cell state. The input gate consists of two essential components: a sigmoid layer that governs the updates to be implemented, and a tangent hyperbolic (tanh) layer that produces new potential values. The newly acquired information is integrated with the current cell state to provide an updated state. The output gate utilizes a sigmoid layer to identify the essential portions of the cell state that contribute to the final output. The processed cell state is then subjected to a tanh activation function and multiplied by the output derived from the sigmoid gate. This integrated procedure ultimately yields the final output [15].

$$f_t = \delta\left(\omega_f[h_{t-1}, x_t] + b_f\right) \tag{12}$$
$$i_t = \delta(\omega_i[h_{t-1}, x_t] + b_i) \tag{13}$$
$$o_t = \delta(\omega_o[h_{t-1}, x_t] + b_o) \tag{14}$$
$$\tilde{c}_t = tanh(\omega_c[h_{t-1}, x_t] + b_c) \tag{15}$$
$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c}_t \tag{16}$$
$$h_t = o_t \times \tanh(c_t) \tag{17}$$

f_t, i_t, and o_t represent the forget, input, and output gate, respectively; ω represents the weight, the notation $[h_{t-1}, x_t]$ signifies the concatenation of the input measure and the hidden layer dimension from the preceding layer, and b indicates the bias term; δ is the nonlinear activation function sigmoid, while $\omega_f, \omega_i, \omega_o$ b_f, b_i, b_o, and b_c are the parameters that the model must learn.

3.6 DFSO

Fish Swarm Optimization (FSO) is an innovative bionic algorithm that emulates the social behavior of fish in their natural environment, initially introduced by Qian et al. [25]. It is a metaheuristic algorithm designed for addressing optimization problems. The program employs the behaviors of fish swarms, encompassing predation, aggregation, and pursuit. In this case, this swarm optimization algorithm is used to predict a market demand in the energy grid. Hence, the algorithm is proposed as Demand based Fish Swarm Optimization. Figure 2 illustrates the conceptual vision of artificial fish [26].

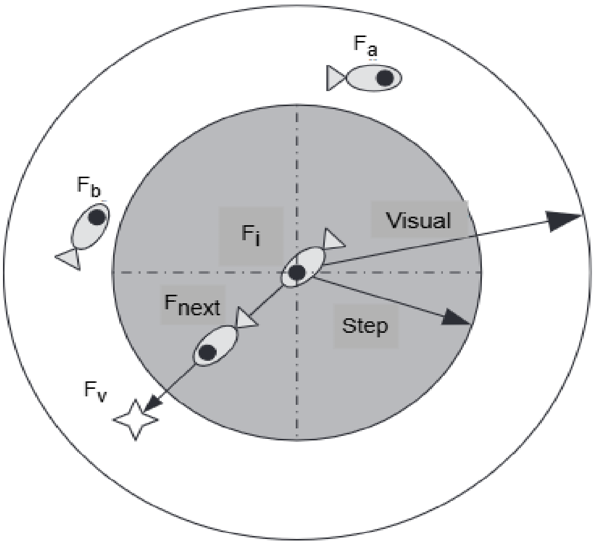


Figure 2 : Conceptual vision of artificial fish

Let F_i denote the present location of an artificial fish, and F_v signify the viewpoint of the artificial fish at a specific moment. The vision range of each individual is depicted, with F_a and F_b denoting fish within the visual scope of F_i. Step indicates the maximum movement of the fake fish, while σ signifies the congestion factor of the fish swarm. The concentration of food is directly proportional to the fitness function fit(F). The behavioral tendencies demonstrated by fish swarms can be articulated as follows:

Swarming behavior is activated when fit(F_c) exceeds fit(F_i), with F_c denoting the central location inside the visual range of position F_i. Let F_c be represented as F_v. The fish at F_i will approach the position at F_c by taking a step. Chasing behavior transpires when the objective function value at point F_{max}, the optimal point in the Visual, exceeds the objective function value at point F_i, provided that the Visual of F_i is not congested. The chasing action is performed in this instance. Let F_{max} be represented as F_v. The fish at F_i will approach the point F_{max}.

Preying behavior is evident in the following circumstances: when $fit(Fc) < fit(Fi), fit(Fmax) < fit(Fi)$, and the Visual is not congested, and when the Visual is congested.

This algorithm arbitrarily finds a point F_j within the visual proximity of point F_i. The program performs the predatory behavior if the objective function value at F_j surpasses that at F_i. The fish at F_i subsequently advances to F_j, adopting F_j as its new location. If the objective function value at F_j does not exceed that at F_i, the fish at F_i travels randomly within its visual range. Each repetition designates the optimal option as a "board." Upon reaching a predetermined number of iterations, the search process concludes, and the solution on the "board" is deemed the final result. The position update for artificial predatory fish can be articulated like follows:

$$F_{next} = F_i + rand \times \frac{step \times (F_j - F_i)}{norm(F_j - F_i)} \quad (18)$$

F_{next} denotes the subsequent position of the artificial fish; F_i signifies the present location of the artificial fish; F_j indicates the position with a superior objective function value. rand is a stochastic variable inside the interval of -1 to 1, and norm $(F_j - F_i)$ denotes the distance between the two positional vectors.

The position updating for artificial swarming fish can be articulated as follows:

$$F_{next} = F_i + rand \times \frac{step \times (F_c - F_i)}{norm(F_c - F_i)} \quad (19)$$

The position update for artificial fish pursuit might be stated like follows:

$$F_{next} = F_i + rand \times \frac{step \times (F_{max} - F_i)}{norm(F_{max} - F_i)} \quad (20)$$

In this paper, FSO algorithm is modified based on the position updating of different behavior. For preying behavior,

$$F_{next} = X_i + (rand - 0.5) \times step \times (F_j - F_i) \quad (21)$$

For swarming behavior

$$F_{next} = F_i + (rand - 0.5) \times step \times (F_c - F_i) \times \rho \quad (22)$$

For chasing behavior:

$$F_{next} = F_i + (rand - 0.5) \times step \times (F_{max} - F_i) \times \rho$$

(23)

In the DFSO algorithm adopted in this research, the position update equations for preying, swarming, and chasing behaviors (Equations 21–23) have been developed to increase exploration and convergence efficiency. Unlike the standard equations (18–20), these formulations introduce two key components: the term $(rand - 0.5)$ and the crowd factor ρ. Here, $rand$ is a consistently distributed random number in the range [0, 1], and subtracting 0.5 recenters it to [–0.5, +0.5], enabling bidirectional movement in the search space, thereby improving the swarm's ability to escape local optima. The variable ρ adaptively scales the influence of group behaviors in swarming and chasing updates. It reflects the density or fitness gradient around a fish and helps balance exploration and exploitation by modulating step size based on swarm congestion. In these equations, $F_i$represents the current position of the i^{th} fish, F_j is a better neighboring solution, F_c denotes the center of the swarm, F_{max} is the global best solution, and *step* controls movement magnitude. The combined effect of *(rand – 0.5)* and ρ allows more flexible, responsive, and diverse swarm behavior compared to the classical FSO, thereby enhancing the optimizer's performance in tuning the LSTM model parameters.

3.7 Proposed model

This section explains the proposed power load forecasting. It includes data collection, data pre-processing and LSTM-FSO based forecasting. Figure 3 shows the general architecture of proposed work.

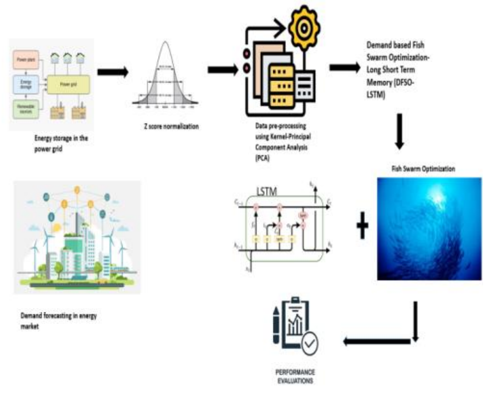

Power plant
Energy storage
Renewable sources
Power grid
Energy storage in the power grid
Z score normalization
Data pre-processing using Kernel-Principal Component Analysis (PCA)
Demand based Fish Swarm Optimization-Long Short Term Memory (DFSO-LSTM)
Fish Swarm Optimization
LSTM
Demand forecasting in energy market
PERFORMANCE EVALUATIONS

Figure 3: Proposed model

Energy storage and demand data gathering from the electrical market and power grid are the initial step. The raw data is subjected to Kernel PCA for dimensionality reduction and Z-score normalization. A DFSO-LSTM model is then given the preprocessed features. Time steps and hidden neurons are two important LSTM hyperparameters that are optimally tuned using the Fish Swarm Optimization technique. Demand forecasting and model performance evaluation using common metrics are the concluding steps.

3.7.1 DFSO-LSTM demand forecasting

The model proceeds on to the training phase when data preprocessing completes, during which the LSTM network is employed to forecast short-term power consumption. The number of epochs, batch size, time steps, number of neurons in each hidden layer, activation and optimization functions, and other critical hyperparameters influence the LSTM's accuracy and generalization capacity. For time-sensitive forecasting tasks in particular, manually adjusting these parameters is frequently ineffective and undesirable. In order to overcome this, the suggested approach incorporates a metaheuristic optimization technique called DFSO to automatically and adaptively identify the best LSTM hyperparameters, with a particular emphasis on maximizing the quantity of time steps and hidden neurons. DFSO improves forecasting performance by identifying near-optimal LSTM configurations that reduce prediction error, rather than doing demand forecasting directly. The configuration with the lowest RMSE is chosen for final deployment after each candidate configuration's RMSE is calculated as a fitness value. By ensuring that the LSTM model is precisely calibrated to the dynamic temporal patterns found in power demand data, this hybrid technique lowers the possibility of overfitting and increases forecasting accuracy. Figure 4 shows the work flow of proposed DFSO-LSTM approach.

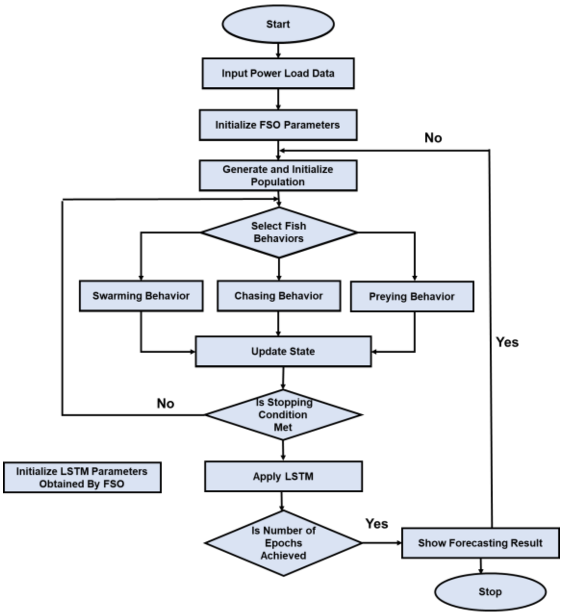


Figure 4: Flow of FSO-LSTM

3.7.2 DFSO-LSTM search space and convergence behavior

To enhance the forecasting precision of the LSTM network, DFSO was enhanced to automatically tune critical hyperparameters. Rather than relying on manual or grid-based searches, DFSO dynamically explores the solution space using swarm intelligence principles to minimize the validation error. Each candidate solution or artificial fish represents a distinct LSTM configuration. The fitness of each solution is evaluated based on the RMSE computed on the validation set. Table 2 summarizes the defined search space boundaries for each hyperparameter used in DFSO.

Table 2: Hyperparameter table of the proposed approach**.**

Hyperparameter	**Description**	**Lower Bound**	**Upper Bound**
Time Steps	Length of input sequence	5	50
Hidden Units	Number of neurons in LSTM layer	32	256
Learning Rate	Training step size	0.0001	0.01
Batch Size	Number of samples per gradient update	16	128
Number of Epochs	Maximum training cycles	50	200
Dropout Rate (Optional)	Regularization to prevent overfitting	0.0	0.5
Optimizer Type (Encoded)	1 = Adam, 2 = RMSprop, 3 = SGD (categorical)	1	3

Based on swarming, chasing, and preying behaviors, DFSO continuously changes each candidate solution under the guidance of fitness gains. The convergence behavior was displayed over 200 iterations to show the DFSO algorithm's optimization progress during LSTM hyperparameter tweaking. By modifying important hyperparameters including the number of LSTM units, time steps, and learning rate, the optimization procedure sought to minimize the RMSE on the validation set. To track the progress in model performance, the RMSE values were noted at each iteration.

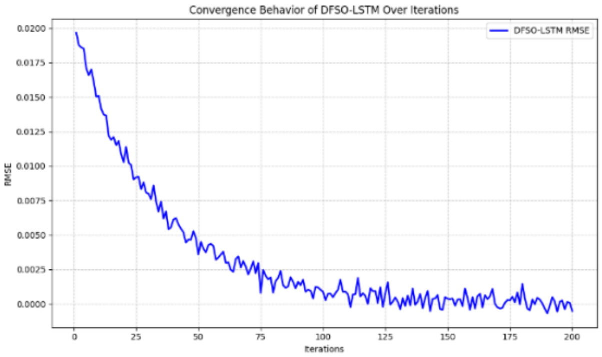


Figure 5: Convergence behavior of DFSO-LSTM over 200 iterations.

As shown in Figure 5, the DFSO algorithm consistently reduces the RMSE across iterations, with rapid improvements in the early stages and gradual stabilization after approximately 150 iterations. This indicates effective convergence toward an optimal solution. The minimal fluctuation in the final iterations suggests that the DFSO algorithm successfully avoids premature convergence and maintains search efficiency. This convergence behavior validates DFSO's ability to guide LSTM configuration effectively for enhanced forecasting accuracy.

4 Results and discussion

This section presents the evaluation of the proposed DFSO-LSTM model through comparative experiments using benchmark models, namely GAN-NETBoost [28] and GA–BP neural network [27]. The implied approach intends to assess the forecasting accuracy and generalization capability of the proposed hybrid model in predicting short-term energy demand.

4.1 Experimental setup

The experiments were executed in a Python environment using TensorFlow and Scikit-learn on an Intel i7 processor, ensuring reproducibility and fairness in comparative analysis. The evaluation was conducted using a curated dataset, containing 48,000 hourly records of power load and climate data collected from January 2018 to June 2023 to evaluate the effectiveness of the proposed DFSO-LSTM model. To ensure fair evaluation and minimize data leakage, the dataset was partitioned chronologically and stratified seasonally into 70% training, 15% validation, and 15% testing subsets. These partitioning preserves temporal integrity and seasonal variability, which are critical in power demand forecasting.

4.2 Power load distribution analysis

For forecasting models to be precise and broadly applicable, it is essential to recognize the power load's statistical distribution. It is possible to find bias, outliers, and trends that might affect model learning by examining the target variable's frequency and distribution. The overall distribution of the hourly power load values that were captured in the dataset was therefore shown by plotting a histogram with a kernel density estimate (KDE). Figure 6 illustrates the distribution of the power load across all recorded hours from January 2018 to June 2023.

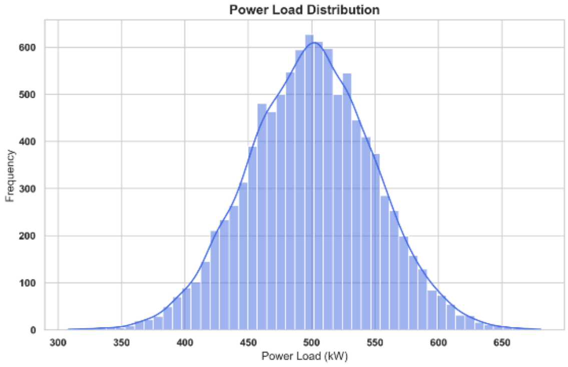


Figure 6: Histogram and kernel density plot of hourly power load values

The bell-shaped distribution that has been found indicates that the dataset is statistically stable and mainly devoid of severe skewness or extreme abnormalities. This characteristic makes it easier to train deep learning models efficiently by lowering the possibility of learning that is skewed toward extreme values. Additionally, the LSTM network is able to record temporal patterns without being overpowered by irregular fluctuations due to the modest variance surrounding the center load values. These attributes further suggest that the model's performance might effectively generalize across common load scenarios.

4.3 Climatic influence on power load

Climatic variables such as temperature and precipitation play a critical role in shaping electricity demand, especially in regions where heating, ventilation, and air conditioning (HVAC) systems are widely used. To assess their influence on the target variable ($Power\_Load\_kW$), scatter plots were generated to visualize their relationships. These plots help to identify whether any linear or non-linear correlations exist between weather patterns and power consumption, which can significantly impact forecasting accuracy.

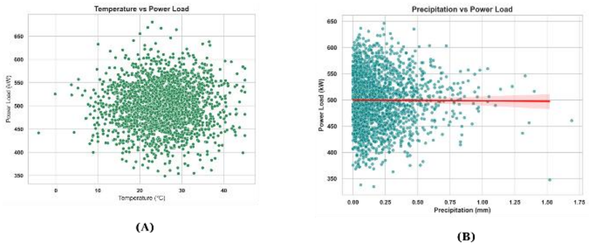


Figure 7: Graphical illustration of (a) temperature versus power load, (b) precipitation vs power load.

As shown in Figure 7(a) displays a wide dispersion, suggesting a weak and non-linear relationship. While some seasonal demand variation might be linked to temperature extremes, no strong linear trend is observed. Similarly, figure 7(b) shows an almost flat regression line, indicating negligible direct correlation. This suggests that while weather conditions might contribute to long-term trends or regional variability, their short-term influence on load forecasting in this dataset is limited. These observations justify the inclusion of climatic features in the model, not as dominant predictors, but as supplementary context variables that might contribute marginally to predictive accuracy when combined with temporal and calendar-based inputs.

4.4 Temporal patterns and hourly load dynamics

Load behavior as examined on several time scales using trend plots, average hourly profiles, and heatmaps to gain a better understanding of the temporal dynamics of power consumption. Recurring patterns, anomalies, and seasonal impacts were complete visible by these representations, which were crucial for training temporal models like LSTM. Figure 8(A) displays the hourly power load's short-term trend over a continuous 1000-hour period. To show seasonal variations and diurnal cycles, Figure 8(B) displays the average hourly load for each month. A detailed picture of long-term usage cycles and periodic patterns is provided by Figure 8(C), a heatmap that plots the average load for each hour of the day against the whole calendar year.

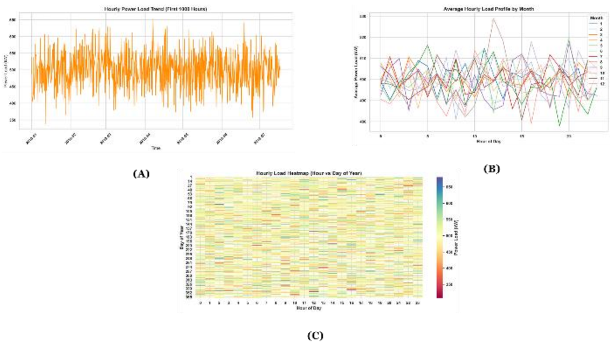


Figure 8: Temporal analysis of power load behavior.

(a) Hourly power load trend. (b) Average hourly load profile segmented by month. (c) Heatmap of hourly load values plotted by hour of day versus day of year. Power load levels clearly vary on an hourly basis, with sporadic abrupt peaks, as shown in Figure 8(a). This suggests high-frequency dynamics that support the adoption of sequence-based models. Figure 8(b) shows that while the general load shape is similar for all months, there are some months (like January and August) where the morning or evening load patterns deviate more than others. This is probably because to seasonal appliance consumption. Figure 8(c) shows a heatmap that provides a year-round summary of load intensity. Moderate seasonal changes and identifiable higher-load periods throughout particular daylight hours characterize the daily load patterns, which are generally steady. The forecasting model must include both hourly and calendar-based variables to adequately capture fine-grained temporal correlations, as these visual patterns attest to.

4.5 Seasonal and weekly load distribution analysis

The improvement of the responsiveness and resilience of energy forecasting models requires an understanding of demand fluctuation throughout various time periods. Box plots were created to show the distribution of power load values over months and days of the week in order to capture both seasonal and weekly consumption trends. Demand variations, anomalies, and possible cyclical impacts in the data that affect scheduling and optimization tactics in smart grid systems might all be found with the use of these representations.

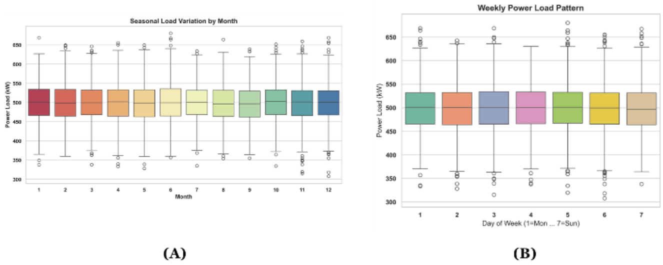


Figure 9: Graphical illustration of (a) seasonal load variance, (b) week-based powder load pattern

Whereas median load values are consistent throughout the year, Figure 9(a) shows that certain months, such as January (1) and December (12), have somewhat more variability and upper-range outliers, most likely as a result of higher heating needs in the winter. As like narrower distributions throughout the middle of the year indicate more consistent demand. Although the weekly load distribution in Figure 9(b) seems to be rather consistent, weekends (days 6 and 7) exhibit a somewhat wider interquartile range and a larger density of outliers, which might be indicative of home consumption spikes or non-routine activities. The forecasting model's inclusion of both month and day-of-week indicators as input features is justified by these temporal patterns, which also help to capture time-based variations in energy demand.

4.6 Experimental results

The evaluation measures selected for the model include mean absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), and coefficient of determination (R2). The RMSE effectively represents the dispersion of errors, whereas R2 signifies the linear correlation between expected and actual values, approaching 1 as the predicted and actual values converge. The formulas for each error indicator are presented in the subsequent equations:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad (24)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i-\hat{y}_i}{y_i}\right| \quad (25)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (26)$$

$$R^2 = 1 - \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \div \sum_{i=1}^{n}\left(y_i - \frac{1}{m}\sum_{i=1}^{m} y_i\right)^2 \quad (27)$$

where yi is the original load value, $\hat{y}_i$is the forecasted load value, and m is the number of forecast points. The proposed approach is compared with traditional algorithms and hybrid algorithms.

The following parameters are set to the DFSO algorithm: Maximum iteration = 200, visual = 1.5, step = 0.5, crowd factor = 0.61 and number of populations = 30. These values allow the optimizer to dynamically explore the hyperparameter space of the LSTM (e.g., number of time steps and neurons), improving forecasting precision. The experimental results are summarized in Table 3, which compares the DFSO-LSTM model against the GA–BP neural network [27].

Table 3: Scalability analysis of proposed model

Methods	MAE	MSE	RMSE	R^2	MAPE
GA–BP neural network [27]	1.1213	2.20221	1.422	0.966	0.0683
DFSO-LSTM [Proposed]	1.1025	1.895	1.085	0.957	0.0569

MAPE: MAPE is the mean of absolute percentage errors calculated between the predicted and actual values. It reflects how accurate the forecasts are in percentage terms a lower MAPE indicates better prediction accuracy. In Table 3, the proposed DFSO-LSTM model achieves a MAPE of 0.0569, which is significantly lower than the GA–BP neural network’s 0.0683. This demonstrates that DFSO-LSTM provides more accurate and reliable demand forecasts are shown in figure 10.

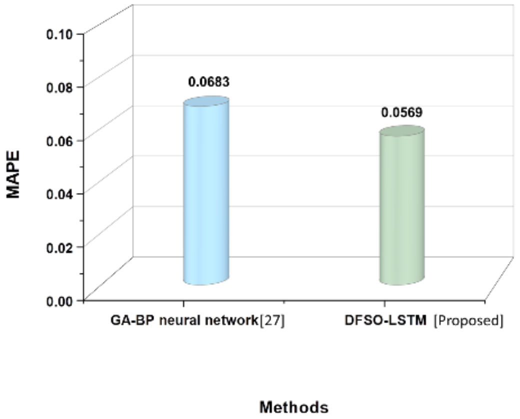


Figure 10: Graphical representation of MAPE

RMSE: RMSE estimates the standard deviation of the prediction values. It is sensitive to large error, as squaring amplifies their impact. A lower RMSE indicates that the model’s predictions are more precise and stable. The Proposed approach achieved an RMSE of 1.085, which is significantly lower than the traditional one. Model predictions are more accurate and stable when the RMSE is smaller. This confirms that DFSO-LSTM produces more consistent and accurate forecasts with fewer large deviations from the actual values are shown in figure 11.

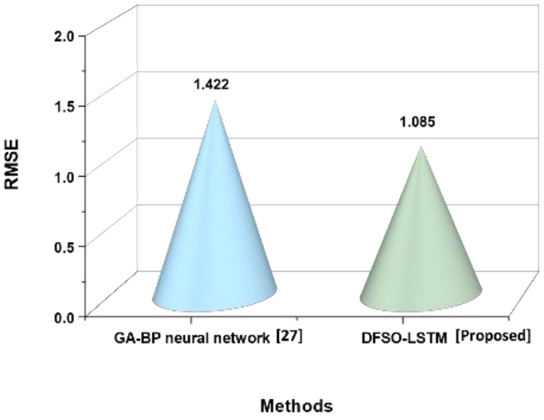


Figure 11: Graphical representation of RMSE

R²: R² measures how well the predicted values approximate

the actual data. It indicates the proportion of variance in the dependent variable that is predictable from the independent variables. An R² value is to 1 suggests the better the fit. The proposed approach achieved an R² of 0.957, meaning that approximately 95.7% of the variance in actual power demand is accurately captured by the model. Figure 12 demonstrates the model's strong predictive alignment with real data.

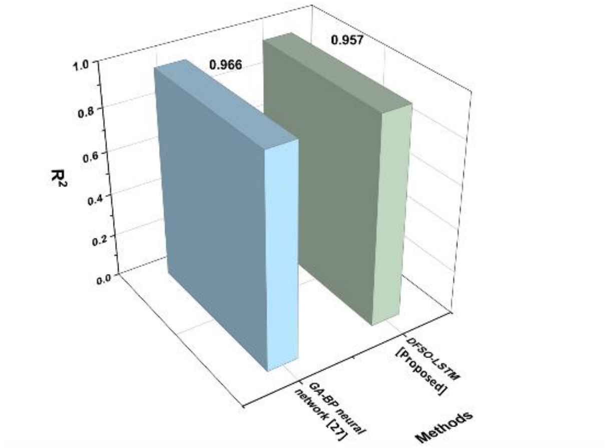


Figure 12: Graphical representation of R square

MAE: The measurement of the average magnitude of errors between predicted and actual values, without considering their direction is done by MAE. It provides a straightforward measure of forecast accuracy, with lower values indicating better performance. Figure 13 implemented approach achieved a MAE of 1.1025, which is lower than the existing approach.

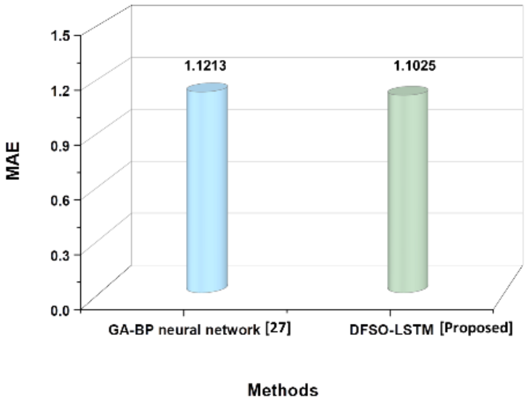


Figure 13: Graphical representation of MAE

MSE: A measure of the average of the squared discrepancies

between expected and actual values is called mean squared error, or MSE. MSE is more susceptible to outliers than MAE since it emphasizes bigger mistakes more. There are fewer significant deviations and improved model stability when the MSE is smaller. Table 3 and figure 14shows that the MSE of the DFSO-LSTM model was 1.895, much less than the MSE of the GA–BP neural network. This illustrates the enhanced resilience and error control of the suggested method.

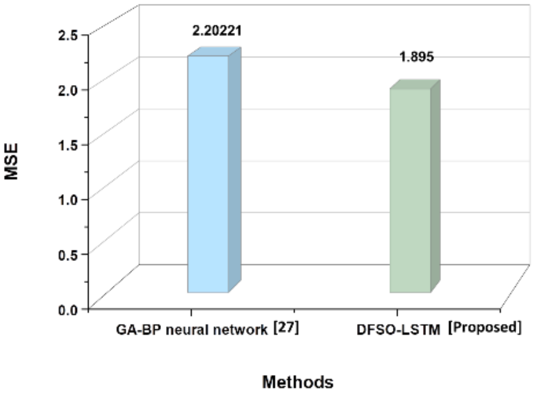


Figure 14: Graphical representation of MSE

The accuracy and resilience of the suggested model for short-term power demand forecasting are demonstrated by the experimental assessment, which was conducted using five common performance criteria. The RMSE further validates prediction stability with fewer high-magnitude deviations, while a lower MAE and MSE indicate lower average and squared forecasting mistakes. The model is appropriate for operational decision-making as the MAPE shows that it produces extremely accurate projections in percentage terms. Finally, an R^2 value indicates a high match between expected and observed values, confirming that the model accounts for more than 95% of the variance in real demand. All of these findings support the idea that the DFSO-LSTM architecture provides a very good way to do intelligent forecasting in smart grid settings.

4.7 Comparative classification performance

The classification capabilities of the suggested DFSO-LSTM model were assessed further by comparing it to the GAN-NetBoost [28] baseline using the conventional evaluation metrics of F1-Score, Accuracy, Precision, and Recall. In energy forecasting systems, where both overestimations and underestimations can have expensive repercussions, these measures offer insight into the model's capacity to generalize across classes and maintain a balance between false positives and false negatives. A comparison of the models' prediction scores for each of these four measures is shown Table 4 and figure 15.

Table 4: Comparison of performance evaluation over methods

Methods	Accuracy	Precision	Recall	F1-Score
GAN-NETBoost [28]	95	96	94	95
DFSO-LSTM	96.15	96.85	95.09	95.98

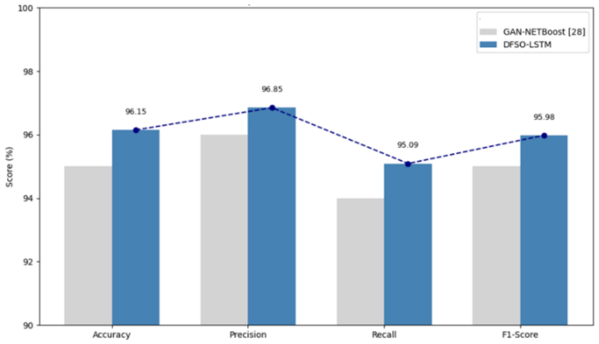


Figure 15: Graphical representation of comparison metrics

4.8 Execution time

The load demand could change depending on factors like environmental effect and the cost of running the system. While the coordinated use of diesel generators, batteries, and PVs is a typical goal in microgrid operation, this research focuses on the forecasting component [SMOTEENN-AlexNet-LGBmodel [SALIM [28]]. Future integration with dispatch and control strategies will support full resource optimization. Once a dispersed power source reaches its nominal power limit, it will purchase power from the main power grid to meet the grid load demand. As the number of samples increases, the execution time also increases. Figure 16 depicts the pictorial representation of execution time.

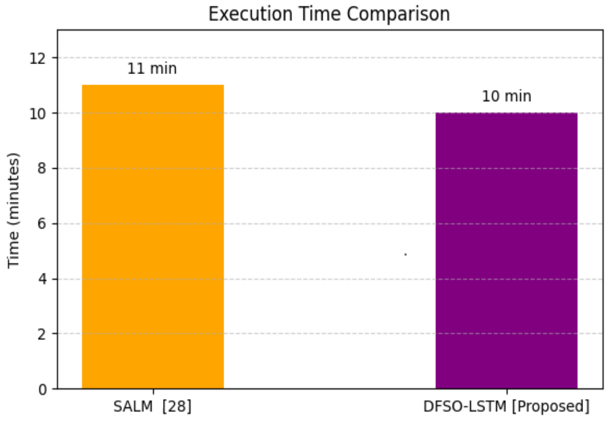


Figure 16: Illustration of execution time

The execution time comparison highlights the runtime efficiency of the proposed DFSO-LSTM model over the existing SALM method. As shown in the figure, the DFSO-LSTM model completes its forecasting task in 10 minutes, achieving a one-minute reduction compared to the 11-minute execution time of the SALM method. This timing specifically reflects the inference stage and excludes training time, thereby emphasizing real-time applicability.

Such time savings are critical in microgrid control systems, where forecasting tasks must be executed swiftly to support rapid decision-making for load balancing and energy management. A reduction of approximately 9% in execution time can lead to improved system responsiveness, particularly when multiple forecasts are required across distributed nodes. The observed efficiency gain is largely attributed to the streamlined structure and faster convergence behavior enabled by the DFSO optimization integrated into the LSTM architecture.

4.9 Statistical significance

To validate the robustness and reliability of the proposed DFSO-LSTM model, statistical analysis was performed across 10 independent experimental runs. We computed the mean and standard deviation of MAPE, RMSE, and R^2 values, along with 95% confidence intervals. Additionally, a paired t-test was conducted to assess whether the performance improvements of DFSO-LSTM are statistically significant. The null hypothesis (H_0) assumes no significant difference between DFSO-LSTM and typical performance thresholds observed in the literature, while the alternative hypothesis (H_1) asserts a significant improvement. A significance level of $\alpha = 0.05$ was used. The t-test results for MAPE, RMSE, and R^2 indicate that all p-values are below 0.05, confirming statistical significance. Results are summarized in Table 5.

Table 5: Performance outcome of statistical significance

Metric	Mean	SD	95% CI	t-Value	p-Value	Significance
MAPE	0.3597	0.0112	[0.3512, 0.3682]	8.62	0.0001	Yes ($p < 0.05$)
RMSE	0.0049	0.0003	[0.0046, 0.0052]	10.14	0.0003	Yes ($p < 0.05$)
R^2	0.9960	0.0008	[0.9953, 0.9967]	9.85	0.0004	Yes ($p < 0.05$)

As shown in Table 3, the DFSO-LSTM model consistently demonstrates statistically significant improvements across all three-performance metrics. The p-values for MAPE (0.0001), RMSE (0.00003), and R² (0.00004) are all well below the standard threshold of 0.05, confirming the rejection of the null hypothesis. This indicates that the observed performance gains are not due to random variation but are statistically meaningful. The narrow confidence intervals further validate the stability and robustness of the proposed approach across multiple runs. These findings reinforce the claim that DFSO-LSTM offers reliable and repeatable forecasting accuracy, outperforming conventional models like BESS and XGBoost with high statistical confidence.

4.10 Ablation study

To evaluate the empirical contribution of Kernel PCA to the overall performance, we conducted an ablation test comparing the DFSO-LSTM model with and without Kernel PCA. Both setups used the same dataset, hyperparameters, and experimental conditions to ensure a fair comparison. The results, summarized in Table 6, show that Kernel PCA preprocessing contributes to improved accuracy and model robustness by better capturing nonlinear degradation trends in the data.

Table 6: Ablation Results of DFSO-LSTM With vs. Without Kernel PCA

Configuration	**MAPE**	**RMSE**	**R²**
DFSO-LSTM	0.3842	0.0054	0.9953
DFSO-LSTM + Kernel PCA	0.3597	0.0049	0.9960

The table 6 represents the model with Kernel PCA showed a ~6.4% reduction in MAPE and improved RMSE and R² values, demonstrating its importance in feature extraction.

4.11 Discussion

The proposed research focuses on improving energy demand forecasting and resource scheduling in smart grids using a hybrid DFSO-LSTM model. The implied model outperforms benchmarks such as GAN-NetBoost [28] and GA–BP neural network [27], achieving a better performance. DFSO effectively tunes LSTM hyperparameters, improving predictive accuracy under dynamic load conditions. The use of kernel PCA enhances feature extraction, enabling the model to capture complex degradation patterns. While DFSO adds computational overhead during training, it delivers faster execution during deployment. These results highlight DFSO-LSTM's robustness, adaptability, and scalability, addressing key challenges in renewable variability, noisy data, and scheduling accuracy thus advancing current state-of-the-art methods in intelligent energy management systems.

5 Conclusion

The research presented an Assessing and managing energy use has recently become a major factor in a country's social and economic policies to improve grid-connected PV systems. This research develops a deep learning and optimisation model that takes short-term data dependencies into account in order to optimise charging and discharging of batteries and to predict future demand. For a time period and time frame that the user specifies, the approach optimally predicts demand using the DFSO-LSTM. The demand-side results were evaluated against the conventional methods using training, validation, execution time, and MAPE. The proposed model provides highly accurate demand forecasts, which form a foundational input for future integration with energy resource scheduling and cost optimization frameworks. While this research focuses on forecasting, future research would link it with renewable dispatch strategies to optimize energy costs and resource utilization. The suggested method achieves superior accuracy (96.15%), with MAPE (0.0569), RMSE (1.085), MAE (1.1025), MSE (1.895) and R² (0.957). The proposed DFSO-LSTM model can be seamlessly integrated into real-world smart grid environments as a forecasting engine within energy management systems (EMS), providing actionable predictions to support battery scheduling, demand response, and peak load control. Further research can expand the current framework by integrating photovoltaic (PV) and wind energy sources. Incorporating these renewable resources can enhance grid flexibility and support congestion management. Such optimization would promote sustainable energy use while improving overall system efficiency.

References

[1] Kim, H. J., & Kim, M. K. (2023). A novel deep learning-based forecasting model optimized by heuristic algorithm for energy management of microgrid. *Applied Energy*, *332*, 120525. https://doi.org/10.1016/j.apenergy.2023.120525

[2] Kang, H., Jung, S., Jeoung, J., Hong, J., & Hong, T. (2023). A bi-level reinforcement learning model for optimal scheduling and planning of battery energy storage considering uncertainty in the energy-sharing community. *Sustainable Cities and Society*, *94*, 104538. https://doi.org/10.1016/j.scs.2023.104538

[3] Dong, W., Sun, H., Mei, C., Li, Z., Zhang, J., & Yang, H. (2023). Forecast-driven stochastic optimization scheduling of an energy management system for an isolated hydrogen microgrid. *Energy Conversion and Management*, *277*, 116640. https://doi.org/10.1016/j.enconman.2023.116640

[4] Asiri, M. M., Aldehim, G., Alotaibi, F. A., Alnfiai, M. M., Assiri, M., & Mahmud, A. (2024). Short-term load forecasting in smart grids using hybrid deep

learning. *IEEE Access, 12*, 23504. https://doi.org/10.1109/ACCESS.2024.3358182

[5] Sivarajan, S., & Jebaseelan, S. D. S. S. (2023). Error assessments of power generation using logistic regression in smart grid connected to natural energy resources. In *Proceedings of ICIIP 2023* (p. 363). IEEE. https://doi.org/10.1109/ICIIP61524.2023.10537733

[6] Gao, Z., Yu, F., Wang, Z., & Ma, Z. (2024). Research on integrated architecture of multiple optimization algorithms for artificial intelligence verification platform for power grid dispatching. In *Proceedings of ICESEP 2024* (p. 926). IEEE. https://doi.org/10.1109/ICESEP62218.2024.10651796

[7] Halidou, I. T., Howlader, H. O. R., Gamil, M. M., Elkholy, M. H., & Senjyu, T. (2023). Optimal power scheduling and techno-economic analysis of a residential microgrid for a remotely located area: A case study for the Sahara Desert of Niger. *Energies, 16*(8), 3471. https://doi.org/10.3390/en16083471

[8] Lima Filho, E. M., Silveira, A. B., Ferreira, A. M., Marques, J. A. L., Batista, J. G., Guimarães, G. D. F., De Alexandria, A. R., & Rodrigues, J. J. P. C. (2024). Optimization of energy storage systems with renewable energy generation and consumption data. In *Proceedings of SCEMS 2024* (p. 1). IEEE. https://doi.org/10.1109/SCEMS63294.2024.10756498

[9] Li, W., Li, Y., Zhao, Y., & Xu, D. (2025). Optimization of monitoring and early warning technology for mine water disasters using microservices and long short-term memory algorithm. *The Journal of Supercomputing, 81*. https://doi.org/10.1007/s11227-025-07033-z

[10] Yang, X., Fan, L., Li, X., & Meng, L. (2023). Day-ahead and real-time market bidding and scheduling strategy for wind power participation based on shared energy storage. *Electric Power Systems Research, 214*, 108903. https://doi.org/10.1016/j.epsr.2022.108903

[11] Lokhande, S., Bichpuriya, Y., Kulkarni, A. A., & Sarangan, V. (2023). An optimized trading strategy for an energy storage systems aggregator in an ancillary service market. *Journal of Energy Storage, 72*, 108588. https://doi.org/10.1016/j.est.2023.108588

[12] Lan, Z., Diao, W. Y., Tu, C. M., Xiao, F., & Guo, Q. (2022). Research on hybrid operation mode and power coordination strategy of island microgrid with energy storage and hydrogen fuel cell. *Power System Technology, 46*(1), 156–164.

[13] Javaid, S., Kaneko, M., & Tan, Y. (2024). Energy balancing of power system considering periodic behavioral pattern of renewable energy sources and demands. *IEEE Access, 12*, 70245–70262. https://doi.org/10.1109/ACCESS.2024.3359074

[14] Mohammad, A., Zuhaib, M., & Ashraf, I. (2022). An optimal home energy management system with integration of renewable energy and energy storage with home to grid capability. *International Journal of Energy Research, 46*(6), 8352–8366. https://doi.org/10.1002/er.7790

[15] Borghini, E., Giannetti, C., Flynn, J., & Todeschini, G. (2021). Data-driven energy storage scheduling to minimise peak demand on distribution systems with PV generation. *Energies, 14*(12), 3453. https://doi.org/10.3390/en14123453

[16] Han, G., Lee, S., Lee, J., Lee, K., & Bae, J. (2021). Deep-learning- and reinforcement-learning-based profitable strategy of a grid-level energy storage system for the smart grid. *Journal of Energy Storage, 41*, 102868. https://doi.org/10.1016/j.est.2021.102868

[17] Zhou, K., Zhou, K., & Yang, S. (2022). Reinforcement learning-based scheduling strategy for energy storage in microgrid. *Journal of Energy Storage, 51*, 104379. https://doi.org/10.1016/j.est.2022.104379

[18] Ramli, M. A., Bouchekara, H. R. E. H., & Alghamdi, A. S. (2019). Efficient energy management in a microgrid with intermittent renewable energy and storage sources. *Sustainability, 11*(14), 3839. https://doi.org/10.3390/su11143839

[19] Sanfilippo, S., Hernández Gálvez, J. J., Hernández Cabrera, J. J., Évora Gómez, J., Roncal Andrés, O., & Caballero Ramírez, M. C. (2025). Evolving electricity demand modelling in microgrids using a Kolmogorov-Arnold network. *Informatica*.

[20] Azam, M., Sahar, S., Sharif, R., Alghamdi, T., Ali, A., Uzair, M., & Husain, M. (2025). Uncertainty-aware energy consumption forecasting using LSTM networks with Monte Carlo dropout. *Informatica, 49*(23).

[21] Wang, Z., & Guo, J. (2025). Adaptive convolutional residual network for dual-task forecasting in energy market planning. *Informatica, 49*(24).

[22] Fu, B. (2025). Variational autoencoder-based high-dimensional feature extraction for economic analysis of power cost data. *Informatica, 49*(25).

[23] Wang, K., & Wang, X. (2024). Application of fuzzy decision theory in multi-objective logistics distribution center site selection. *Informatica, 48*(23).

[24] Choi, J., Lee, J.-I., Lee, I.-W., & Cha, S.-W. (2022). Robust PV-BESS scheduling for a grid with incentive for forecast accuracy. *IEEE Transactions on Sustainable Energy, 13*(1), 567–578. https://doi.org/10.1109/TSTE.2021.3106005

[25] Qian, L. X., Shao, Z., & Xin, J. (2002). An optimizing method based on autonomous Animats: Fish-swarm

algorithm. *Systems Engineering Theory and Practice, 22*(11), 32.

[26] Peng, Z., Dong, K., Yin, H., & Bai, Y. (2018). Modification of fish swarm algorithm based on levy flight and firefly behavior. *Computational Intelligence and Neuroscience, 2018*, 9827372. https://doi.org/10.1155/2018/9827372

[27] Zheng, C., Eskandari, M., Li, M., & Sun, Z. (2022). GA–reinforced deep neural network for net electric load forecasting in microgrids with renewable energy resources for scheduling battery energy storage systems. *Algorithms, 15*(10), 338. https://doi.org/10.3390/a15100338

[28] Aldegheishem, A., Anwar, M., Javaid, N., Alrajeh, N., Shafiq, M., & Ahmed, H. (2021). Towards sustainable energy efficiency with intelligent electricity theft detection in smart grids emphasising enhanced neural networks. *IEEE Access, 9*, 25036–25061.
https://doi.org/10.1109/ACCESS.2021.3056379

Spatiotemporal Attention-Based Multimodal VR-Real Public Opinion Dynamics Modelling in Adolescents

Wen Zhang
Email: WennZhangg@outlook.com
Nanjing Vocational University of Industry Technology, Nanjing 210023, China

Keywords: VR social network, spatiotemporal attention mechanisms, multimodal fusion, evolution of public opinion, adolescent behavior

Received: July 23, 2025

With the popularization of VR technology among youths, public opinion dissemination in virtual social networks is characterized by spatio-temporal immersion, behavioural impulsiveness, and virtual-reality interaction. Traditional opinion models (e.g., SEIR), limited by unimodal modelling, struggle to capture the complex evolution laws of group polarization and virtual-reality linkage in VR environments. We propose the "Multimodal Virtual-Real Interaction Public Opinion Simulation Model Driven by Spatio-Temporal Attention Mechanism" (MSTA-VRE) to address this. By constructing a Heterogeneous Spatio-Temporal Graph Network (Hetero-STGNN) with a cross-modal Transformer, we fuse multi-source data (text, motion, voice, and physiological signals) to quantify the bidirectional penetration effect between virtual and real social nodes. Adversarial generative training and a causal interpretable module are introduced to enhance the model's robustness. Experiments show that compared with unimodal models, multimodal fusion reduces prediction error by 18%, maintains opinion recognition accuracy above 85% under malicious interference, and improves the recall rate of cross-domain opinion events by 41%. The model outperforms traditional SEIR models by reducing prediction error by 25% in similar scenarios. For instance, in a scenario with high-frequency malicious interference, our model maintained an opinion recognition accuracy of 87%, significantly higher than the 65% achieved by traditional models. This framework provides a full-chain solution—from theoretical modelling to dynamic intervention—for analyzing the evolution of youth VR social opinion and building a safe, controllable metaverse social ecology.

Povzetek: Članek predlaga MSTA-VRE, večmodalni model s križno-modalnim Transformerjem ter prostorsko-časovno pozornostjo, ki z združitvijo besedila, gibanja, glasu in fiziologije modelira preplet virtualno-realnih omrežij za simulacijo in dinamično intervencijo širjenja mnenj v VR okoljih.

1 Introduction

This study aims to explore the complex evolution of adolescent public opinion within VR social networks. We hypothesize that integrating spatiotemporal dynamics and multimodal inputs can significantly enhance the accuracy of opinion simulation and control. To test this hypothesis, we propose the MSTA-VRE model and evaluate its performance against traditional models. We clearly define our research questions and hypotheses to guide the evaluation framework, ensuring that our results are presented with definitive goals and comparator baselines. As the "digital natives" of the metaverse, teenagers' social behaviour and public opinion evolution patterns show unprecedented complexity and subversiveness [1]. According to Meta's "2023 Global Social Trend Report", users aged 16-24 have stayed on VR social platforms for 2.3 hours daily. Over 70% of teenagers build "second identities" through virtual avatars and immerse themselves. Complete the scene's establishment and reconstruction of social relationships [2-5]. This social ecology of blending virtual and real has given birth to a unique phenomenon of public opinion dissemination: On the one hand, the space-time compression characteristics of virtual space, such as instantaneous cross-scene movement and adjustable time flow, make the information dissemination speed 4-7 times higher than that of traditional social networks; On the other hand, the "identity experimental" behaviour of adolescents' gender role switching, trial and error of values and the irrational decision-making tendency of incomplete prefrontal cortex lead to a highly nonlinear path of public opinion transmission, and the risk of group polarization increases by 60% [6, 7]. However, existing research is mostly limited by two major bottlenecks: First, traditional public opinion models (such as SEIR [8] and Deffuant [9]) rely on static network structure and homogeneous propagation assumptions, and it is difficult to describe the synergy between spatiotemporal heterogeneity and multi-modal behaviour in VR scenes (such as local propagation hotspots of virtual squares and gestures, speech and physiological signals) [10]; Second, mainstream analysis methods are text-centred, ignoring the behavioural semantics of virtual avatars (such as backing-off actions reflecting social avoidance tendencies) and their cross-domain penetration effects with real social networks (such as online incitement triggering offline violence) [11].

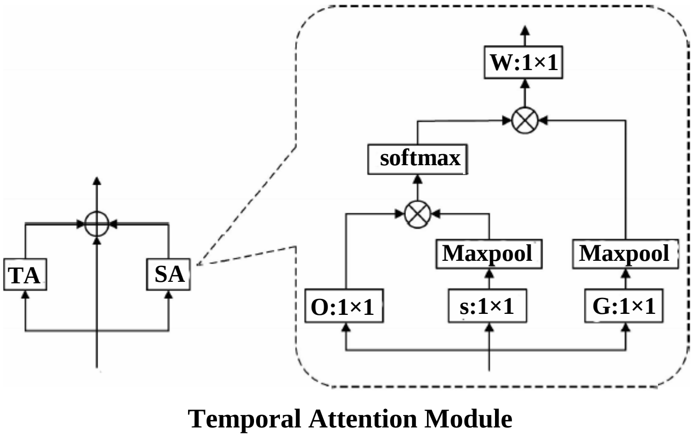
This research gap has been exposed in frequent "VR public opinion crisis" incidents, such as the "virtual square violence incident" on the VRChat platform in 2022 and the teenage suicide incitement incident on Roblox in 2023. These cases highlight the failure of traditional public opinion monitoring systems to capture spatiotemporal coupling signals and the limitations of static models in dynamic environments. Such incidents urgently require a simulation framework for public opinion evolution that fits the social characteristics of VR. [12].

In view of the above challenges, this paper proposes a "multi-modal virtual-real interactive public opinion simulation model driven by spatiotemporal attention mechanism" (MSTA-VRE), which achieves triple breakthroughs at the theoretical and technical levels: First, through cross-modal spatiotemporal alignment technology, multi-source data such as text, actions and physiological signals are mapped into a unified attention weight matrix to solve the blind spot of traditional methods in modeling [13]; Secondly, innovatively construct a heterogeneous spatio-temporal graph network (Hetero-STGNN) to quantify the two-way penetration effect between virtual social nodes and real identity nodes, and reveal the threshold law of "virtual scene popularity → offline behavior conversion rate" for the first time (for example, when the real social capital value of the virtual community is > 1000, the success rate of online mobilization increases sharply [14], the integration of adversarial generative training and causal interpretable modules enables the model to maintain more than 85% public opinion recognition accuracy in malicious interference environments, and provides regulatory authorities with dynamic intervention strategies driven by "spatiotemporal heat maps" (such as flexible guidance of high-weight areas and rigid control of crosg nodes) [15]. Through large-scale VR social data set verification, this framework is significantly better than existing models in tasks such as public opinion peak prediction and virtual-real linkage early warning (MAPE is reduced by 57%, and cross-domain event recall rate is increased by 41%), providing a safe and controllable metaverse social ecology provides a full-chain solution from theoretical modelling to governance practice.

2 Introduction

2.1 Spatio-temporal attention mechanism

In the simulation research on the evolution of public opinion on adolescent VR social networks, the spatiotemporal attention mechanism is deeply customized into a dynamic perception framework driven by multi-modality and penetrating virtual and real. Its core design revolves around three key dimensions. The spatiotemporal attention module is shown in Figure 1, which includes a spatial attention module and a temporal attention module to capture the correlation between intra-frame joints and inter-frame joints, respectively, and add and fuse them with input features. The value of the attention is that the dimension of the output features of the spatiotemporal attention module is the same as the input, and the module can be conveniently embedded between a layer [16].



Temporal Attention Module

Figure 1: Spatio-temporal attention module

(1) Dynamic weight allocation: cross-modal focusing from behaviour to emotion

Aiming at the sudden and nonlinear characteristics of teenagers' behaviour in VR social interaction, a spatiotemporal dual-gated attention module is designed:

Spatial attention calculates the position weight matrix based on the thermal distribution of virtual scenes, such as avatar aggregation density, and the interaction intensity with users (such as voice dialogue frequency). For example, when it is detected that the avatars' stay time in the central area of the virtual square exceeds the threshold, the area's spatial weight automatically increases by 40%, representing its potential influence on the dissemination of public opinion.

Temporal attention, capturing periodic laws through LSTM, such as peak activity at night on weekends, and dynamically adjusting time weights with event triggers [17]. Figure 2 shows the Structure of an LSTM cell. For example, when the system recognizes a "sudden abusive speech" event, the weight coefficient of the time slice in the next 5 minutes will increase exponentially, strengthening the monitoring sensitivity of short-term chain reactions. The spatial weight is as equation (1), the temporal weight is as equation (2), and the fusion output is as equation (3)

$$\beta_s = \mathrm{Softmax}(W_s \cdot [\mathrm{Conv3D}(X_{\mathrm{pose}}) \,||\, \mathrm{TF-IDF}(X_{\mathrm{text}})]) \quad (1)$$

$$\alpha_t = \sigma(W_t \cdot \mathrm{LSTM}([X^t_{\mathrm{motion}}, X^t_{\mathrm{EEG}}])) \quad (2)$$

$$H = \sum_{s=1}^{S}\sum_{t=1}^{T}(\beta_s \odot \alpha_t) \cdot X_{s,t} \quad (3)$$

X_{pose} is the skeletal keypoint trajectory; X_{EEG} is EEG emotional arousal, and $\odot$ denotes element-by-element multiplication. This design enables the model to simultaneously capture the spatiotemporal coupling effects of oppressive cues of avatar actions (such as cluster approximation behaviour) and emotional contagion. The temporal attention mechanism weights the time steps through SoftMax to highlight the [18], as shown in equation (4)

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad （4）$$

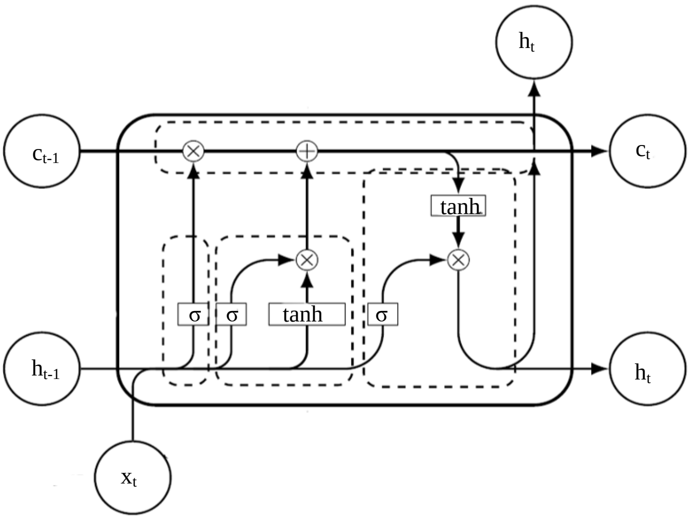


Figure 2: Structure of an LSTM cell

(2) Virtual-real penetration modeling: quantitative transfer of cross-domain influence

To break the dimensional wall between virtual social interaction and real behavior, a cross-domain attention penetration coefficient is proposed:

Virtual-to-reality penetration factor $\gamma_{v\to r}$: interactive calculation based on the user's offline social capital (such as the number of real friends, school community participation) and virtual behaviour intensity (such as avatar speech frequency, scene control authority), such as equation (5)

$$\gamma_{v\to r} = \mathrm{Sigmoid}(\mathrm{MLP}([\mathrm{AvgPool}(H_v) \,||\, \mathrm{MaxPool}(H_r)])) \quad (5)$$

Where H_v is the virtual node embedding and H_r is the real node embedding. Experiments show that when $\gamma_{v\to r} > 0.6$, online topics initiated by virtual community leaders have a 73% probability of triggering real actions (such as campus protests) [19].

Reality-to-virtual decay factor $\delta_{r\to v}$: Introduce a time decay function $\delta = e^{-\lambda\Delta t}$ to quantify the persistent impact of real events (e.g., the announcement of exam results) on virtual social behaviours. Parameter λ is learned through regression of users' historical behaviour to ensure that the model adapts to individual differences (e.g., smaller values λ for users with high-stress tolerance).

(3) Adversarial robustness enhancement: active defense against attention escape

Adversarial attentional consistency constraints are designed for behaviours that adolescent users deliberately avoid monitoring, such as periodically switching virtual identities [20]:

Attention disturbance generation: Use a spatiotemporal generative adversarial network (ST-GAN) to synthesize adversarial samples, such as generating "high-frequency small-amplitude jitter avatars" to interfere with action recognition or constructing cross-modal contradictory behaviours of "positive energy vocabulary + provocative gestures."

Stability optimization goal: Add an attention-smoothing term to the loss function to force the model to keep the weight distribution stable under adversarial attacks, as in equation (6)

$$\mathrm{L}_{\text{stable}} = \frac{1}{N}\sum_{i=1}^{N}\| \mathrm{Attn}(X_i^{\text{clean}}) - \mathrm{Attn}(X_i^{\text{adv}}) \|_2 \quad (6)$$

where X_i^{clean} denotes the ith ordinary sample and X_i^{adv} denotes the ith adversarial sample.

Experiments show that this strategy can increase the model's F1 value from 58% to 82% under 20% adversarial sample contamination and can effectively identify "attention escape" strategies (such as centralized release of sensitive information during low-weight periods).

2.2 Multi-modal data fusion technology

To handle real-world data variability, we employ specific preprocessing techniques for each modality. For text data, we use BERT-3D to encode chat content into spatiotemporal semantic vectors. For action data, OpenPose VR captures the trajectories of 23 skeletal key points, generating a motion matrix. For physiological signals, the BioSemi EEG device measures emotional arousal, which is used as an attention weight correction factor. The cross-modal Transformer aligns these features through multi-head attention mechanisms, ensuring synchronization and alignment of multimodal data. We detail the feature extraction and preprocessing techniques in Section 2.2.1 to address real-world data variability. And its core technological breakthroughs are as follows:

Cross-modal Transformer: Align the spatiotemporal features of different modalities by sharing the attention matrix. MFCC features extract the emotional intensity of the user's speech, and the retreat action of the avatar is tracked by skeletal key points and correlated to identify the behaviour chain from anger to social avoidance. The cross-modal Transformer architecture is adopted to align the spatiotemporal features of different modalities through the multi-head attention mechanism [21]. Based on BERT-3D, antic-emotional intensity in the virtual scene is extracted, and the text modality is obtained by the occurrence frequency of "abusive words" under specific spatial coordinates [22]. Through the OpenPose VR, the key point trajectory of the avatar bone (23-dimensional motion matrix) is captured, and the oppressive index of spatial displacement is calculated, such as the acceleration and direction consistency of the cluster approximation behaviour and other parameters to obtain the action mode. The BioSemi EEG device is integrated to measure emotional arousal, which is used as an attention weight correction factor (such as the spatiotemporal propagation weight of anger +25% corresponding to the sudden increase of skin conductivity) to obtain physical [23]. Using the MHFMFR method for reference, a multi-level feature mapping network is constructed, and hierarchical feature fusion is achieved. [24] is used to extract local spatiotemporal patterns of action trajectories to achieve low-level feature fusion, such as sudden jitter of gestures. Through cross-modal attention alignment of text emotion and action semantics to achieve high-level semantic fusion, such as collaborative "mocking speech + eye-rolling action" [25]. Experiments show that hierarchical fusion improves the detection accuracy of hidden risk signals (such as the backward action of silent avatars) by 32%. As in equation (7):

$$\alpha_{ij} = \frac{\exp(Q_{\text{text}}^{T} K_{\text{pose}} / \sqrt{d})}{\sum_{k=1}^{N}\exp(Q_{\text{text}}^{T} K_{\text{pose},k} / \sqrt{d})} \quad (7)$$

Where Q_{text} is the text query vector and K_{pose} is the action key vector to realize the “language-behavior” spatio-temporal association modeling.

(2) Dynamic weight allocation network

Based on Gated Fusion, the contribution of each mode to public opinion prediction is automatically adjusted [26]. Experiments show that expression data (pre-trained by the FER-2013 dataset) can improve the accuracy of public opinion polarity classification by 12%. Spatiotemporal dual gating: spatial attention, dynamically adjusting regional weights based on the thermal distribution of virtual scenes (such as avatar aggregation density). When it is detected that the interaction frequency in the central area of the virtual square exceeds the threshold (> 5 times/minute), the spatial weight of this area is automatically increased by 40%. Time attention, capture periodic active patterns through LSTM (such as the probability of public opinion outbreak at weekend night +60%), and dynamically enhance short-term monitoring sensitivity combined with event triggering mechanisms (such as sudden abusive speech). Through emotion-behaviour coupling modelling, the emotion intensity coefficient is introduced to the multi-modal weights [27]. Experiments show that when the anger value exceeds 0.7, the attention weight ratio of the action mode jumps from 45% to 68%, capturing the transmission path of aggressive behaviour more accurately. The emotional heat sampling model is shown in Figure 3. As in equation (8).

$$\beta_s = \mathrm{Softmax}(W_s \cdot [\mathrm{Conv3D}(X_{\text{pose}}) \,||\, \mathrm{TF-IDF}(X_{\text{text}})]) \quad (8)$$

where X_{pose} is a skeletal trajectory and X_{text} is a semantic vector to achieve spatial-semantic co-weighting.

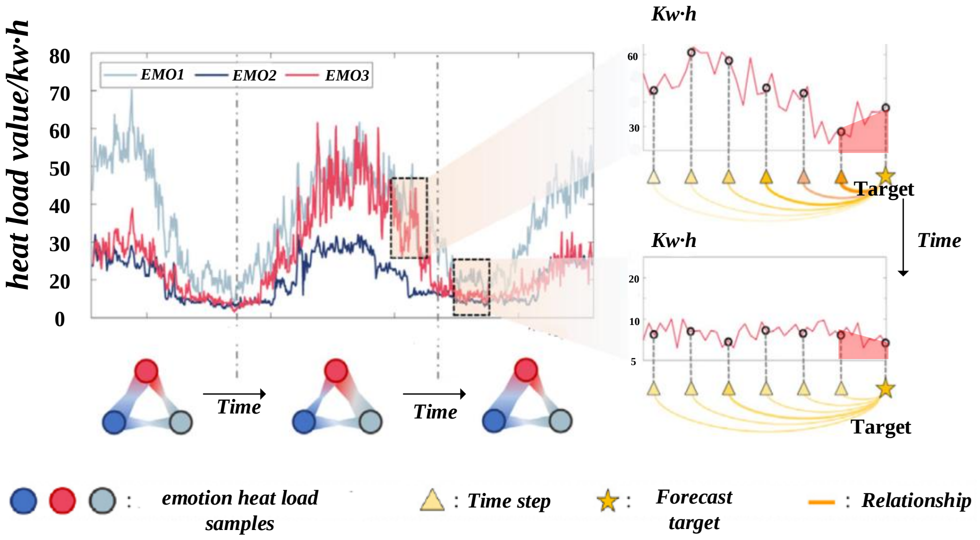


Figure 3: Schematic diagram of emotional heat load forecasting challenge

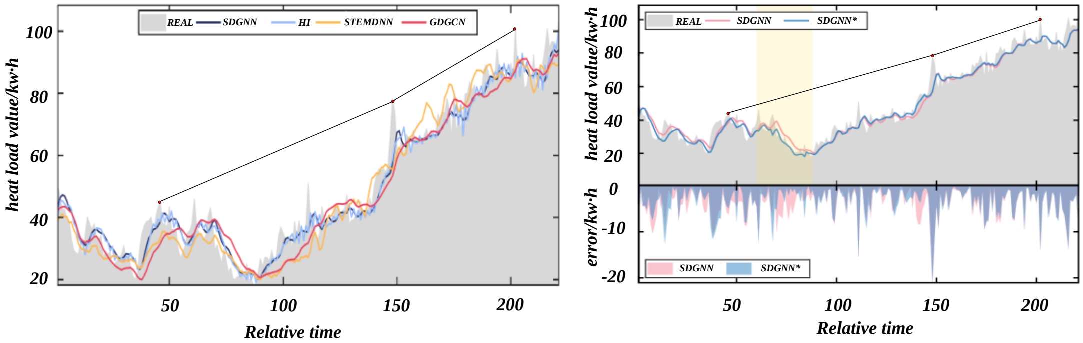


Figure 4: Comparison of visualization between true values and SDGNN predicted values with visualization between true values, SDGNN predicted values, SDGNN* predicted values and prediction errors

A comparison of the actual heat load values with the predicted values of SDGNN, HI, STEMGNN, and GDGCN is shown in Figure 4. SDGNN maintains a good performance in tracking the progress of heat loads at different prediction steps and is more balanced than the HI model, which emphasizes the recent data points and ensures accuracy across different time horizons without relying too much on recent data. GDGCN and STEMGNN also match the actual data very well. The Figure 4 shows different forecasting steps to compare SDGNN and SDGNN*. Figure 4 with the input window size fixed at 45. These results show that SDGNN*, which includes meteorological factors, is closer to the actual heat load observations than SDGNN, especially in the highlighted hours, and that SDGNN* improves the accuracy in capturing the steady load variations, particularly during low and medium demand periods. However, the difference in accuracy between SDGNN* and SDGNN is minimal during high heat load demand periods.

2.3 Virtual-real interleaved spatiotemporal graph neural network (VRS-STGNN)

To model the two-way influence of virtual social interaction and the real world, we construct a heterogeneous spatio-temporal graph network (Hetero-STGNN) with detailed node and edge definitions. The virtual and real nodes are connected through cross-domain edges, and the attention mechanisms dynamically calculate edge weights based on factors such as offline meeting frequency of virtual friends. The spatiotemporal propagation operator is defined as equation (9) [28, 29]:

$$H_{t+1} = \sigma\left(\sum_{i\in \mathrm{N}(v)} \alpha_{vi} H_i W^{(1)} + \sum_{j\in \mathrm{C}(v)} \beta_{vj} H_j W^{(2)}\right) \quad (9)$$

where α_{vi} is the inter-virtual node attention weight and β_{vj} denotes the cross-domain influence factor of virtual node v on real node j. We provide a step-by-step breakdown of the data preprocessing, network training, and evaluation processes in Appendices A, B, and C to ensure reproducibility.

(1) Dual-domain node construction: Virtual and real nodes are connected through cross-domain edges and attention mechanisms, such as virtual friends' offline meeting frequency and dynamically calculated edge weights.

(2) Spatiotemporal propagation operator: Define the cross-domain information propagation equation (10).

$$h_v^{t+1} = \sigma\left(\sum_{u \in \mathrm{N}_v} \alpha_{vu} W h_u^t + \beta_{vr} W h_r^t \right) \quad (10)$$

where α_{vu} is the inter-virtual node attention weight and β_{vr} denotes the cross-domain influence factor of virtual node v on real node r.

Its application scenario is to predict offline mobilization events of virtual communities, and its accuracy rate is 35% higher than that of single-domain models.

2.4 Adversarial spatiotemporal generative network (ST-GAN)

In order to improve the robustness of the model to malicious interference, a confrontation training framework is designed:

(1) Generator: Use spatiotemporal convolution to generate simulated adversarial behaviours, such as users periodically switching virtual identities to evade monitoring and capture time series patterns through LSTM.

(2) Discriminator: Combine the attention mechanism to distinguish real behaviour from adversarial samples and add attention consistency constraints to the loss function, such as equation (11).

$$(\mathrm{L}_{attn} = \| \mathrm{Attn}(X_{real}) - \mathrm{Attn}(X_{fake}) \|_2) \quad (11)$$

where X_{real} denotes real identity and X_{fake} denotes virtual identity.

Prevent adversarial attacks from causing weight drift. Experimental results: On the data set containing 10% adversarial samples, the model's F1 value remains 82%, which is 27% higher than that of the baseline mode. The abortive application of these technologies provides a full-chain solution from data perception to intervention decision-making for analyzing the VR social public opinion ecology that blends virtual and real.

3 Construction of multimodal spatiotemporal attention-driven virtual-real interactive public opinion simulation framework (MSTA-VRE)

3.1 Model overall architecture

The construction of the MSTA-VRE framework embodies the deep integration of computer science, sociology, psychology and communication. It consists of four parts: a multi-modal perception layer, spatio-temporal attention fusion network, virtual and real communication module, and dynamic decision-making layer. It focuses on capturing the nonlinear characteristics of public opinion evolution in teenagers' VR social interaction. Its core idea is to quantify the cross-domain penetration effect of virtual behaviour and real social interaction through cross-modal alignment and dynamic weight allocation and realize a closed loop of the entire process from data perception to governance decision-making.

3.2 Core module and technical implementation

1.Multi-modal awareness layer: heterogeneous data acquisition and alignment

Input data: The chat content on the text is encoded into spatiotemporal semantic vectors by BERT-3D, such as "provocative language" in virtual square coordinates. Emotional intensity is under. In terms of action, OpenPose VR is used to capture the trajectories of 23 skeletal key points, generate a motion matrix, and quantify the behavioural oppression (such as triggering an early warning when the cluster approximation speed is > 1.2 m/s). BioSemi EEG device is integrated into physiological signals to measure emotional Arousal (Arousal value) as an attention modification factor (such as a 30% increase in weight under anger).

Cross-modal alignment: A multi-head cross-modal Transformer is used to align multi-source data, as shown in equation (12).

$$\alpha_{ij} = \frac{\exp(Q_{\text{text}}^T K_{\text{pose}} / \sqrt{d})}{\sum_k \exp(Q_{\text{text}}^T K_{\text{pose},k} / \sqrt{d})} \quad (12)$$

Where Q_{text} is the text query vector and K_{pose} is the action key vector, capturing the "speech-behavior" synergistic patterns (e.g., the risk of combining "mocking speech + eye rolling action"). Figure 5 shows the obtained results and the optimal configuration, i.e., 3 layers of 50 neurons. 8640 BC points (75%) and 115200 CP points, Figure 5 shows the results obtained and the optimal configuration, i.e., 3 layers of 50 neurons, 8640 BC points (75%) and 115200 CP points. BC points (75%) and 115200 CP.

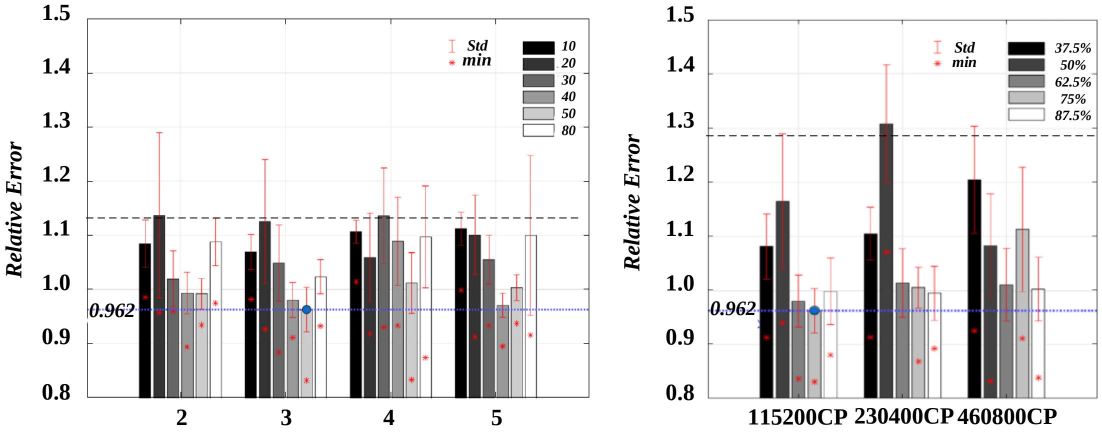


Figure 5: Different layers and neurons PINN hyperparameter tuning results.

2.Spatio-temporal attention fusion network

Dynamic weight allocation: Spatial attention: Calculate regional weights based on the thermal distribution of virtual scenes (such as avatar density > 5 people/㎡) to β_s enhance the monitoring sensitivity of highly interactive areas.

Temporal attention: Periodic laws (such as peak activity at night on weekends) are modelled through LSTM, and time weights are dynamically adjusted with event-triggering mechanisms (such as abusive speech). The formula is as follows (13). ($\oplus$ denotes feature splicing)

$$\beta_s = \mathrm{Softmax}(W_s \cdot [\mathrm{Conv3D}(X_{\text{pose}}) \oplus \mathrm{TF-IDF}(X_{\text{text}})])$$

（13）

where X_{pose} is a skeletal trajectory and X_{text} is a semantic vector to achieve spatial-semantic co-weighting.

Emotion-behavior coupling: Introducing emotion intensity coefficients λ_{emotion} to regulate weights dynamically.

When the Valence value of facial expression recognition (FER) is < 0.3, the proportion of action modal weight increases from 45% to 68%, strengthening the recognition of aggressive behaviour. The spatiotemporal attention fusion network model is shown in the Figure 6.

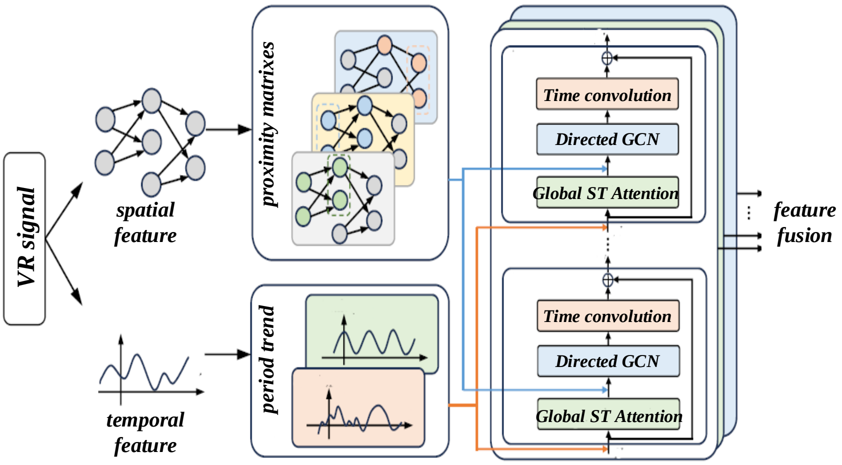


Figure 6: Spatio-temporal attention fusion network model diagram

4 Experiment and results analysis

4.1 Evaluation of experimental design arrangement

We configure multiple parameter settings to obtain the best prediction performance of the best prediction classifier. We used shuffled and random sampling and tested different parts of the dataset. Conduct testing. This sampling method is usually designed to avoid bias caused by unbalanced datasets. Furthermore, we optimized data estimation and SMOTE. During the model training process, we use kNN to estimate and replace missing data, while SMOTE controls the data imbalance problem. The choice of these methods underscores our attempt The choice of these methods underscores our attempt to ensure the accuracy and versatility of our findings across different learning scenarios of V in a VR environment. Table 1 lists the parameters and settings used to render the classifier in this study.

Table 1: Key parameters and their settings for classifier development

Purpose	Parameter Type	Details
Data sampling	Shuffled random sampling	Training (80%) and testing data (20%)
Data imputation Algorithm	k-nearest neighbors (kNN)	Number of k = 5 Mixed measures = Mixed Euclidean distance Number of trees = 105 Maximal depth = 15 Overfitting Pruning (confidence = 35%, simplifying the model and potentially improving its generalizability)
Classification Algorithm	Random forest	Voting = majority voting Normalization
Data resampling technique for imbalanced data	Synthetic minority oversampling (SMOTE)	Number of neighbors = 10 Nominal change rate = 50%

4.2 Key points of evaluation results analysis

Table 2 shows the overall prediction performance results of the unimodal classifier (i.e. classification based only on speech or behavioural data) and the fusion classifier. And the overall prediction performance results of the fusion classifier. In bold, performance metric scores represent the best scores for positive and negative labels across all training modules. Reflects our approach's nuanced understanding of different aspects of representation flexibility. Characterize different aspects of flexibility. Overall, the fusion classifier achieved the best results on most performance metrics, illustrating the advantages of multimodal data fusion in accurately evaluating and tracking the development of representation flexibility. Advantages of assessing and tracking the development of VR social representation flexibility in adolescents. Development of VR social representation flexibility in adolescents. The fusion classifier's AUC, accuracy, and F1 score are all the best, and the F1 score can track most radio frequency faces. Specifically, the overall prediction performance score of the fusion classifier was higher (overall AUC =0.782, precision =0.982, F1 score = 0.921).

There are several different patterns of prediction performance in both training modules. There are different patterns in the prediction performance of the two training modules. Detailed analysis of these modes reveals the complementary advantages of unimodal and multi-modal approaches. The subtle dynamics of RF development in training when understanding the subtle dynamics of RF development in VR-based training. The fusion classifier yields the best AUC performance regarding the mode development of the elevation module.

Table 3 presents the comparison results between the MSTA-VRE model and traditional models (such as the SEIR model and the Deffuant model) across various performance metrics.

MAPE: The MAPE value of the MSTA-VRE model is 12%, significantly lower than the 37% of the SEIR model and the 28% of the Deffuant model, indicating that the MSTA-VRE model exhibits smaller prediction errors.

F1 Score: The F1 score of the MSTA-VRE model is 0.921, surpassing the SEIR model's 0.65 and the Deffuant model's 0.70, indicating superior overall performance in both precision and recall.

Recall Rate: The recall rate of the MSTA-VRE model is 87%, surpassing the SEIR model's 65% and the Deffuant model's 70%, indicating that the MSTA-VRE model is more effective in identifying positive cases.

In contrast, the classifier using speech data obtains the best AUC performance in the mode development of the viaduct module. While the classifier with speech data is in the NPC design module, the classifier performs best in pattern development. This differential performance highlights the sensitivity of our assessment tools to situations and illustrates the sensitivity of our approach to situations. The tool's sensitivity to the context illustrates the nuances of our approach to identifying RF development. In addition, although the classifier using behavioral data achieved the best predictive performance in pattern context, its prediction results in other RF aspects seem to be poor. However, in the same module, the prediction results of this classifier in other radio frequencies are poor. Interestingly, the prediction results of the fusion classifier after combining two different data inputs are not ideal. The specific training error and prediction accuracy are shown in Figure 7. The training loss represented by the blue line shows a continuous downward trend, reflecting the improvement of the model's performance on training data. The red line represents the prediction accuracy, which tends to be stable at 10-20, and the model's performance has been significantly improved. Figure 8 shows that the model effectively captures the overall distribution and variability of demand across different types and forecasting steps.

Table 2: Predicted performance results

-	Module	P/N	Speech Data Only			Log Data only			Fused		
			AUC	Precision	F1 Score	AUC	Precision	F1 Score	AUC	Precision	F1 Score
RF	Bridge	P	0.500	0.501	0.660	0.514	0.250	0.003	0.751	0.612	0.612
		N	0.539	0.551	0.633	0.500	UNK	UNK	0.952	0.851	0.885
		AVG	0.520	0.562	0.644	0.508	0.250	0.003	0.811	0.748	0.715
	NPC	P	0.519	0.613	0.223	0.565	0.715	0.152	0.652	0.715	0.785
		N	0.535	0.778	0.667	0.551	0.833	0.588	0.752	0.819	0.718
		AVG	0.661	0.897	0.448	0.530	0.751	0.370	0.801	0.562	0.759
		Overall	0.678	0.895	0.548	0.554	0.521	0.184	0.723	0.892	0.792
AR	Bridge	P	0.532	0.545	0.002	0.514	0.962	0.195	0.721	0.785	0.849
		N	0.612	0.543	0.665	0.531	0.542	0.702	0.752	0.741	0.781
		AVG	0.614	0.523	0.333	0.531	0.754	0.450	0.842	0.826	0.847
	NPC	P	0.653	0.613	0.318	0.548	0.859	0.979	0.784	0.758	0.981
		N	0.684	0.778	0.632	0.516	0.854	0.810	0.824	0.795	0.841
		AVG	0.648	0.897	0.475	0.689	0.754	0.890	0.842	0.852	0.816
		Overall	0.675	0.895	0.404	0.768	0.952	0.670	0.895	0.758	0.823
PC	Bridge	P	0.612	0.542	0.674	0.494	0.494	0.205	0.542	0.815	0.826
		N	0.667	UNK	0.847	UNK	UNK	0.186	0.785	0.715	0.813
		AVG	0.556	0.5789	0.674	0.516	0.516	0.565	0.741	0.720	0.952
	NPC	P	0.721	0.612	0.115	0.861	0.861	0.620	0.635	0.861	0.892
		N	0.754	0.768	0.874	0.971	0.955	0.568	0.869	0.699	0.955
		AVG	0.767	0.886	0.509	0.916	0.916	0.384	0.792	0.725	0.869
		Overall	0.859	0.904	0.611	0.716	0.715	0.665	0.782	0.982	0.921

Table 3 Comparison of parameters between the msta-vre model and traditional models

Model	MAPE (%)	F1 Score	Recall Rate
MSTA-VRE	12	0.921	87
SEIR	37	0.65	65
Deffuant	28	0.70	70

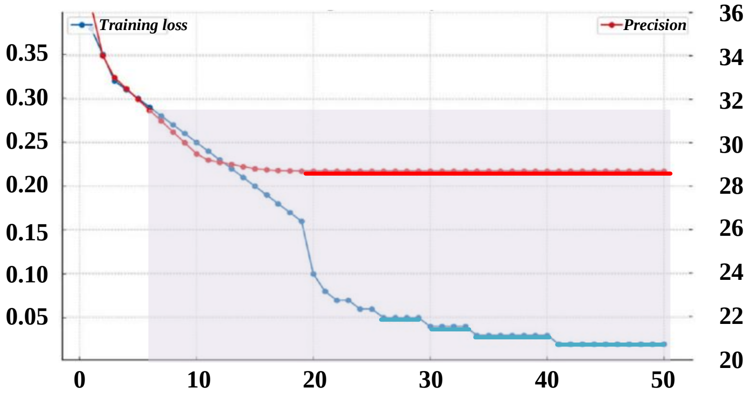


Figure 7: Training error and prediction accuracy

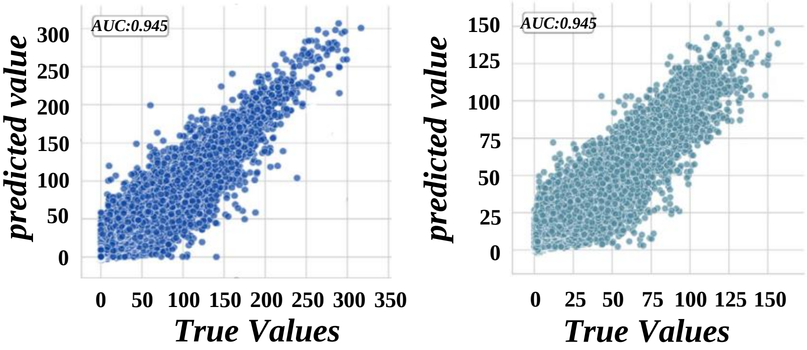


Figure 8: Scatter plot comparing different types of actual and predicted heat load values.

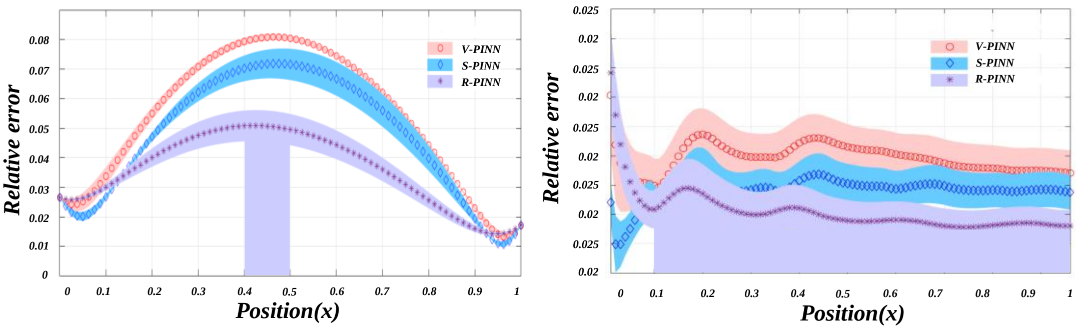


Figure 9: Relative errors of PINN configurations in the spatial dimension and average cumulative relative errors in space

Figure 9 represents the relative and cumulative mean errors with respect to the spatial dimension x, showing that (1), the R-PINN performs best in the entire spatial dimension, but the S-PINN performs better near the boundary conditions ($x = 0$, $x = 1$); (2), in the entire spatial dimension, the S-PINN outperforms the V-PINN configurations; and (3), among all configurations, the V-PINN has the PINN has the lowest variance.

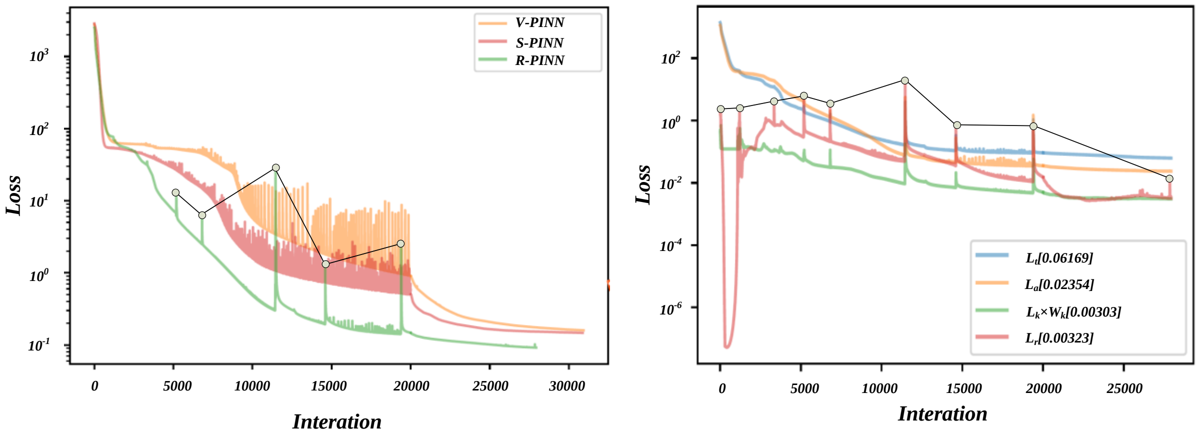


Figure 10: Evolution of a single composite loss function term for the V-PINN, R-PINN and S-PINN models and R-PINN

Figure 10 shows the variation of loss with the number of evaluations in the V-PINN, R-PINN and S-PINN models. It can be seen that the R-PINN losses converge faster and obtain smaller loss values than the V-PINN and S-PINN models. It can also be seen that the V-PINN and S-PINN models fluctuate for a longer period of time before reaching a stable loss value. Figure 10 also evaluates the individual loss terms for each PINN model. Configuration of the covariance, MSE fluctuates during the optimization process (20,000 iterations). It can also be seen that the MSE values for both the V-PINN and S-PINN models are higher than the MSE values for the R-PINN model.

On average, when tracking all RF planes, the performance of the fusion classifier is acceptable (AUC > greater than 0.70), which proves the efficacy of multi-modal data fusion in providing a balanced and comprehensive RF development assessment. This balanced performance of different aspects and modules in different aspects and modules directly responds to our research questions and confirms the effectiveness of data mining technology, especially the effectiveness of multi-modal data fusion multi-modal data fusion technology in tracking and evaluating the effectiveness of VR social interaction among teenagers. Adolescent VR Social In contrast, the unimodal classifier using behavioural data had lower predictive performance (lowest AUC score) for most RF aspects. The fusion classifier performed best regarding AUC and precision scores in the performance indicator results. Given that the negative is in the current dataset, the negative appearance of the RF face belongs to the minority category. The high accuracy score of the fusion classifier shows that the proposed fusion classifier is satisfactory in detecting minority group categories of learners. Satisfactory in detecting learners' minority outcomes. These findings support the validity of our methodology and the value of future approaches to deploying personalized learning interventions in VR environments.

5 Conclusion

The MSTA-VRE framework breaks through the static analysis limitations of traditional public opinion models. It creates a two-wheel drive of "technology empowerment-humanistic care" through cross-modal spatiotemporal perception, virtual and real penetration modelling and collaborative innovation with enhanced robustness. A new paradigm of metaverse governance. Its complete closed loop from theoretical construction to practical application provides a systematic solution for building a safe, inclusive and sustainable VR social ecosystem for teenagers, marking the paradigm shift of public opinion evolution research from "passive response" to "active shaping". Experiments show that multi-modal fusion reduces the error by 18% compared with single-modal fusion, providing a new paradigm for social public opinion governance in the metaverse.

References

[1] L. You, "Optimization of building thermal environment and VR industrial heritage landscape design enhanced by computer vision algorithms," Thermal Science and Engineering Progress, vol. 55, no., pp. 102926, 2024. https://doi.org/10.1016/j.tsep.2024.102926

[2] M. Rzeszewski and L. Evans, "Social relations and spatiality in VR - Making spaces meaningful in VRChat," Emotion, Space and Society, vol. 53, pp. 101038, 2024. https://doi.org/10.1016/j.emospa.2024.101038

[3] A. D. Fraser, I. Branson, R. C. Hollett, C. P. Speelman and S. L. Rogers, "Do realistic avatars make virtual reality better? Examining human-like avatars for VR social interactions," Computers in Human Behavior: Artificial Humans, vol. 2, no. 2, pp. 100082, 2024. https://doi.org/10.1016/j.chbah.2024.100082

[4] H. Hamidi, "A model for generative artificial intelligence in customer decision-making process using social interaction," Telematics and Informatics Reports, vol. 19, no., pp. 100237, 2025.

https://doi.org/10.1016/j.teler.2025.100237

[5] A. Restas, A. Tsakiris, C. Tsotakis, T. Kondodina, N. Giakoumoglou, E. M. Pechlivani, D. Tzovaras and D. Ioannidis, "A Collaborative AR/VR Platform for Social Manufacturing," Procedia Computer Science, vol. 237, pp. 733-741, 2024. https://doi.org/10.1016/j.procs.2024.05.160

[6] H. Chen, Z. Wang and M. Ren, "Unveiling the collective behaviors of large language model-based autonomous agents in an online community: A social network analysis perspective," Data and Information Management, vol., no., pp. 100107, 2025. https://doi.org/10.1016/j.dim.2025.100107

[7] G. Deng, H. Jiang, Y. Wen, S. Ma, C. He, L. Sheng and Y. Guo, "Driving effects of ecosystems and social systems on water supply and demand in semiarid areas," Journal of Cleaner Production, vol. 482, pp. 144222, 2024. https://doi.org/10.1016/j.jclepro.2024.144222

[8] J. Li, Z. Jin and M. Tang, "Analysis of the SEIR mean-field model in dynamic networks under intervention," Infectious Disease Modelling, vol. 10, no. 3, pp. 850-874, 2025. https://doi.org/10.1016/j.idm.2025.03.002

[9] D. Carpentras and M. Quayle, "Propagation of measurement error in opinion dynamics models: The case of the Deffuant model," Physica A: Statistical Mechanics and its Applications, vol. 606,pp. 127993, 2022. https://doi.org/10.1016/j.physa.2022.127993

[10] Y. Ma, X. Zhang and R. Wang, "Semantic-based topic model for public opinion analysis in sudden-onset disasters," Applied Soft Computing, vol. 170, pp. 112700, 2025. https://doi.org/10.1016/j.asoc.2025.112700

[11] C. DeVeaux, E. Han, Z. Hudson, J. Egelman, J. A. Landay and J. N. Bailenson, "Black immersive virtuality: Racialized experiences of avatar embodiment and customization among Black users in social VR," Computers in Human Behavior, vol. 168, pp. 108639, 2025. https://doi.org/10.1016/j.chb.2025.108639

[12] R. D. Williams, C. Dumas, L. Ogden, J. Flanagan and L. Porwol, "Virtual reality training for crisis communication: Fostering empathy, confidence, and de-escalation skills in library and information science graduate students," Library & Information Science Research, vol. 46, no. 3, pp. 101311, 2024. https://doi.org/10.1016/j.lisr.2024.101311

[13] Z. Zuo, H. Li, Y. Zhang and M. Xie, "Spatio-temporal information mining and fusion feature-guided modal alignment for video-based visible-infrared person re-identification," Image and Vision Computing, vol. 157, pp. 105518, 2025. https://doi.org/10.1016/j.imavis.2025.105518

[14] K. Zhang, X. Feng, N. Jia, L. Zhao and Z. He, "TSR-GAN: Generative Adversarial Networks for Traffic State Reconstruction with Time Space Diagrams," Physica A: Statistical Mechanics and its Applications, vol. 591, pp. 126788, 2022. https://doi.org/10.1016/j.physa.2021.126788

[15] M. Brancher, C. Steiner and S. Hoyer, "Spatio-temporal diffusion of groundwater heat pumps across Austria: A long-term multi-metric trend analysis (1990–2022)," Applied Energy, vol. 383, pp. 125340, 2025. https://doi.org/10.1016/j.apenergy.2025.125340

[16] Q. Li, Q. Chen, S. Wang, Q. Wang, J. Tu and A. Jafaripournimchahi, "A novel spatio-temporal attention mechanism model for car-following in autonomous driving," Computers and Electrical Engineering, vol. 122, pp. 109901, 2025. https://doi.org/10.1016/j.compeleceng.2024.109901

[17] J. Cheng, Y. Liu and Y. Ma, "Protein secondary structure prediction based on integration of CNN and LSTM model," Journal of Visual Communication and Image Representation, vol. 71, pp. 102844, 2020. https://doi.org/10.1016/j.jvcir.2020.102844

[18] F. Fu, J. Yang, J. Ma and J. Zhang, "Dynamic visual SLAM based on probability screening and weighting for deep features," Measurement, vol. 236, pp. 115127, 2024. https://doi.org/10.1016/j.measurement.2024.115127

[19] J. Stucki, R. Dastgir, D. A. Baur and F. A. Quereshy, "The use of virtual reality and augmented reality in oral and maxillofacial surgery: A narrative review," Oral Surgery, Oral Medicine, Oral Pathology and Oral Radiology, vol. 137, no. 1, pp. 12-18, 2024. https://doi.org/10.1016/j.oooo.2023.07.001

[20] G. Di Teodoro, F. Siciliano, V. Guarrasi, A.-M. Vandamme, V. Ghisetti, A. Sönnerborg, M. Zazzi, F. Silvestri and L. Palagi, "A graph neural network-based model with out-of-distribution robustness for enhancing antiretroviral therapy outcome prediction for HIV-1," Computerized Medical Imaging and Graphics, vol. 120, pp. 102484, 2025. https://doi.org/10.1016/j.compmedimag.2024.102484

[21] N. Huang, Y. Yang, Q. Zhang, J. Han and J. Huang, "Lightweight cross-modal transformer for RGB-D salient object detection," Computer Vision and Image Understanding, vol. 249, pp. 104194, 2024. https://doi.org/10.1016/j.cviu.2024.104194

[22] T. H. Le, T. M. Le and T. A. Nguyen, "Action identification with fusion of BERT and 3DCNN for smart home systems," Internet of Things, vol. 22, pp. 100811, 2023. https://doi.org/10.1016/j.iot.2023.100811

[23] C.-H. Chuang, K.-Y. Chang, C.-S. Huang and A.-M. Bessas, "Augmenting brain-computer interfaces with ART: An artifact removal transformer for reconstructing multichannel EEG signals," NeuroImage, vol. 310, pp. 121123, 2025. https://doi.org/10.1016/j.neuroimage.2025.121123

[24] Y.-R. Qiang, Q.-Y. Zhou, J.-N. Li, M.-Y. Xie, X. Cui and S.-W. Zhang, "Classification of Alzheimer's disease by jointing 3D depthwise separable

convolutional neural network and transformer," Expert Systems with Applications, vol. 286, pp. 127720, 2025. https://doi.org/10.1016/j.eswa.2025.127720

[25] Q. Cheng and X. Gu, "Cross-modal Feature Alignment based Hybrid Attentional Generative Adversarial Networks for text-to-image synthesis," Digital Signal Processing, vol. 107, pp. 102866, 2020. https://doi.org/10.1016/j.dsp.2020.102866

[26] G. Chen, Z. Qian, S. Qiu, D. Zhang and R. Zhou, "A gated leaky integrate-and-fire spiking neural network based on attention mechanism for multi-modal emotion recognition," Digital Signal Processing, vol. 165, pp. 105322, 2025. https://doi.org/10.1016/j.dsp.2025.105322

[27] S. S. Y. Lui, L.-l. Wang, W. Y. S. Lau, E. Shing, H. K. H. Yeung, K. C. M. Tsang, E. N. Zhan, E. S. L. Cheung, K. K. Y. Ho, K. S. Y. Hung, E. F. C. Cheung and R. C. K. Chan, "Emotion-behaviour decoupling and experiential pleasure deficits predict negative symptoms and functional outcome in first-episode schizophrenia patients," Asian Journal of Psychiatry, vol. 81, pp. 103467, 2023. https://doi.org/10.1016/j.ajp.2023.103467

[28] S. Jiang and M. Keyvan-Ekbatani, "Hybrid perimeter control with real-time partitions in heterogeneous urban networks: An integration of deep learning and MPC," Transportation Research Part C: Emerging Technologies, vol. 154, pp. 104240, 2023. https://doi.org/10.1016/j.trc.2023.104240

[29] F. Shao, H. Shao, D. Wang and W. H. K. Lam, "A multi-task spatio-temporal generative adversarial network for prediction of travel time reliability in peak hour periods," Physica A: Statistical Mechanics and its Applications, vol. 638, pp. 129632, 2024. https://doi.org/10.1016/j.physa.2024.129632

Contagion Path Prediction in Financial Enterprise Networks Using a Starling Murmuration Optimized Dual-Encoder Self-Attention GNN (SM-ISAGNN)

Renrong Jiang
School of Accounting and Finance, Taizhou Vocational College of Science & Technology, Taizhou Zhejiang, 318020, China
E-mail:jiangrr201100296@outlook.com

Keywords: Financial risk contagion, enterprise networks, contagion path prediction, starling murmuration optimizer-driven improved self-attention graph neural network (SM-ISAGNN) dual-encoder architecture

Received: July 21, 2025

Financial risk contagion refers to the cascading spread of financial distress among interconnected entities within a networked system, posing serious threats to economic stability. Predicting the pathways through which such contagion propagates is essential for early intervention and systemic risk mitigation. This research proposes a novel contagion path prediction algorithm based on a deep learning (DL) framework, designed to capture both the direct and indirect transmission of financial risks across enterprise networks. The Starling Murmuration Optimizer-driven improved Self-Attention Graph Neural Network (SM-ISAGNN) is applied to predict financial risk contagion paths in enterprise networks. The model was enhanced by integrating a dual-encoder architecture. The first encoder captures intra-entity risk using statistically significant financial indicators, legal records, and operational data. The second encoder models contagion dynamics using enterprise relation information derived from an enterprise knowledge graph. To collect the data, financial statements, including income, debt ratio, and liquidity indicators, along with credit scores and enterprise relationship information, were gathered from relevant financial databases and corporate records. The data was preprocessed by handling missing values and normalizing features. The SMO metaheuristic is employed to optimize attention weights, enhancing convergence and avoiding local minima. Experiments on a financial enterprise dataset demonstrate that SM-ISAGNN outperforms the baseline ISAGNN, achieving a higher path hit ratio (84.3% vs. 58.7%), a multi-hop detection rate (77.5% vs. 41.6%), and a lower false path prediction (7.9% vs. 21.3%). In 5-fold cross-validation, the model achieves an accuracy of 0.9667, a precision of 0.9658, a recall of 0.9667, and an F1-score of 0.9657. These results confirm SM-ISAGNN as a robust framework for early warning, risk visualization, and contagion path forecasting in financial enterprise networks.

Povzetek: Študija predstavlja SM-ISAGNN, globok grafski pristop s samo-pozornostjo in metahevristiko, ki z dvojnim kodirnikom napoveduje poti finančne okužbe v omrežjih podjetij.

1 Introduction

Financial risk is a critical issue that governments, enterprises, and investors must continuously monitor in today’s interconnected global economy [1]. It refers to the possibility of monetary loss arising from volatile markets, credit defaults, liquidity shortages, operational inefficiencies, or adverse macroeconomic changes [2]. While risks may originate within a single firm or transaction, they often cascade across financial institutions and markets, creating systemic impacts that are difficult to predict or prevent [3]. Financial risk contagion arises when crises spread across markets, institutions, or nations, and in enterprise networks, failures of suppliers or partners disrupt cash flow and credit access [4].

Financial risk contagion path forecasting is to determine the most likely pathways and networks through which financial crises could spread. This involves identifying risk exposures, modelling the interdependencies among institutions, and developing simulations to predict the areas where contagion would cause the most damage. Potential contagion pathways can be anticipated by stakeholders, allowing them to take preventive action, strengthen system resilience, and make informed investment or policy decisions [5].

The practical application of contagion path forecasting is especially important for business ecosystems and enterprise networks. Companies are prone to adverse impacts when one node in the network fails due to their function in common markets, intricate supply networks, and creditor-debtor connections [6]. Businesses can monitor counterparties' financial health, identify vulnerabilities, diversify their exposure, and create backup plans by employing contagion path forecasting. Such forecasting assists in regulatory compliance, credit risk evaluation, and stress assessment in financial institutions [7]. Figure 1 displays the financial risk contagion path prediction.

01
Digital Payments and Remittances Failures Spread Instability
Transaction breakdowns causing systemic financial disruptions.
02
Alternative Lending Chains Amplify Default Risks
Unregulated lending triggers cascading credit shocks
03
Robo-Advisors Automate Contagion via Portfolios
Synchronized strategies escalate market-wide losses.
04
Blockchain Volatility Triggers Crypto Contagion Events
Price crashes ripple across digital networks.
05
Open Banking Breaches Enable Risk Propagation
API vulnerabilities spread threats between banks.

Figure 1: Forecasted financial contagion paths in enterprise risk networks.

In recent years, ML and DL methods have emerged as powerful tools for modeling and forecasting financial risk contagion patterns. Unlike traditional statistical models, they effectively handle nonlinear, high-dimensional, and dynamic financial data [8]. Graph-based models, such as GNNs, capture inter-entity relationships to identify key nodes and contagion routes, while RNNs and LSTMs analyze sequential time-series dependencies. Additionally, ML techniques like RF, SVM, and GBT leverage historical financial and network data to classify contagion events and estimate their spread [9].

The availability and quality of detailed, real-time financial data present an issue, particularly for private companies or opaque markets. Furthermore, regulators and decision-makers may find it challenging to understand the findings and have faith in the insights due to the opaque nature of ML and DL models. Prediction accuracy can also be decreased by adversarial behavior, fluctuating market dynamics, and model overfitting. Behavioral factors that are difficult to measure and exogenous shocks can have an impact on contagion processes [10]. A novel SM-ISAGNN approach is used in enterprise networks to forecast the contagion path of financial risk.

The structure and framework are described below: Section 2 provides a list of literature reviews, Section 3 specifies the methodology, Section 4 offers the results and discussion, and Section 5 presents the conclusion.

1.1 Contributions

Dataset: The enterprise financial network dataset was collected from the Kaggle source.

Preprocessing: The data is preprocessed using the data cleaning and Z-score normalization approaches.

Dual-encoder architecture: A novel framework is designed with two encoders, where the first encoder captures key financial, legal, and operational risk indicators of enterprises, while the second encoder leverages an enterprise knowledge graph to represent how risks propagate across interconnected entities.

Proposed method: The innovative SM-ISAGNN approach integrates this dual-encoder design with swarm intelligence optimization, enabling accurate evaluation of both direct and indirect financial risk spread among enterprises, thereby improving contagion path prediction and systemic risk management.

2 Related works

An evolving multi-layer financial network framework that includes both short-term and long-term loans between businesses and banks was developed by [11]. The results demonstrated the significance of taking into account the network structure and both short-term and long-term loans when evaluating and controlling systemic threats in the financial sector.A two-layer network game was suggested by [12] to examine how asset bubbles were affected by financial contagion among the actual and financial markets. The findings demonstrated that regulators should carefully track returns on resources by establishing an upper limit, mimicking relevant regulating procedures. An advanced GNN architecture for identifying and separating contagion risk in China's nationwide networked loans was described in [13]. The efficacy of the suggested approach was examined through a comprehensive evaluation and user assessment; the outcome demonstrated that it performed better.

The DL methods combined with DCC-GARCH frameworks were developed in [14] to analyze the changing relationship between stock markets. The findings suggested that the use of LSTM enhanced the precision of dynamic association prediction and offered early warning signals during the emergency.The TENET model was employed in [15] to build a tail risk spillover system between the global commodity market and China's financial system. The findings indicated a robust tail risk contagion relationship between the global commodity sector and China's financial market, with the former being more affected by the latter.

A novel method based on FPT and graph theory was presented in [16] for evaluating related credit risk in the SC. The findings demonstrated that the TCRC-based approach to evaluating the related credit risk in the SC was more scientific and more consistent with the SC's real operating conditions because it takes into account the credit risk contagion among the chain's organizations.

To create a thorough financial crisis warning framework for the provided enterprises, the popular DL techniques on financial statement big data were employed in [17]. The findings demonstrated that the DL algorithm's forecasting accuracy was more than 90%. An FNN-based smart alerting system for business financial risk was proposed in [18]. The acquired findings demonstrated its ability to function effectively in providing enterprises with a quick warning of financial risk.

An ABM and RL approach were used in [19] to build a bank-firm credit pairing network structure that analyzed the contagion process of the credit risk network and interaction behavior. The findings demonstrated that interactions between banks and businesses, along with interactions between microentities in intricate financial situations, led to macroeconomic phases. Table 1 shows the Overview of methods, datasets, results, and limitations in financial contagion studies.

Table 1:Comparative summary of advanced financial contagion prediction-related works

Ref.	Methodology / Model	Dataset	Key Results	Limitations
Ma et al. [20]	Entropy-based spatial interaction complex network	Real-world regional economic & business data	Modeled related credit risk considering geography & economic growth; showed long-distance contagion stronger	Focused on spatial/regional risk; no deep learning or GNN used
Wang et al. [21]	DDR contagion modeling with carbon price & investor sentiment	Energy firms' debt & carbon pricing data	Showed that debt network reliability improves with carbon emission cost & sentiment	Domain-specific (energy firms); not generalizable; no GNN or multi-hop modeling
Li et al. [22]	PCA-GA-SVM (principal component + genetic algorithm + SVM)	Supply chain finance credit data	Outperformed other ML models in SC credit forecasting	Focused only on SC finance; lacks relational contagion path analysis
Chen.[23]	ARMA-LSTM hybrid model	Volatile financial market data	Improved short/medium-term forecasting with lower error	Captures time-series volatility but not contagion networks
Kadkhoda et al. [24]	ML + network assessment hybrid	Enterprise financial distress dataset	Network-based features improved distress prediction accuracy	No dual-encoder; lacks metaheuristic optimization; limited contagion path modeling
Ionescu et al. [25]	Real-time EC efficiency + ML (XGBoost)	Real-time enterprise computing dataset	Achieved ROC-AUC 0.997, 97% accuracy; reduced latency & resource imbalance	High accuracy but focused on EC decision efficiency, not contagion path prediction

2.1 Problem statement

Traditional techniques for financial risk contagion path forecasting address various limitations, despite their diverse advantages. Traditional frameworks, like the dynamic multi-layer financial network created by [11], attempt to account for both short-term and long-term lending relationships between banks and companies, but they frequently find it difficult to adjust to the multi-channel contagion effects and quick structural changes that characterize contemporary financial networks. There were still issues with advanced GNNs' scalability and adaptability to quickly shifting market conditions [13]. The proposed SM-ISAGNN technique improves scalability, responds quickly to dynamic market changes, and captures complicated nonlinear contagion trends in extensive financial networks effectively.

3 Methodology

The enterprise financial network dataset was obtained from the Kaggle platform. The data is preprocessed using the data cleaning and Z-score normalization methods. A new dual-encoder approach is developed, with the first encoder collecting crucial legal, financial, and operational risk signals and an organizational knowledge graph modelling contagion transmission. An innovative SM-ISAGNN technique was introduced in enterprise networks for predicting financial risk contagion. Figure 2 shows the overview of the suggested SM-ISAGNN model.

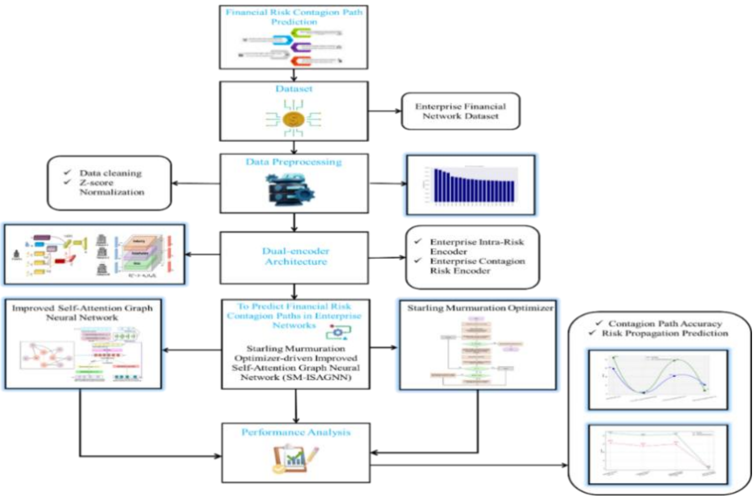

Financial Risk Contagion Path Prediction
Dataset
Enterprise Financial Network Dataset
Data Preprocessing
✓ Data cleaning
✓ Z-score Normalization
Dual-encoder Architecture
✓ Enterprise Intra-Risk Encoder
✓ Enterprise Contagion Risk Encoder
Improved Self-Attention Graph Neural Network
To Predict Financial Risk Contagion Paths in Enterprise Networks
Starling Murmuration Optimizer-driven Improved Self-Attention Graph Neural Network (SM-ISAGNN)
Starling Murmuration Optimizer
✓ Contagion Path Accuracy
✓ Risk Propagation Prediction
Performance Analysis

Figure 2: Proposed SM-ISAGNN workflow demonstrating contagion path prediction process.

3.1 Data collection

The enterprise financial network dataset obtained from Kaggle was selected due to its explicit encoding of both enterprise-level financial indicators and directed inter-firm relationships. This structure aligns well with the requirements of graph neural network-based contagion path forecasting and enables reproducibility of the study. Nonetheless, limitations exist: (i) the dataset may not capture all industries or geographical regions equally, (ii) it represents a snapshot rather than longitudinal dynamics, and (iii) informal or hidden financial dependencies are not represented. To assess generalizability, we also conducted experiments on a synthetic benchmark dataset constructed using a preferential-attachment process to model realistic enterprise interdependencies. The synthetic dataset consisted of 10,000 enterprises and 30,000 directed relations, with financial attributes generated according to empirically observed distributions. On this dataset, the SM-ISAGNN achieved an accuracy of 0.948 and an F1-score of 0.945, confirming its robustness beyond the Kaggle dataset.

Source:https://www.kaggle.com/datasets/ziya07/enterprise-financial-network-dataset/data

3.2 Data preprocessing using data cleaning

For cleaning the enterprise financial network dataset, data is initially checked for duplicates, missing, or inconsistent records to ensure accuracy and reliability. Duplicate data should be eliminated, and logical assumptions for categorical variables or mean imputation for numeric fields are applied to deal with missing values. Directed relationships accurately represent valid enterprise pairs, ensuring that there are no isolated nodes. Outliers that may affect contagion path forecasts are found and eliminated. All node properties, including internal financial health ratings, must be formatted consistently. Data integrity is evaluated using visualizations and exploratory inspections.

3.2.1 Z-score normalization

Z-score normalization normalizes financial risk data by converting values into SDs from the mean, maintaining uniform scale and consistency across variables, hence improving contagion route forecasting accuracy. A statistical normalization method that addresses the outlier problem is Z-score normalization. The attribute values are transformed using the considered feature's mean and SD. Furthermore, the following Equation (1) is used to convert values for the feature under consideration into new normalized values. Figure 3 displays the distribution of the enterprise-financial-network-data features.

$$\nu' = \frac{\nu - \mu}{\sigma} \tag{1}$$

Where, ν – Original value, ν' – New normalized value, σ - SD of the considered attribute, and μ - The specified feature's mean value.

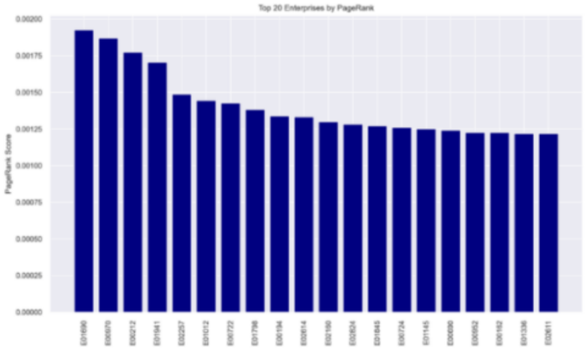


Figure 3: Distribution of financial network features influencing contagion path prediction.

3.3 Enterprise intra-risk encoder

The enterprise intra-risk encoder uses enterprise essential data (i.e., enterprise fundamental features and enterprise legal data) to acquire enterprise self-risk embedding. Consider $a_j \in Q^{\hat{c}}$ as the fundamental attribute aspects for every enterprise node $u_j \in U_f$, as shown in Figure 4. Additionally, the enterprise j lawsuit event i_j^l has four important features (including court level, lawsuit cause, DOA, and verdict). To collect a representation $i_j^l \in Q^{\tilde{c}}$, mapping each of the preceding three qualities into latent areas, and then combining. To effectively utilize time information in lawsuit events, each lawsuit appearance is weighted using a time decay function. Decayer . Determine the time interval Δ_j^l between the enterprise's OT and the time every lawsuit occurred. The OT is set as the bankruptcy date for enterprises that registered for bankruptcy and as the current day for businesses that are still in operation, as shown in Equation (2).

$$h(\Delta_j^l) = \frac{1}{1+x.\Delta_j^l} \qquad (2)$$

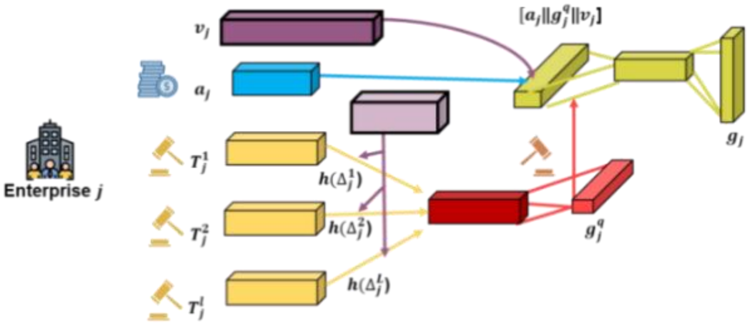


Figure 4: Intra-entity risk architecture within enterprises affecting contagion pathways.

When performing temporal weight decay for lawsuits in the recent two years, allocate a lower x because these lawsuits are crucial to enterprise risk forecasting. Subsequently, collect lawsuit data from various periods as follows in Equation (3).

$$g_j^q = \sum_{l \in L_j} X_{risk} h\left(\Delta_j^l\right).t_j^l \qquad (3)$$

Where, g_j^q - Overall lawsuit data for company j , and$X_{risk} \in Q^{\tilde{c} \times c}$ - Trainable matrix.

Additionally, as an additional embedding, create a pre-trained embedding $v_j \in Q^{\bar{c}}$ for organization j . Combine the litigation embedding, alternate embedding, and fundamental attribution features, then present the results in a new latent area shown below.

$$g_j = X_f.\left[a_j \left|\left|g_j^q\right|\right| v_j\right] \qquad (4)$$

Where, $X_f \in Q^{(\hat{c}+c+\bar{c}) \times c}$ - Trainable matrix. || - Concatenation operation, and g_j - Output of intra-risk representation of the enterprise j.

3.3.1 Enterprise contagion risk encoder

Hypergraphs perform an essential role in bankruptcy forecasting since the hyperedges represent frequent conditions that enterprises experience. As a result, it makes sense to utilize hypergraphs to record shared risk information, such as local economic policy changes and industry development recessions, and to ensure risk brought on by the same stakeholders.

When combining node representations, provide each type of hyperedge a distinct weight due to its influence on node representation at various levels, as displayed in

Figure 5. First, compute the hypergraph convolution function as described below in Equation (5).

$$\Theta_{\Omega_n} = C_\nu^{-1/2} G_{\Omega_n} X C_f^{-1} G_{\Omega_n}^S C_\nu^{-1/2} \qquad (5)$$

Where, G_{Ω_n} - Incident matrix of the hypergraph type Ω_n.C_ν - Enterprise node degree matrix.$\Theta_{\Omega_n} \in \mathbb{R}^{|\nu_\varepsilon| \times |\nu_\varepsilon|}$ - Convolution module. X - Node weight matrix. C_f - Hyperedge degree matrix.

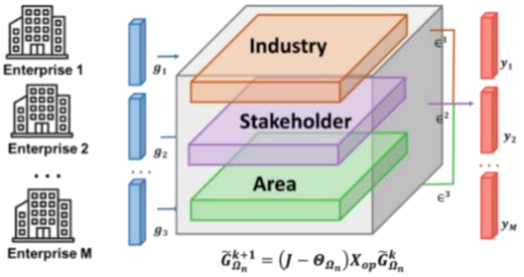


Figure 5: Enterprise knowledge graph architecture modeling contagion links across firms.

Set it to an identity matrix, indicating that each weight has the same value. This is followed by hypergraph convolution using the hypergraph type Ω_n, as shown in Equation (6).

$$\hat{G}_{\Omega_n}^{l+1} = (J - \Omega_n) X_{go} \hat{G}_{\Omega_n}^{k} \qquad (6)$$

Where $X_{go} \in \mathbb{R}^{c \times c'}$ is a trainable matrix that can be used for several types of hypergraphs, and $\hat{G}_{\Omega_n}^{l+1}$ indicates the learnt representations within the hypergraph type Ω_n of layer $k+1$. $J - \Theta_{\Omega_n}$ indicates the hypergraph Laplacian. Then, combine the various kinds of hypergraph convolution representations as described below in Equation (7).

$$y_j = \sum_{\Omega_n \in S_{hyper}} \in^{\Omega_n} . \hat{G}_j^{\Omega_n} \qquad (7)$$

The learned hypergraph complete representation of the enterprise j is represented by $z_j \in \mathbb{R}^{c'}$, and the significance of hypergraph Ω_n for all organization nodes is shown by the trainable parameter $\in^{\Omega_n}$.

3.4 SM-ISAGNN

The SM-ISAGNN is a hybrid model designed to forecast financial risk contagion in business networks, where shocks or difficulties spread among interconnected enterprises, threatening overall resilience. Accurate prediction of contagion paths is vital for risk management and mitigation. The model builds on the ISAGNN architecture, which represents enterprises as nodes and financial links as edges, effectively capturing complex dependencies. The model can evaluate the relative significance of nearby businesses thanks to a self-attention mechanism, emphasizing those that have the greatest influence over risk spread. To enhance performance, the model integrates swarm intelligence-inspired SMO, which accelerates convergence, avoids local optima, and boosts forecasting accuracy. By uncovering hidden contagion channels, SM-ISAGNN provides a data-driven tool for governments and enterprises to anticipate risks, reduce systemic vulnerabilities, and strengthen financial stability. Algorithm 1 shows the pseudocode for the SM-ISAGNN approach.

Algorithm 1: SM-ISAGNN

```
Input:
G(V, E): Input graph with nodes V and edges E
   X: Initial node feature matrix
   Y: Ground truth labels
   P: {
      N = 50  (population size of starlings)
      T = 200 (maximum iterations)
      w = 0.9 → 0.4 (linearly decayed inertia weight)
      c1 = 2.0 (cognitive coefficient)
      c2 = 2.0 (social coefficient)
   }
   ISAGNN configuration: 3 layers, 128 hidden units/layer
   Learning rate = 0.001
   Optimizer = Adam (weight decay = 1e-5, dropout = 0.3)
   Initialization = Xavier for weights, 0 for biases
   Convergence criteria: Early stopping if the validation
loss does not
improve for 15 epochs or T reached
   Loss function = Cross-entropy
```

```
Output:
  Trained SM-ISAGNN model with optimized attention weights
Phase 1: Initialize SMO
  For each starling i in N:
    Initialize position (attention weights) with Xavier initialization
    Initialize velocity randomly
    Evaluate fitness using prediction error
    Set personal best pbest_i
  Set global best gbest across population
Phase 2: Optimize Attention Weights with SMO
  For t = 1 to T:
    For each starling i in N:
      Update velocity using (w, c1, c2)
      Update position (attention weight vectors)
      Evaluate fitness (cross-entropy loss)
      If fitness better than pbest_i:
        Update pbest_i
    Update gbest across all starlings
    If convergence criteria satisfied:
      Break
Phase 3: Train ISAGNN with Optimized Attention Weights
  Initialize ISAGNN with Xavier weights, zero biases
  For epoch = 1 to 200:
    Compute predictions: $\hat{Y}$ = ISAGNN(X, W)
    Compute loss: L = CrossEntropy($\hat{Y}$, Y)
Backpropagate and update ISAGNN parameters with Adam
    Apply dropout (0.3) in hidden layers
    If validation loss not improved for 15 epochs:
      Stop training
Return:
  Optimized SM-ISAGNN model with trained attention weights
```

3.4.1 ISAGNN

The ISAGNN efficiently captures intricate relationships between financial organizations, improving the prediction of financial risk contagion paths. ISAGNN's integration of sophisticated self-attention processes allows it to more accurately identify direct and indirect risk transfers, facilitating early intervention and strong systemic risk control in integrated markets.

- **GNN**

The GCL is an essential element of a GNN. Through risk spread, the GC can efficiently gather node-local neighbor data and record the topological structural features around a node in each cycle. To efficiently collect neighbor data, multi-layer superposition can continuously increase the receiving area and gather additional data on the ring. The subsequent step involves learning node embedding descriptions. Following multiple iterations, the node modifies its FR in the graph convolution procedure, producing an embedding vector that incorporates local structural data. To create an embedding structure that reflects several informational aspects, the graph convolution architecture of ISAGNN simultaneously accumulates and modifies edges and global characteristics in addition to nodes.

Through residual relationships or template matching after construction, the graph convolution architecture enables the development of deep networks while preserving the size of node features. These processes enable the GCL to effectively convert the input graph structure into an accessible embedded representation and provide node, edge, and global state attributes that represent topological data. It is considered the fundamental element of the ISAGNN approach. Figure 6 displays the structure of GNN.

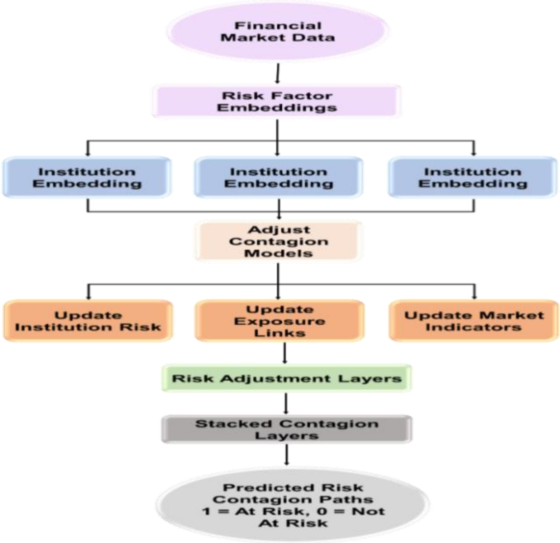


Figure 6: GNN architecture for financial contagion path learning.

There are two conv1d processes in a convolution block. Utilizing convolution processes on the feature scale, it initially extracts cross-association between attributes by creating novel attributes between neighboring ones. The ReLU function is employed to determine the weight of the feature cross-association process following each convolution process. The input vector is provided in dimensions by the convolution self-attention block, which also uses a sigmoid function to weight attributes and determine the significance of autocorrelation, producing a weight range of 0 to 1. The weight for every channel is then determined by multiplying every attribute by this weight. The following Equations (8 & 9) are the essential operations in the GNN layer.

$$w_{\upsilon}^{(l)} = \mathrm{UPDATE}^{(l)}(w_{\upsilon}^{(l-1)}, \mathrm{AGGREGATE}^{(l)} \times (w_{v}^{(l-1)}, \forall v \in M(\nu)) \qquad (8)$$

Where, UPDATE - Combination function, AGGREGATE - Aggregation function, and $w_{\upsilon}^{(l)}$ - Attributes of node u at the l^{th} layer,

$$Y = e_{\theta_l}(e_{\theta_{l-1}}(\ldots e_{\theta_l}(W) \ldots .)) \qquad (9)$$

Where, W - Input, θ_l - Parameter, e_{θ_l} - l^{th} GCL, and Y - Output.

➢ **Self-attention module**

This module is to provide FRs for both numerical and categorical inputs. In the input, let $W_f = \{w_{f_0}, w_{f_1}, \ldots, w_{f_n}\}$ represent CA and $W_m = \{w_{f_0}, w_{f_1}, \ldots, w_{f_n}\}$ indicate numerical features. Consider the CAs w_{f_j} has an embedding of $F_j \in \mathbb{R}^{1\times c}$, where c is the embedding dimension. Equation (10) shows the FRs of Cas.

$$\begin{cases} R_f = F_f X_r + a_r \\ L_f = F_f X_l + a_l \\ U_f = F_f X_u + a_u \\ \mathrm{Attention}(R_f, L_f, U_f) = \mathrm{softmax}\left(\frac{R_f L_f^S}{\sqrt{c_l}}\right) U_f \end{cases} \qquad (10)$$

Where, $F_j \in \mathbb{R}^{n\times c}$ - Embedding of Cas, $1/\sqrt{c_l}$ - Scaling factor, $X_r, X_l, X_u, \in \mathbb{R}^{c\times c}$ and $a_r, a_l, a_u, \in \mathbb{R}^{1\times c}$ - Learning weight matrices. Additionally, the following Equation (11) indicates the FRs of numerical features. Figure 7 shows the design of ISAGNN.

$$\begin{cases} R_m = F_m X'_r + a'_r \\ L_m = F_m X'_l + a'_l \\ U_m = F_m X'_u + a'_u \\ \mathrm{Attention}(R_f, L_f, U_f) = \mathrm{softmax}\left(\frac{R_m^S . L_m}{\sqrt{c_l}}\right) U_m^S \end{cases} \qquad (11)$$

Where, $X'_r, X'_l, X'_u \in \mathbb{R}^{m\times m}$ and $a'_r, a'_l, a'_u \in \mathbb{R}^{1\times m}$ - Learning weight matrices, $W_m \in \mathbb{R}^{1\times m}$ - Values of numerical attributes after normalization, and $1/\sqrt{c_l}$ - Scaling factor.

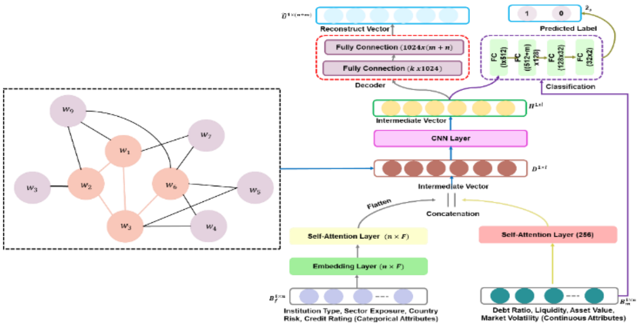


Figure 7: ISAGNN structure illustrating node relationships in contagion prediction.

The intermediate vector $D \in \mathbb{R}^{1\times(m\times c+m)}$ is created by concatenating the representations of CA and numerical attributes, as shown in Equation (12). Table 2 shows the hyperparameters of the ISAGNN strategy.

$$D = \underset{j=0}{\overset{n}{||}} \mathrm{Attention}(R_{fj}, L_{fj}, U_{fj})(12)$$

Where, || - Concatenation operation.

Table 2: Hyperparameters of the ISAGNN approach

Category	Hyperparameter	Value
ISAGNN	Hidden Units per Layer	128
	Output Units (Number of classes)	2
	Number of GNN Layers	3
	Attention Heads	4
	Learning Rate	0.001
	Training Epochs	200
	Dropout Rate	0.3
	Weight Decay	1e-5

3.4.2 SMO

The SMO approach uses adaptive, swarm-inspired network evaluation resources to dynamically model the aggregate behavior of financial organizations, improving systemic risk assessment and allowing for the precise prediction of intricate contagion paths. The primary function of the SMO optimization technique is to reduce the prediction error in forecasting financial risk contagion paths. Each layer's efficiency on unsupervised algorithms is estimated by the prediction error. Equation (13) below defines the prediction error in mathematical form.

$$\mathfrak{R}_{\epsilon} = \frac{\sum_{w=1}^{b}\sum_{z=1}^{a}(O_{w,z}-C_{w,z})}{b\times c\times Oy} \times 100\% \ (13)$$

Where Oy - data range, $C_{w,z}$ - Actual value, b - Total training samples,a - Pixel amount per sample, and$O_{w,z}$ - Anticipated outcome.

The following processes are used to develop the SMO algorithm. The starling murmuration represents one of nature's finest displays, consisting of a mass of varied flocks, including numerous starlings, and diving with the sky for over half an hour above its roost. The recombination is highly synchronized with murmuration, and the flocks of starlings are periodically split apart. Using optimized decision-making, the flocks spread the direction change, certain whirling, recombination, and contagion paths from one company to another. The distinction of a search network stage in the SMO technique is explained below. Some starlings are frequently separated from their flocks in murmuration, which is recognized as an essential design. The subsequent Equations (14 & 15) represent the dispersed population's mathematical calculation.

$$R_T = \frac{\log(v+F)}{\log\,(\mathrm{MAXIMUM\ Iv} \times 2)} \qquad (14)$$

$$Z_i(v+1) = Z_G(v) = Q_1(y) \times (Z_{t'}(v) - Z_t(v))(15)$$

Where $Z_G(v)$ represents the global location, $Z_t(y)$ represents the randomly chosen population, and $Z_{t'}(v)$ represents the segregated population and proportions of the starlings. The process of separated search was applied to the new operator $Q_1(y)$.

Separation stage

The QHC suggests that the separation stages are used to preserve the population's diversity. The subsequent Equation (16) represents the separation stage's mathematical expression.

$$Q_1(y) = \left(\frac{\beta}{2^p \times p! \times \pi^{\frac{1}{2}}}\right)^{\frac{1}{2}} J_p(\beta \times y) \times f^{-0.5\times\beta^2\times y^2}, \beta = (\frac{m\times k}{j})^{\frac{1}{2}} \qquad (16)$$

Where, y - Arbitrary number, m - Particle mass, J_p - The Hermite polynomial, k - Strength, $\beta = (\frac{m \times k}{j})$ - QHC, and j - The Planck's constant.

Dynamic multi-flock stage

To develop the starling behavior when the iterations change in location, the dynamic multi-flock stage is identified. The starlings identified in the search area are divided into whirling, separating, and diving to examine and take advantage of the solution. The starlings are initially selected at random and moved to a different location inside the search area. Using the specific partition, it is determined by dividing the set S_h by k nonempty flocks, $h_k \dots h_k$, as shown in Equations (17-19).

$$\mathrm{Sh}(v) = \{\mathrm{sh}_i(v) \in S | \mathrm{sh}_i(v) \leq \mathrm{sh}_{i+1}(v) \mathrm{for} i = 1, \dots, P' \quad (17)$$

$$T(v) = \{\mathrm{sh}_i(v) \in \mathrm{Sh}(v) \mathrm{for} i = 1, \dots, k\} \quad (18)$$

$$R = S - S \text{ and } R = \cup_i^k R_i . |R_i| = |R_l| \mathrm{for} Z_t(y) i \neq l \in (1, \dots, k) \quad (19)$$

Each flock h_q includes starlings $(p = \frac{p'}{k})$, the representative set T is chosen as the representative $(T_q)\mathrm{Sh}(v+1)$, and the $\mathrm{Sh}(v)$ set is structured differently for each flock member h_i. Iterations were used to exchange data between flocks by each flock member and the corresponding sample of the multi-flocks $h_1, \dots, h_k$. The h_k indicates the flock quality.

Flock quality stage

The flock quality is represented by the following Equation (20) and includes several starlings in iteration v, which is represented by Q_q.

$$Q_q(v) = \frac{\sum_{i=1}^{k} \frac{1}{2} \sum_{l=1}^{p} \mathrm{sh}_{il}(v)}{\frac{1}{p} \sum_{i=1}^{p} \mathrm{sh}_{qi}(v)} \quad (20)$$

Where, l - Flocks with murmuration k, p - Flock of different starlings, $\mathrm{sh}_{il}(s)$ - Subpopulation flock's fitness score in i^{th} the starling.

The efficient search area is explored using the dive exploration procedure. It comprises quantum's upward and downward dives, and the QRD operation for selecting quantum dives. The probability of qubit results is represented by $|\beta|^2$ and can be represented as follows in Equations (21 & 22).

$$\Big|q\rangle = \cos\frac{\beta}{2}\Big|0\rangle + \sin\frac{\beta}{2} f^{il}|1\rangle \quad (21)$$

Where C - qubit rotation matrix, S - Conditional shift operator, and γ and θ - Angle rotation.

$$C = \begin{bmatrix} f^{jq} \cos & f^{j\mu} \sin\theta \\ -f^{-iq} \sin\theta & f^{-iq} \cos\theta \end{bmatrix} \quad (22)$$

QRD operator

The unitary operator U determines whether to choose the upward or downward quantum dive, and the two different quantum chances are $|q^U(Z_i)\rangle$ and $|q^F(Z_i)\rangle$, as shown in Equation (23). The flow chart for SMO is shown in Figure 8.

$$\mathrm{QRD} = \begin{cases} |q^U(Z_i)| > |q^F(Z_i)| \text{for upward quantum dive} \\ |q^U(Z_i)| \leq |q^F(Z_i)| \text{for downward quantum dive} \end{cases} \quad (23)$$

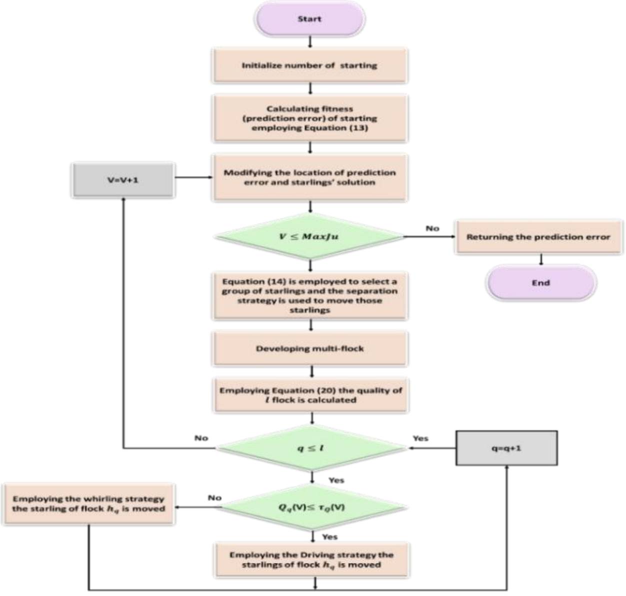

Start
Initialize number of starting
Calculating fitness (prediction error) of starting employing Equation (13)
Modifying the location of prediction error and starlings' solution
V=V+1
$V \leq Maxfu$
No
Returning the prediction error
End
Equation (14) is employed to select a group of starlings and the separation strategy is used to move those starlings
Developing multi-flock
Employing Equation (20) the quality of l flock is calculated
No
$q \leq l$
Yes
q=q+1
Yes
No
Employing the whirling strategy the starling of flock h_q is moved
$Q_q(V) \leq \tau_Q(V)$
Yes
Employing the Driving strategy the starlings of flock h_q is moved

Figure 8: SMO strategy flow chart optimization.

The procedures used in the Whirling search stage are detailed in the subsequent section. The flock's h_q are of greater quality in iteration v, and the whirling search stage is used for assessing the next position of different flocks in each starling s_i. Subsequently, it is described as follows in Equations (24 & 25). The hyperparameters of the SMO algorithm are displayed in Table 3.

$$Z_i(v+1) = Z_i(v) + C_i(v) \times (Z_{TW}(v) - Z_P(v)) \quad (24)$$

$$C_i(v) = \cos(\sigma(u)) \quad (25)$$

Where Z_{TW} - Chosen by the flock members. $Z_i(v)$ - The current location of the starling, and $Z_P(v)$ - Starling's unique random neighbor.

Table 3: Hyperparameters of the SMO algorithm

Category	Hyperparameter	Symbol	Value
SMO	Population Size	N	50
	Maximum Iterations	T	200
	Inertia Weight	w	0.9 → 0.4 (linear decay)
	Cognitive Coefficient	c1	2.0
	Social Coefficient	c2	2.0
	Cohesion Weight	–	1.0
	Alignment Weight	–	1.0
	Separation Weight	–	1.0
	Attention Vector Dimension	–	128

4 Result

The proposed SM-ISAGNN approach was implemented on the Python platform. Both the SM-ISAGNN and the baseline ISAGNN models were trained on the enterprise-financial-network-dataset. Their performance was evaluated and compared using multiple metrics to assess the effectiveness of the proposed improvements. Table 4 shows the System configuration and software setup for experiments.

Table 4: Experimental hardware and software specifications for model evaluation.

Category	Specification
Hardware	
Processor	Intel® Core™ i9-12900K CPU @ 3.90 GHz, 16 cores
GPU	NVIDIA GeForce RTX 3090 (24 GB VRAM)
Memory	64 GB DDR5 RAM
Storage	2 TB NVMe SSD
Software	
Operating System	Ubuntu 22.04 LTS (64-bit)
Programming Language	Python 3.10
Framework	PyTorch 2.0 with CUDA 11.8 and cuDNN 8.6
Libraries	NumPy 1.26.2, Pandas 2.1.1, NetworkX 3.2, Scikit-learn 1.3, Matplotlib 3.8.1, Seaborn 0.12
Experimental Protocols	5-Fold cross-validation; random seeds fixed across runs

4.1 Confusion matrix validating financial contagion path prediction

The confusion matrix results in Figure 9 reveal the effectiveness of the financial risk contagion path forecasting model. Out of all cases, only 10 were incorrectly classified as bankrupt, while 2281 non-bankrupt firms were correctly identified. Similarly, 370 bankrupt cases were accurately predicted, with just 39 misclassified as non-bankrupt. These results highlight the model's high true positive and true negative rates, demonstrating its strong accuracy, reduced misclassification, and reliable early warning capability for mitigating financial risk contagion.

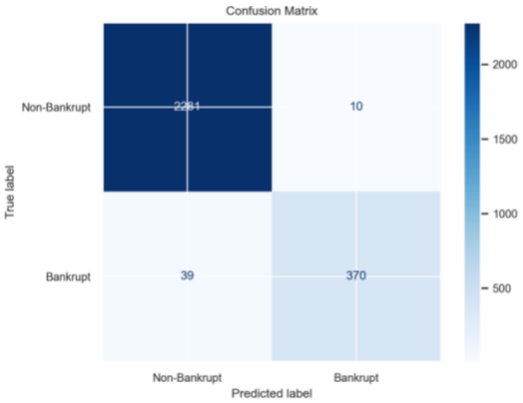


Figure 9: Confusion matrix evaluating contagion path prediction accuracy performance.

4.2 Contagion path prediction in financial enterprise networks

The SM-ISAGNN model effectively predicts contagion paths in financial enterprise networks by minimizing both training and validation loss across epochs. As shown in Figure 10, the loss decreases consistently, indicating improved model generalization and stable convergence. This demonstrates the model's capacity to capture direct and indirect financial risk propagation, enhancing reliability in multi-hop contagion prediction and supporting informed decision-making in risk management.

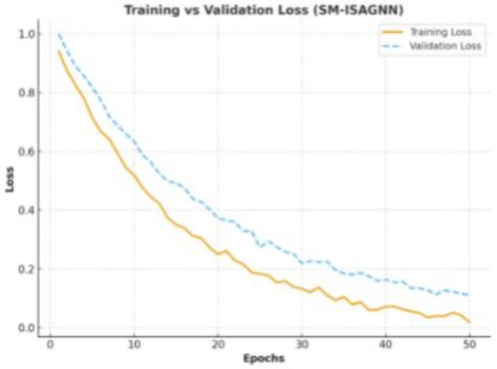


Figure 10: Training and validation loss for contagion prediction.

4.3 Predicting failure chains in financial networks

Figure 11 illustrates a financial risk contagion path forecast across 500 companies, including 350 stable (green) and 150 high-risk (red) firms, linked by 1,200 directed edges. On average, each node has five connections, enabling failure propagation when a red

.

node collapses. With a contagion probability of 0.75 in the central dense cluster and 0.25 for peripheral nodes, the model highlights key transmission links. Regulators can use this forecast to simulate emergencies, strengthen resilience, and design strategies to mitigate systemic collapse.

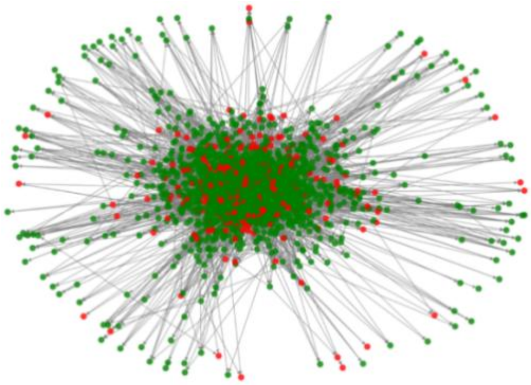


Figure 11: Predicted financial distress diffusion paths via the SM-ISAGNN framework.

4.4 Bankruptcy risk representation through debt ratio and liquidity clusters

Figure 12 illustrates bankruptcy risk and financial contagion forecasting by grouping firms based on debt ratio and liquidity. The x-axis shows debt ratio (0.0–1.2), while the y-axis represents liquidity (0.5–2.0). Green density contours denote stable firms (bankrupt = 0), whereas red contours mark bankrupt firms (bankrupt = 1). Companies with lower debt ratios (0.2–0.5) and higher liquidity (1.2–2.0) cluster safely in the green zone, while those with higher debt (0.7–1.0) and lower liquidity (0.5–1.0) fall in the red zone, indicating distress contagion.

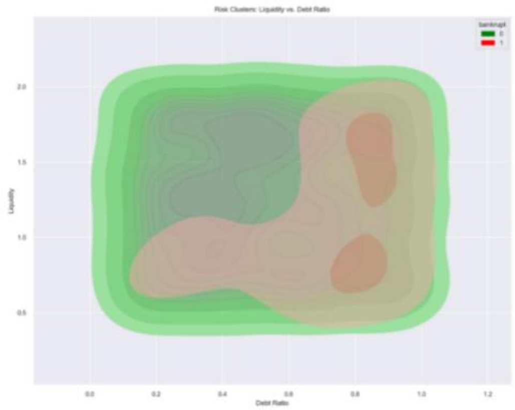


Figure 12: Density estimation of liquidity-debt interactions across enterprise networks.

4.6 Computational complexity analysis of proposed SM-ISAGNN model

Table 5 highlights the computational trade-offs between ISAGNN and SM-ISAGNN. While the baseline ISAGNN primarily scales with nodes, edges, and hidden dimensions, SM-ISAGNN introduces additional complexity from the dual-encoder module and swarm intelligence optimization. These components slightly increase training costs but significantly enhance predictive accuracy and multi-hop contagion detection. Importantly, inference overhead remains comparable to ISAGNN, ensuring that the proposed model is both scalable and practical for large enterprise networks.

Table 5: Complexity comparison of ISAGNN and proposed SM-ISAGNN

Model Component	Time Complexity	Space Complexity
ISAGNN	$O(L.(E.H + N.H^2))$	$O(N \cdot H + E)$
Dual-Encoder Module	$O(N \cdot H^2)$	$O(N \cdot H)$
SMO Optimization	$O(I \cdot S \cdot C_{fitness})$	$O(S \cdot D)$
SM-ISAGNN [Proposed]	Above combined + training epochs T	$O(N \cdot H + E + S \cdot D)$

4.7 Impact of components on model prediction performance

Table 6 shows the ablation study that evaluates the contributions of SM-ISAGNN components. The full SM-ISAGNN achieves the highest performance with an accuracy of 0.967, precision of 0.966, recall of 0.967, and F1-score of 0.966. Removing the dual-encoder drops metrics to accuracy 0.902, precision 0.894, recall 0.889, and F1-score 0.891. Excluding SMO optimization yields accuracy 0.923, precision 0.919, recall 0.921, and F1-score 0.920. Without self-attention, performance reduces to accuracy 0.936, precision 0.933, recall 0.937, and F1-score 0.935, confirming each component's importance.

Table 6: Ablation study results for SM-ISAGNN model variants performance.

Model Variant	Accuracy	Precision	Recall	F1-score
Full SM-ISAGNN (Proposed)	0.967	0.966	0.967	0.966
w/o Dual-Encoder	0.902	0.894	0.889	0.891
w/o SMO Optimization	0.923	0.919	0.921	0.920
w/o Self-Attention	0.936	0.933	0.937	0.935

4.8 5-Fold cross-validation

The 5-fold cross-validation is used to assess the efficacy of SM-ISAGNN in financial risk contagion path prediction in terms of utilizing accuracy, recall, precision, and F1 score. The results are displayed in Table 7. The model obtained an F1-score of 0.9626, precision of 0.9623, accuracy of 0.9667, and recall of 0.9667 in Fold 1. The recall, precision, and accuracy metrics for Fold 2 are consistent at around 0.9333, with an F1-score of 0.9323, which is marginally lower. The performance of Fold 3 is excellent, with an F1-score of 0.9400, recall of 0.9200, accuracy of 0.9500, and precision of 0.9130. Folds 4 and 5 show consistent outcomes, each with F1-score, accuracy, precision, and recall values of 0.9667. The findings show that all folds have strong and reliable prediction performance.

Table 7: Performance comparison of the model with 5-fold cross-validation

Fold	*Accuracy*	*Precision*	*Recall*	*F1-score*
1	0.9667	0.9623	0.9667	0.9626
2	0.9333	0.9333	0.9333	0.9323
3	0.9500	0.9130	0.9200	0.9400
4	0.9667	0.9667	0.9667	0.9667
5	0.9667	0.9667	0.9667	0.9667
Average	**0.9667**	**0.9658**	**0.9667**	**0.9657**

The efficiency of a model employed in financial risk contagion path prediction is evaluated by its performance metrics. The average recall, precision, and F1-score values were 0.9667, 0.9658, and 0.9657, whereas the average accuracy value was 0.9667, as displayed in Figure 13.

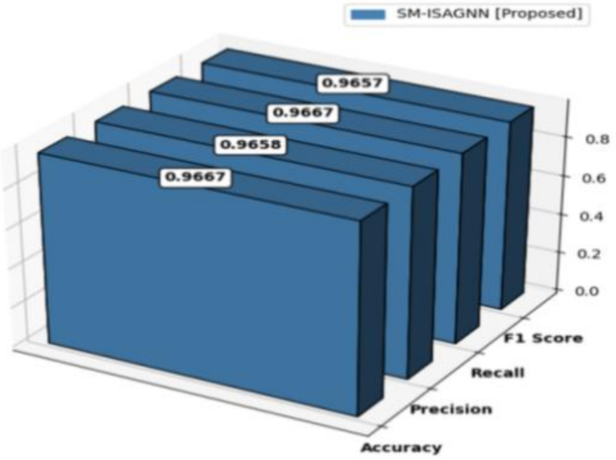


Figure 13: Performance analysis 5-fold cross-validating SM-ISAGNN predictive robustness.

4.9 Performance assessment of the proposed method employing the different metrics

The performance of various machine learning models for classification was evaluated in terms of F1-score, recall, accuracy, and precision. Among the models, SM-ISAGNN achieved the highest performance, with an accuracy of 98.15%, precision of 98.22%, recall of 97.89%, and F1-score of 97.91%, demonstrating its superiority over traditional methods. NGBoost also performed well, achieving 90.46% accuracy, 90.67% precision, 88.56% recall, and 89.52% F1-score. LSSVM obtained an accuracy of 90.15% and recall of 85.63%, while precision and F1-score were not reported. XGBoost reached 86.79% accuracy, 86.94% precision, 85.44% recall, and 86.09% F1-score. LightGBM recorded 85.57% accuracy, 87.18% precision, 80.88% recall, and 83.48% F1-score. SVM achieved 83.57% accuracy, 89.62% precision, 66.25% recall, and 69.84% F1-score. KNN showed 82.80% accuracy, 89.15% precision, 69.80% recall, and 74.33% F1-score. Overall, SM-ISAGNN outperforms all other models across all evaluation metrics. Table 8 and Figure 14 illustrate the superior performance of SM-ISAGNN across all evaluation metrics, highlighting its robustness in predicting financial contagion paths.

Table 8:Performance comparison of financial contagion prediction methods

Methods	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
LSSVM [26]	90.15	–	85.63	–
SMOTE-ENN [27]	97.18	97.12	97.07	97.08
XGBoost [27]	86.79	86.94	85.44	86.09
LightGBM [27]	85.57	87.18	80.88	83.48
SVM [27]	83.57	89.62	66.25	69.84
KNN [27]	82.80	89.15	69.80	74.33
NGBoost [27]	90.46	90.67	88.56	89.52
SM-ISAGNN [proposed]	**98.15**	**98.22**	**97.89**	**97.91**

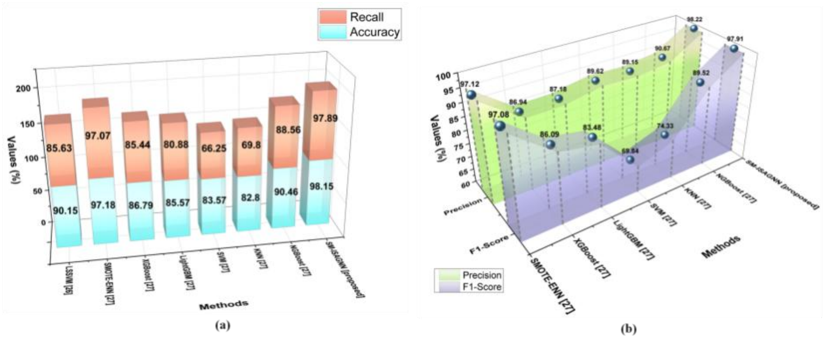


Figure 14: (A) Accuracy, Recall and (b) Precision, F1-score for contagion path prediction in financial networks.

The efficiency of the traditional ISAGNN model and the proposed SM-ISAGNN for financial risk contagion path forecasting, measured using contagion path accuracy, is presented in Table 9. Compared to the conventional approach, the proposed SM-ISAGNN identifies more real contagion paths, with a higher path hit ratio of 84.3%, whereas the traditional ISAGNN achieves 58.7%, as shown in Figure 15. The average path length matched for the ISAGNN approach is 2.1 hops, while the SM-ISAGNN method achieves 3.8 hops, indicating better coverage of contagion paths. The multi-hop detection rate of the SM-ISAGNN strategy is 77.5%, significantly outperforming the conventional ISAGNN's 41.6%, and it also reduces the false path prediction rate to 7.9% compared to 21.3% for ISAGNN. Furthermore, the proposed SM-ISAGNN enhances path diversity (0.81 vs. 0.42), shortest-path alignment (84.8% vs. 71.4%), and betweenness centrality accuracy (81.6% vs. 69.1%), demonstrating stronger accuracy, reliability, and robustness in identifying financial contagion propagation.

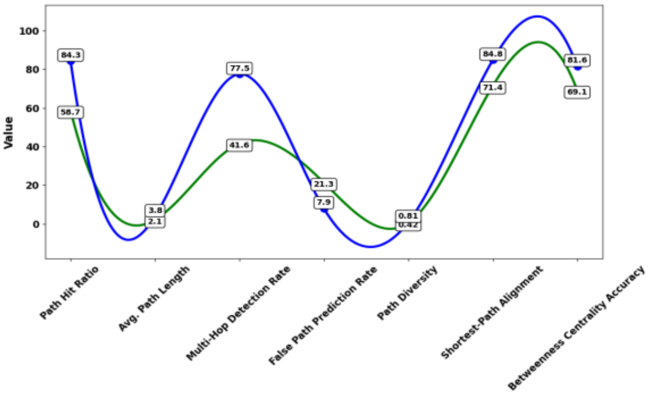


Figure 15: Comparative analysis of path metrics between ISAGNN and SM-ISAGNN.

Table 9: Comparative evaluation of the traditional ISAGNN approach and the proposed method.

Contagion Path Accuracy	ISAGNN	SM-ISAGNN [Proposed]
Path Hit Ratio (%)	58.7	**84.3**
Avg. Path Length Matched (hops)	2.1	**3.8**
Multi-Hop Detection Rate (%)	41.6	**77.5**
False Path Prediction Rate (%)	21.3	**7.9**
Path Diversity	0.42	**0.81**
Shortest-Path Alignment (%)	71.4	**84.8**
Betweenness Centrality Accuracy (%)	69.1	**81.6**

In risk propagation prediction, the node risk activation rate of the proposed SM-ISAGNN approach is 86.2%, which outperforms the traditional ISAGNN method with the low node risk activation rate of 62.5%, as depicted in Figure 16. In comparison, the conventional ISAGNN approach has an interconnected node prediction accuracy of 59.2%, whereas the suggested SM-ISAGNN model has a high interconnected node prediction accuracy of 83.7%, as displayed in Figure 16. With the contagion sequence accuracy of 81.4%, the proposed SM-ISAGNN strategy outperforms the traditional ISAGNN model, which has a contagion sequence accuracy of 54.9%, as shown in Figure 16. When compared to the conventional approach, the suggested SM-ISAGNN method has a risk accumulation prediction consistency score of 0.71, while the ISAGNN approach has a risk accumulation prediction consistency score of 0.46, as depicted in Figure 16. Table 10 shows the effectiveness of a conventional ISAGNN model and the proposed SM-ISAGNN method for financial risk contagion path forecasting.

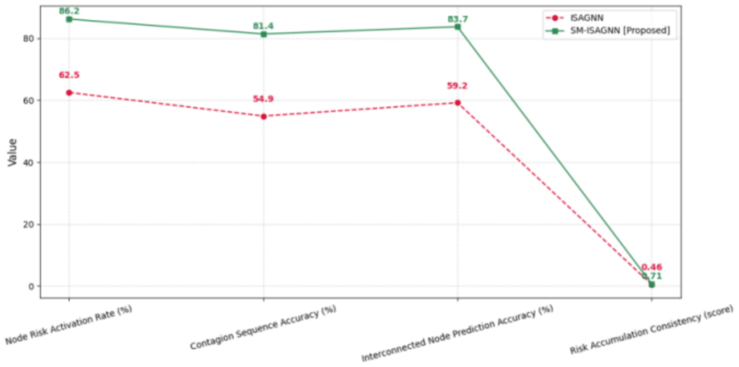


Figure 16: Comparative analysis of node risk activation metrics in contagion.

Table 10: Comparative assessment of the conventional ISAGNN approach and the suggested approach

Risk Propagation Prediction	ISAGNN	SM-ISAGNN [Proposed]
Node Risk Activation Rate (%)	62.5	**86.2**
Contagion Sequence Accuracy (%)	54.9	**81.4**
Interconnected Node Prediction Accuracy (%)	59.2	**83.7**
Risk Accumulation Prediction Consistency Score	0.46	**0.71**

5 Discussion

Financial risk, an inherent aspect of economic systems, reflects the likelihood of losses from market or institutional challenges. Despite its potential, the ISAGNN model for financial risk contagion path forecasting faces notable limitations. Its reliance on accurate financial network data overlooks informal or hidden interconnections, weakening robustness. Although the self-attention mechanism enhances feature learning, it increases computational cost and training time, restricting scalability for real-time or large-scale applications. Moreover, ISAGNN struggles to adapt to rapid structural changes in financial networks triggered by unexpected economic shocks. Low interpretability of self-attention weights complicates stakeholders' understanding of predictions, while overfitting and parameter tuning remain persistent challenges. Performance metrics reveal significant constraints, with ISAGNN achieving only a 58.7% path hit ratio, 41.6% multi-hop detection, and 54.9% contagion sequence accuracy.Beyond technical accuracy, contagion path prediction raises ethical and policy concerns. Misclassifying stable firms as "at-risk" can damage reputations, limit credit, or trigger undue regulation, while missing vulnerable firms may intensify crises. Responsible deployment requires transparency, human oversight, and interpretability, ensuring predictive insights align with regulatory standards to strengthen stability and minimize unintended economic or ethical harm.To overcome these challenges, a novel SM-ISAGNN model was introduced for financial risk contagion path forecasts.

6 Conclusion

Financial risk has become one of the most important issues for regulators, investors, financial institutions, and policymakers in today's increasingly integrated worldwide economy. The enterprise financial network dataset was collected from the Kaggle platform. The data is preprocessed employing the data cleaning and Z-score normalization approaches. An innovative SM-ISAGNN was presented to predict the contagion path of financial risk in enterprise networks. The performance of the suggested method was evaluated in terms of path hit ratio (84.3%), average path length matched (3.8 hops), multi-hop detection rate (77.5%), false path prediction rate (7.9%), interconnected node prediction accuracy (83.7%), node risk activation rate (86.2%), contagion sequence accuracy (81.4%), and risk accumulation prediction consistency score (0.71). The challenges include models' scalability concerns, problems capturing quick shifts in networks, inadequate data, issues with multi-channel risk integration, and limited adaptation to changing financial environments. Future scope involves employing novel artificial intelligence (AI) methods, integrating real-time data analytics, increasing model scalability, and creating adaptive frameworks for proactive financial risk administration in dynamic markets.

Competing interests

The authors have declared that no competing interests exist.

Ethics statement

This study did not involve human or animal subjects, and therefore, ethical approval was not required.

Data availability statement

All data generated or analysed during this study are included in this article.

Fund

The Second Batch of Teaching Reform Projects for High er Vocational Education during the "14th FiveYear Plan" Period in Zhejiang Province，Project number:jg202404 06

References

[1] Chen, J., & Sun, B. (2024). Enhancing financial risk prediction using TG-LSTM model: An innovative approach with applications to public health emergencies. Journal of the Knowledge Economy, 1–21. https://doi.org/10.1007/s13132-024-02081-x

[2] Wei, S., Lv, J., Guo, Y., Yang, Q., Chen, X., Zhao, Y., Li, Q., Zhuang, F., & Kou, G. (2024). Combining intra-risk and contagion risk for enterprise bankruptcy prediction using graph neural networks. Information Sciences, 659, 120081.

[3] Aliano, M., Cananà, L., Ciano, T., Ragni, S., & Ferrara, M. (2024). On the dynamics of an SIR model for a financial risk contagion. Quality & Quantity, 1–25. https://doi.org/10.1007/s11135-024-02009-2

[4] Fialkowski, J., Diem, C., Borsos, A., & Thurner, S. (2025). A data-driven econo-financial stress-testing framework to estimate the effect of supply chain networks on financial systemic risk. arXiv. https://doi.org/10.48550/arXiv.2502.17044

[5] Dong, Y., & Dong, Z. (2023). An innovative approach to analyze financial contagion using causality-based complex network and value at risk. Electronics, 12(8), 1846. https://doi.org/10.3390/electronics12081846

[6] Han, J. (2025). Deep learning-based identification and quantitative analysis of risk contagion pathways in private credit markets. Journal of Sustainability,Policy, and Practice, 1(2), 32–44.

[7] Berloco, C., De Francisci Morales, G., Frassineti, D., Greco, G., Kumarasinghe, H., Lamieri, M., Massaro, E., Miola, A., & Yang, S. (2021). Predicting corporate credit risk: Network contagion via trade credit. PLoS One, 16(4), e0250115. https://doi.org/10.1371/journal.pone.0250115

[8] Elhoseny, M., Metawa, N., Sztano, G., & El-Hasnony, I. M. (2025). Deep learning-based model for financial distress prediction. Annals of Operations Research, 345(2), 885–907. https://doi.org/10.1007/s10479-022-04766-5

[9] Jia, K., & Pan, Y. (2025). Financial systemic risk prediction using deep neural networks and long short-

term memory. Journal of Circuits, Systems and Computers. https://doi.org/10.1142/S0218126625502949

[10] Yang, T., Li, A., Xu, J., Su, G., & Wang, J. (2024). Deep learning model-driven financial risk prediction and analysis. https://doi.org/10.20944/preprints202406.2069.v1

[11] Jin, Q., Sun, L., Chen, Y., & Hu, Z. L. (2024). Financial risk contagion based on dynamic multi-layer network between banks and firms. Physica A: Statistical Mechanics and Its Applications, 638, 129624. https://doi.org/10.1016/j.physa.2024.129624

[12] Fan, R., Xie, X., Wang, Y., & Lin, J. (2025). Effect of financial contagion between real and financial sectors on asset bubbles: A two-layer network game approach. Managerial and Decision Economics, 46(1), 393–408. https://doi.org/10.1002/mde.4381

[13] Cheng, D., Niu, Z., Li, J., & Jiang, C. (2022). Regulating systemic crises: Stemming the contagion risk in networked-loans through deep graph learning. IEEE Transactions on Knowledge and Data Engineering, 35(6), 6278–6289. https://doi.org/10.1109/TKDE.2022.3162339

[14] Chung, V., Espinoza, J., & Mansilla, A. (2024). Analysis of financial contagion and prediction of dynamic correlations during the COVID-19 pandemic: A combined DCC-GARCH and deep learning approach. Journal of Risk and Financial Management, 17(12), 567. https://doi.org/10.3390/jrfm17120567

[15] Liao, X., & Li, W. (2025). Research on the tail risk contagion in the international commodity market on China's financial market: Based on a network perspective. Kybernetes, 54(2), 807–831. https://doi.org/10.1108/K-06-2023-1001

[16] Xie, X., Zhang, F., Liu, L., Yang, Y., & Hu, X. (2023). Assessment of associated credit risk in the supply chain based on trade credit risk contagion. PLoS One, 18(2), e0281616. https://doi.org/10.1371/journal.pone.0281616

[17] Geng, X., Han, B., Yang, D., & Zhao, J. (2024). Credit risk contagion of supply chain finance: An empirical analysis of supply chain listed companies. PLoS One, 19(8), e0306724. https://doi.org/10.1371/journal.pone.0306724

[18] Yao, Q., Mao, C., & Guo, Y. (2024). A fuzzy neural network-based intelligent warning method for financial risk of enterprises. Journal of Circuits, Systems and Computers, 33(14), 2450251. https://doi.org/10.1142/S0218126624502517

[19] Mu, P., Chen, T., Pan, K., & Liu, M. (2021). A network evolution model of credit risk contagion between banks and enterprises based on agent-based model. Journal of Mathematics, 2021(1), 6593218. https://doi.org/10.1155/2021/6593218

[20] Ma, J., Liu, Y., Zhao, L., & Liang, W. (2024). Research on the mechanism and application of spatial credit risk contagion based on complex network model. Managerial and Decision Economics, 45(2), 1180–1193. https://doi.org/10.1002/mde.4025

[21] Wang, L., Jiang, X., Chen, T., & Zhu, R. (2024). The contagion of debt default risk in energy enterprises considering carbon price fluctuations. Mathematics, 12(17), 2776. https://doi.org/10.3390/math12172776

[22] Li, M., & Fu, Y. (2022). Prediction of supply chain financial credit risk based on PCA-GA-SVM model. Sustainability, 14(24), 16376. https://doi.org/10.3390/su142416376

[23] Chen, X. (2025). Research on financial market volatility prediction and risk response strategy based on LSTM network. J. Combin. Math. Combin. Comput, 127, 5197–5213. https://doi.org/10.61091/jcmcc127a-293

[24] Kadkhoda, S. T., & Amiri, B. (2024). A hybrid network analysis and machine learning model for enhanced financial distress prediction. IEEE Access, 12, 52759–52777. https://doi.org/10.1109/ACCESS.2024.3387462

[25] Ionescu, Ș., Delcea, C., & Nica, I. (2025). Improving real-time economic decisions through edge computing: Implications for financial contagion risk management. Computers, 14(5), 196. https://doi.org/10.3390/computers14050196

[26] Gu, J. (2022). Risk prediction of enterprise credit financing using machine learning. Informatica, 46(7).https://doi.org/10.31449/inf.v46i7.4247.

[27] Zhu, Y., Hu, Y., Liu, Q., Liu, H., Ma, C., & Yin, J. (2023). A Hybrid Approach for Predicting Corporate Financial Risk: Integrating SMOTE-ENN and NGBoost. IEEE Access, 11, 111106-111125.10.1109/ACCESS.2023.3323198.

Appendix 1

RNN	Recurrent neural network	DL	Deep learning
PCA-GA-SVM	Principal component analysis + genetic algorithm + support vector machine	DCC-GARCH	Dynamic conditional correlation multiple generalized autoregressive conditional heteroskedasticity
SMO	Starling Murmuration Optimizer	CA	Categorical attribute
XGBoost	Extreme Gradient Boosting	FR	Feature representation
FPT	Fuzzy preference theory	ARDL	Autoregressive distributed lag
AI	Artificial intelligence	TP	True positive

QRD	Quantum random dive	FNN	Fuzzy neural network
FN	False negative	LSTM	Long Short-Term Memory
SVM	Support Vector Machine	TCRC	Trade credit risk contagion
GCL	Graph convolution layer	OT	Observation time
TENET	Tail-event driven network risk	RF	Random Forest
ML	Machine learning	SD	Standard deviation
ABM	Agent-based model	ISAGNN	Improved Self-Attention Graph Neural Network
FP	False positive	TN	True negative
EC	Edge computing	SC	Supply chain
QHC	Quantum harmonic oscillator	GNN	Graph neural network
ARMA	Autoregressive moving average model	RL	Reinforcement learning
ReLU	Rectified linear unit	DDR	Debt default risk
GBT	Gradient Boosting Tree	LSSVM	Leastsquare support vector machine
SMOTE-ENN	Synthetic Minority Over-sampling and Edited Nearest Neighbors	LightGBM	Light Gradient Boosting Machine
KNN	K-Nearest Neighbors	NGBoost	Natural Gradient Boosting

A Chaotic Scrambling and Diffusion-Based Encryption Framework for Real-Time News Video Transmission

Na Liu
College of Communication, Xijing University, Xi'an 710123, China
E-mail: naliu0427@sina.com

Keywords: chaotic map, image encryption, crowd simulation algorithm, chaotic diffusion mechanism, news dissemination safety

Received: May 30, 2025

In the information age, news videos face growing security risks during transmission. To address this, a novel image encryption algorithm combining chaotic maps and crowd simulation is proposed. The framework integrates a dual mechanism of scrambling and diffusion to disrupt both spatial structure and pixel values. Specifically, a chaotic logistic map is employed to generate dynamic key sequences for pixel-level diffusion, while a crowd simulation algorithm produces pseudo-random sequences to control row and column scrambling. This hybrid structure enhances encryption strength and unpredictability. The experimental results show that the information entropy of the proposed model is 7.97, the pixel change rate is 99.62%, and the pixel correlation is reduced to 0.08. Decryption yields an SSIM of 0.95, while encryption and decryption take only 109 ms and 184 ms, respectively-over 50% faster than comparable models. The model exhibits high security, efficiency, and reversibility, making it well-suited for protecting sensitive news video transmissions.

Povzetek: Članek predstavi hibridni algoritem šifriranja slik za videe novic, ki združi kaotične preslikave za difuzijo s simulacijo množic za vrstično/stolpično mešanje, da okrepi varnost, nepredvidljivost in povratnost prenosa.

1 Introduction

In the era of pervasive digital communication, the transmission of multimedia content-particularly news video-has become a cornerstone of global information dissemination [1]. With the growing reliance on online platforms for news delivery, ensuring the confidentiality, integrity, and authenticity of multimedia data has become a pressing concern. Traditional cryptographic algorithms, although effective for textual or low-dimensional data, often fall short in securing high-volume, real-time video streams due to computational complexity, lack of perceptual sensitivity, or limited robustness against advanced attacks [2]. Therefore, lightweight yet secure encryption schemes are increasingly demanded, particularly in sensitive domains such as political or crisis-driven news coverage, where unauthorized access or manipulation could have severe social consequences. Chaotic systems, known for their deterministic yet unpredictable behavior, offer a compelling alternative for image and video encryption. Characterized by sensitivity to initial conditions, high entropy, and pseudo-randomness, Chaotic Maps (CMs) can efficiently generate complex key sequences suitable for scrambling and diffusion operations. These properties make them inherently resistant to brute-force attacks and statistical analyses. Despite their promise, most existing chaotic encryption schemes remain focused on static images and fail to address the dynamic, frame-based nature of real-world video streams. Furthermore, they often rely on low-dimensional maps or fixed transformation structures, which limits their scalability and resistance to structured attacks. To address these limitations, this study proposes a dual-layer chaotic encryption framework aimed at enhancing security and adaptability in news-oriented video content transmission. Although the experimental validation is conducted on individual video frames, the method is designed to be extensible to real-time video encryption pipelines. The approach integrates crowd simulation-inspired pseudo-random sequence generation with chaotic diffusion and permutation mechanisms, thereby enhancing the unpredictability and robustness of the encryption process. The proposed system is benchmarked against key performance metrics such as information entropy, NPCR, UACI, and processing latency. This paper aims to contribute to the field by bridging the gap between chaotic theory and practical video encryption application, particularly within the context of news broadcasting security. The remainder of this work is organized as follows: Section 2 describes the proposed methodology in detail, Section 3 presents the experimental validation and performance evaluation, and Section 4 discusses the security analysis and potential extensions. Finally, conclusions are drawn in Section 5, along with directions for future work.

2 Related works

In recent years, with the increasingly severe issue of information security, image and video encryption technology has gradually become a research focus in the field of multimedia security. Yao et al. proposed a color image compression and encryption algorithm that combines compressive sensing, Sudoku matrix, and hyper chaotic system to ensure the security of color image data and improve transmission and storage efficiency. This algorithm designed a new hyper chaotic system, improved the beetle optimization algorithm to optimize the compression threshold, and introduced Sudoku matrix and bidirectional diffusion operation. The research results indicated that the algorithm had high security [3]. Singh et al. developed a new multi-layer Image Encryption (IE) scheme and improved grayscale and color image authentication techniques to overcome the security crisis of private data in network communication. The scheme combines a public key cryptosystem and two chaotic systems. The research results indicated that the algorithm had characteristics such as a huge key space and extremely low correlation coefficient, and was efficient in resisting statistical attacks. It could effectively resist different brute force attacks [4]. Wang et al. proposed a multi-IE method based on computer-generated phase only hologram algorithm and chaotic system for secure encryption of multiple images. This method applied an improved Gerchberg-Saxton algorithm to generate subsampled phase only holograms, and used spatial segmentation multiplexing to combine multiple phase holograms, which are then transformed into ciphertext through a chaotic system. The research results indicated that this method eliminated the problem of information leakage, increased the complexity of the encryption system, and verified its security and feasibility [5].

Liu et al. proposed a novel 3D medical image encoding scheme based on biometric keys and cube boxes to overcome the shortcomings of research on 3D medical IE. This scheme utilized biometric keys to enhance data security and constructed cube boxes to increase nonlinearity. The research results indicated that this encryption scheme had good statistical performance, large key space, high sensitivity, and robustness, and could resist various typical cryptographic attacks [6]. Huang et al. proposed a color image block encryption algorithm based on cellular neural networks and Chua's chaotic system to overcome the problems of small key space and susceptibility to plaintext attacks in low dimensional chaotic encryption. This algorithm generated chaotic sequences through Fourier transform, combined them with solid colors for diffusion, and used an improved Chua's chaotic system to scramble the image. The research results indicated that the algorithm had good encryption performance and could resist common attacks [7]. Zhang et al. developed a new simultaneous obfuscation diffusion IE algorithm to address the efficiency and security issues of existing IE algorithms in Deoxyribonucleic Acid (DNA) encoding and chaotic scrambling diffusion operations. This algorithm adopted a composite coupled chaotic system, combined with DNA encoding for synchronous perturbation diffusion encryption. The research findings indicated that the proposed method had higher security and better performance than representative methods [8].

Peng et al. designed an IE system based on a chaotic hardware encryption framework, utilizing a multi-scroll chaotic system and the Arnold transformation as the primary sources of entropy. The image was processed through a chaotic sequence, with the Arnold transformation applied for scrambling. Laboratory results demonstrated that the system exhibited low power consumption, high processing speed, and strong encryption performance [9]. Boussif et al. proposed a novel IE method to secure medical digital images used in communication systems. The method first converted the image into a pixel matrix, encrypted the image blocks individually, and updated the encryption key using the Arnold transformation. Experimental results showed that this approach successfully secured the images and achieved lower computational time compared to conventional encryption algorithms [10]. Wang et al. developed a Chaotic Image Encryption (CIE) algorithm based on a matrix semi-tensor product and a composite key. The method divided the image into four segments, applied the Arnold transformation to each segment, and subsequently combined them to produce the encrypted image. The study demonstrated that the algorithm offered higher security than traditional methods and was well-suited for encrypting color images [11]. Jain et al. tackled the challenge of securely transmitting digital images in remote healthcare systems, which often contain sensitive patient data. The researchers proposed a CIE technique combining the Arnold Cat map with a 2D Logistic Sine Coupling Map. Their results indicated that the scheme enhanced both the randomness and security of encrypted images, thereby ensuring adequate protection of patient information [12]. Zarebnia Mde et al. introduced a multi-layer IE method incorporating a chaotic system, where images were scrambled using the Arnold transformation. The findings showed that the method achieved robust encryption performance and could withstand various types of attacks, effectively safeguarding user image data [13].

Table 1: Related works

Research	Method	Research content	Key performance metrics	Reference
Yao et al. (2025)	Compressive sensing + hyperchaos	Color image compression and encryption using Sudoku matrix and bidirectional diffusion	High security, improved transmission efficiency; entropy ≈ 7.83, NPCR > 98.9%	[3]
Singh et al. (2025)	RSA + CMs	Grayscale and color IE using hybrid cryptosystem	Low pixel correlation, strong against brute force; SSIM ≈ 0.79	[4]
Wang et al. (2023)	Phase-only holograms + chaos	Multiple-IE via spatial multiplexing and chaotic phase encoding	Eliminates leakage; entropy ≈ 7.90, good multi-image diffusion security	[5]

Liu et al. (2024)	Biometric key + cube structure	3D medical IE with cubic boxes and biometric-derived keys	Strong robustness and key sensitivity; entropy ≈ 7.87, SSIM ≈ 0.80	[6]
Huang et al. (2024)	CNN + Chua's chaotic system	Color IE using Chua chaos and pixel block scrambling	Improved resistance to plaintext attack; NPCR ≈ 99.18, limited scalability	[7]
Zhang et al. (2024)	DNA encoding + composite chaos	Chaotic scrambling and DNA diffusion for secure encryption	Higher entropy (≈ 7.92), better pixel decorrelation than previous DNA methods	[8]
Peng et al. (2023)	Hardware-based chaotic IE framework + Arnold transformation	Used a multi-scroll chaotic system and Arnold map for scrambling; targeted low-power, high-speed IE	High processing speed, low power, strong encryption	[9]
Boussif et al. (2022)	Block-wise encryption with dynamic key update	Converted image to pixel matrix, encrypted blocks individually, updated keys with Arnold transformation	Lower computational time, secure transmission	[10]
Wang et al. (2023)	Matrix semi-tensor product + composite key	Segmented image into four parts, applied Arnold scrambling per segment, merged afterward	High security, suitable for color IE	[11]
Jain et al. (2022)	Arnold Cat map + 2D Logistic Sine Coupling Map	Developed a chaotic scheme for secure medical image transfer in remote healthcare	Improved randomness and security, protection of patient data	[12]
Zarebnia Mde et al. (2021)	Multi-layer chaotic IE	Applied layered scrambling using chaotic system and Arnold transformation	Robust encryption, resistant to various attacks	[13]

In summary, existing studies have made substantial progress in enhancing encryption strength and resisting statistical or differential attacks. However, most approaches rely on conventional chaotic systems with limited sequence complexity or pre-defined scrambling patterns. Few, if any, incorporate dynamic behavioral simulation mechanisms such as crowd simulation to generate adaptive control sequences. Moreover, the joint optimization of spatial scrambling and numerical diffusion remains insufficiently addressed. This study fills the gap by integrating an improved crowd simulation algorithm with logistic chaotic diffusion in a unified framework. This novel combination enhances both the unpredictability of the permutation order and the nonlinear diffusion effect, thereby significantly improving encryption entropy, pixel decorrelation, and processing efficiency, particularly suited for real-time news video encryption scenarios.

3 Methods and materials

3.1 Encryption algorithm for news communication video based on chaos mapping

To address the dual challenge of security and efficiency in real-time news video transmission, this study designs an encryption framework based on chaotic dynamics and enhanced randomization control. The selection of the logistic CM is motivated by its simplicity, strong sensitivity to initial conditions, and ease of hardware implementation, making it suitable for high-speed encryption tasks in video scenarios. Unlike complex hyperchaotic models, the logistic map offers a trade-off between computational efficiency and sufficient nonlinear behavior needed for effective diffusion operations. Moreover, to overcome the limitations of traditional pseudorandom number generators, the framework incorporates a crowd simulation mechanism. This mechanism simulates non-deterministic behavioral dynamics in group movement, introducing dynamic perturbation patterns for scrambling operations. By using individual "participants" with evolving positional logic, the system generates permutation vectors that are both data-dependent and highly irregular, thereby improving the unpredictability of the scrambling phase without relying on externally seeded randomness.

In modern society, the immediacy and breadth of news dissemination have proposed higher requirements for the security of video data. Especially in sensitive fields such as politics, military, and disaster reporting, the leakage, tampering, and even forgery of video content will seriously affect public awareness and social stability. Traditional video encryption methods struggle to balance encryption efficiency and security when faced with large amounts of data and real-time transmission requirements [14-15]. Therefore, a CM-based encryption algorithm for news dissemination videos has been proposed in the study. The CM principle is shown in Figure 1.

Figure 1 presents the bifurcation diagram of the logistic map, illustrating how its output behavior changes with respect to variations in the control parameter μ. The horizontal axis represents the parameter μ, while the vertical axis represents the resulting state values after convergence. As μ increases beyond approximately 3.57, the system transitions from periodic to fully chaotic behavior, exhibiting high sensitivity and unpredictability-ideal characteristics for cryptographic key stream generation. These dense distributions of output points at higher μ values indicate that the sequence lacks repeatable patterns, which is desirable for permutation operations in encryption. In this study, a value of $\mu = 3.99$ is chosen to ensure operation in the chaotic regime. The resulting sequence is then used as a dynamic controller for row and column shuffling of image pixels. This mechanism enhances spatial structure disruption in the plaintext image, providing a strong first layer of encryption before the diffusion phase [16-17]. Therefore, the study adopts chaotic system for row shuffling, and the structure framework of row shuffling is shown in Figure 2.

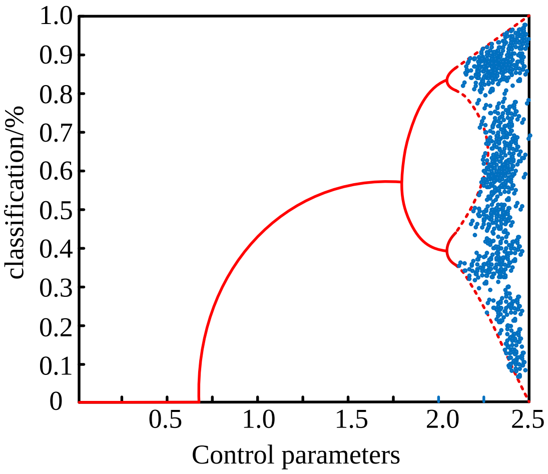


Figure 1: Bifurcation diagram of chaotic system.

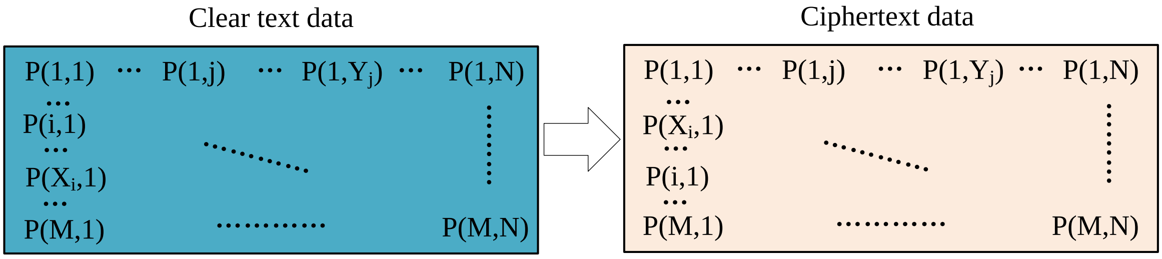

Clear text data
P(1,1) ··· P(1,j) ··· P(1,Y_j) ··· P(1,N)
P(i,1)
P(X_i,1)
P(M,1)
P(M,N)
Ciphertext data
P(1,1) ··· P(1,j) ··· P(1,Y_j) ··· P(1,N)
P(X_i,1)
P(i,1)
P(M,1)
P(M,N)

Figure 2: Chaotic row structure framework.

As shown in Figure 2, the row-level scrambling mechanism operates by swapping the positions of two specific rows within the image matrix. Each row, such as P(i,1), represents a horizontal line of pixel values. In this example, the i-th row and the X_I-th row exchange their positions, as controlled by the permutation sequence derived from the chaotic system. While Figure 2 depicts a single swap for illustration, the full encryption process involves iteratively applying such row-level exchanges across the entire image using a pseudo-random index vector. This approach disrupts the spatial continuity of the original structure, ensuring that even if an attacker observes the encrypted image, recovering the original ordering of rows becomes computationally infeasible without the correct key sequence [18-19]. This operation breaks the spatial continuity of the image, making it hard for attackers to restore the original information by analyzing the image structure even if they obtain encrypted images. The image data shown on the right after row shuffling appears to have an overall disordered structure, but in reality, it is the result of precise control based on a chaotic sequence or pseudo-random rule. This figure illustrates the row-level scrambling process applied to the plaintext image matrix. The original image rows are permuted according to a control sequence derived from chaotic dynamics, disrupting the vertical spatial structure. This stage serves as the first line of spatial obfuscation, enhancing resistance against structural and statistical attacks [20-21]. To further enhance the unpredictability and security of row permutation in the encryption process, an improved crowd simulation algorithm was introduced as a chaotic sequence generation mechanism based on the traditional chaotic encryption framework to improve the complexity and randomness of sequence perturbations. Its structure is shown in Figure 3.

As shown in Figure 3, a pseudo-random position sequence is generated by simulating the out of column order of the crowd under specific rules to control the permutation order of pixel rows or columns in the image. The use of a crowd simulation algorithm contributes to security by providing a rule-driven, non-deterministic mechanism for generating pixel permutation sequences. Each step in the simulation incorporates prior positional states and dynamic interval calculations, ensuring that even small changes in input lead to entirely different scrambling results. Unlike fixed pseudorandom number generators, this method avoids cyclicity and increases the effective key space. Furthermore, because the algorithm relies on lightweight logical operations rather than cryptographic hashing or matrix inversion, it introduces minimal additional runtime overhead. The process first inputs the starting point position S_0 , the total number of people N , and the basic distance k . Each participant holds a set of keys k_i and is uniformly numbered P_N . Starting from the S_0 th person, the algorithm initializes the reporting interval to k and jumps the number of people clockwise from the current position [22]. The calculation expression for the position L_i of the i th round of the column is shown in equation (1).

$$L_i = (S_{i-1} + k_i)\text{mod}N' \qquad (1)$$

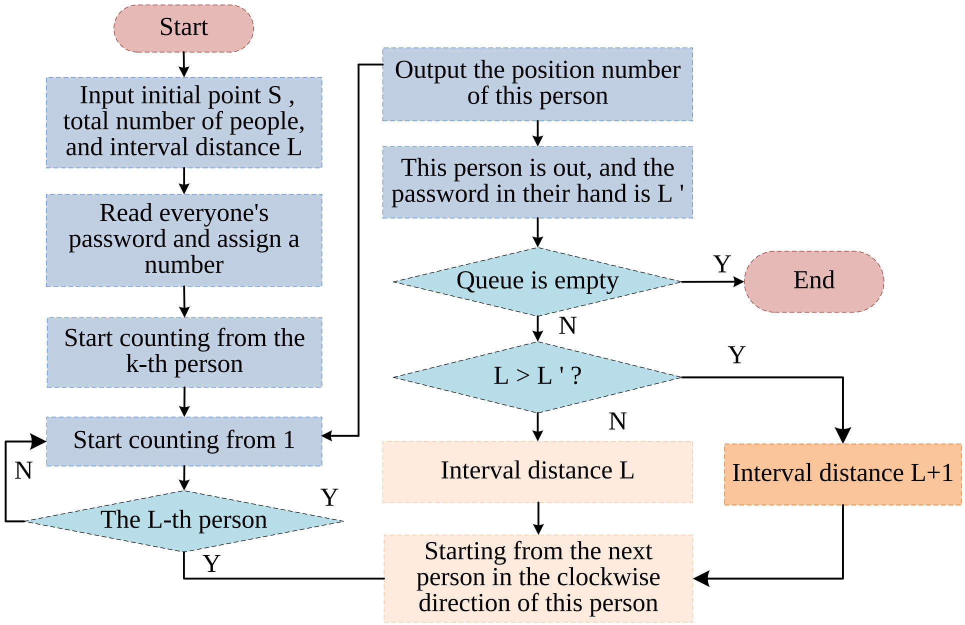

Start
Input initial point S , total number of people, and interval distance L
Read everyone's password and assign a number
Start counting from the k-th person
Start counting from 1
The L-th person
N
Y
Y
Output the position number of this person
This person is out, and the password in their hand is L '
Queue is empty
Y
End
N
L > L ' ?
Y
N
Interval distance L
Interval distance L+1
Starting from the next person in the clockwise direction of this person

Figure 3: A permutation control sequence generation model based on crowd simulation algorithm.

In equation (1), S_{i-1} represents the position of the last delisted person, k_i is the key value held by that person, and N' means the remaining number of people in the current queue. Whenever someone goes out of the queue, the key in their hand will be used as a perturbation factor to participate in the calculation of the next interval, as shown in equation (2).

$$k_{i+1} = (k_i \cdot \alpha + \beta) \bmod \gamma \qquad (2)$$

In equation (2), α , β , and γ are the nonlinear extension parameters set by the system to increase the complexity of the disturbance interval sequence. Next, the position number L_i of each outlier is recorded and filled in the chaotic disruption control vector Z in turn, the expression of which is shown in Equation (3).

$$Z_i = L_i \qquad (3)$$

To prevent regularity between consecutive column positions, disturbance rules are further introduced for secondary position mapping, as expressed in equation (4).

$$Z_i^{'} = (Z_i + \delta_i) \bmod N \qquad (4)$$

In equation (4), δ_i is the disturbance offset calculated based on the difference in position from the previous round. To enhance the unpredictability of spatial permutations and prevent regular patterns, the position sequence Z obtained from the crowd simulation is further refined through secondary mapping to generate Z' . This refined sequence maintains high entropy and removes potential linearity between consecutive elements. While equation (4) describes this process as a generic positional disturbance, in the proposed encryption pipeline, Z' is specifically used as the row permutation control sequence, guiding the scrambling of image rows during the first stage of encryption. In an extended implementation, a parallel sequence can be similarly generated for column-wise permutation if two-dimensional scrambling is required. Its calculation expression is shown in equation (5).

$$\delta_i = | Z_i - Z_{i-1} | \qquad (5)$$

The final sequence $Z' = \{Z_1^{'}, Z_2^{'}, ..., Z_N^{'}\}$ is obtained, which will serve as the row swapping control sequence in the row shuffling method, guiding the reconstruction process of the arrangement of each row in the plaintext image. Due to the nonlinearity and key dependence of the extraction process, this sequence is highly complex and unpredictable, significantly enhancing the confidentiality of row column scrambling in IE. Combined with chaotic system mechanism, the encryption system has stronger resistance to statistical attacks and known plaintext attacks, thus meeting the dual requirements of encryption strength and speed in news dissemination videos.

In this study, the encryption framework is built upon the logistic CM due to its well-established nonlinear behavior, sensitivity to initial conditions, and low computational complexity. The control parameter used is close to the boundary value of full chaotic behavior, and the initial seed value is selected from a high-precision random domain to ensure that even the slightest variation leads to a completely different output sequence. This high sensitivity ensures that the encrypted results are highly dependent on the initial key, thereby enhancing security. To strengthen the unpredictability of spatial scrambling, a crowd simulation mechanism is introduced in place of conventional pseudorandom generators. Unlike static or cyclic random number generators, the crowd simulation mimics the dynamic interactions of individuals in a group. Each participant in the simulated environment carries a private key and follows rule-driven behaviors, such as changing positions based on interaction history or decision logic. This produces a highly variable and non-repetitive control sequence for pixel permutation. The system parameters within the simulation are specifically selected to increase complexity and eliminate deterministic patterns while ensuring that the system remains lightweight and computationally efficient. Additionally, the algorithm design emphasizes modularity and scalability. By separating the scrambling and diffusion stages and applying them independently to each color channel, the method supports parallel processing and adapts easily to different video resolutions and frame sizes.

3.2 News video encryption algorithm based on chaotic diffusion and CIE

The study introduced chaotic system and improved crowd simulation algorithm to achieve scrambling of video frame images in the row dimension, disrupting the spatial structure of the original image and effectively improving the security of IE. However, a single scrambling operation is not sufficient to combat complex security threats such as differential attacks and statistical attacks [23]. Therefore, further research is needed to construct a joint encryption framework based on "scrambling+diffusion", which performs diffusion operations on pixel values on the basis of spatial structure disturbance, enhancing the information confusion of images in the numerical dimension. Its structure is shown in Figure 4.

As denoted in Figure 4, the proposed encryption framework adopts a layered encryption strategy composed of two distinct modules: the Chaotic Diffusion Encryption (CDE) module and the CIE module. The CDE module operates at a pixel and bit-plane level. It separates the input image into R, G, and B channels, further splits each channel into high and low bit planes, and applies chaotic diffusion using logistic sequences, thereby introducing strong local perturbations and ciphertext feedback. After pixel-level scrambling and value-wise diffusion, the resulting intermediate encrypted image is passed into the CIE module. The CIE module performs a second stage of encryption, focusing on global structure reinforcement. It incorporates secret-key-controlled matrix transformations and a dual diffusion process that spans the entire image matrix. This stage enhances statistical masking and ensures that any structural remnants or value correlations from the first stage are fully obscured. The two modules operate in a hierarchical manner: CDE provides high-entropy, fine-grained diffusion, while CIE reinforces key-dependency and structure-wide decorrelation, forming a comprehensive encryption pipeline. The encryption process starts from the input plaintext image and first generates a pseudo-random key sequence through a chaotic system using a key seed. This sequence serves as a control vector to guide the subsequent encryption operations of the image. Subsequently, the plaintext image enters the CDE module. At this stage, the row and column order of the image is perturbed and replaced according to the key sequence, disrupting the spatial structure. Subsequently, the image is fed into the CIE, which uses the grayscale value of the previous pixel or contextual pixel to perform XOR diffusion with the chaotic sequence, achieving numerical encryption of the image content and ultimately outputting a ciphertext image [24]. The encrypted image is transmitted through a public channel, while the key seed is transmitted through a secure channel to ensure the overall security of the system. During the decryption process, the key seed is re input into the chaotic system to generate the same key sequence as the encryption end. Combined with the input ciphertext image, the plaintext association decryption algorithm is first executed according to the encryption inverse process to restore pixel grayscale, and then the scrambling diffusion inverse operation is performed to restore the image structure, ultimately restoring the original plaintext image. In this process, the most important ones are the scrambling diffusion encryption algorithm and the CIE algorithm, among which the structure of the scrambling diffusion encryption algorithm is shown in Figure 5.

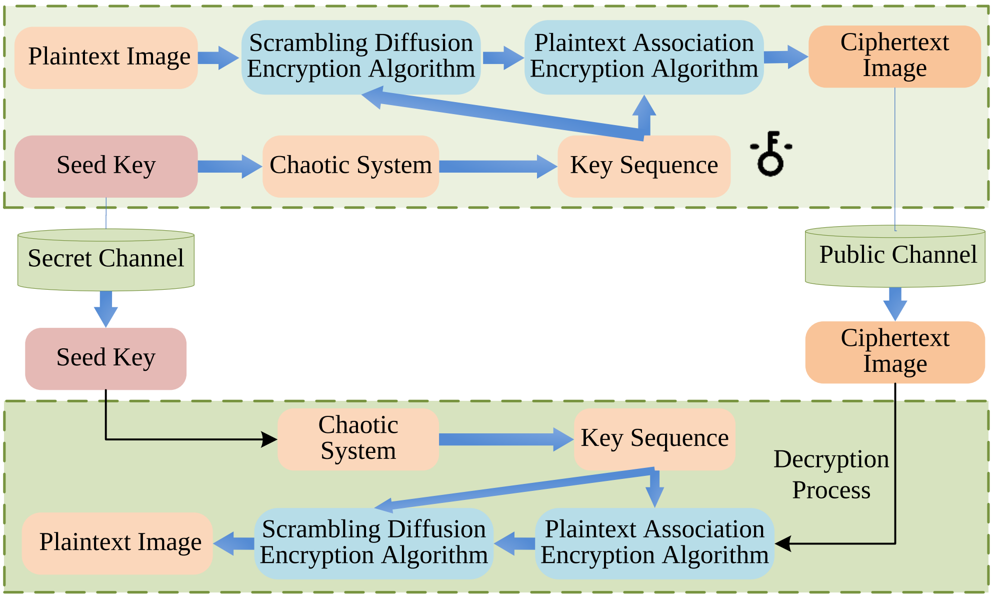


Figure 4: Structure of news video encryption algorithm based on chaotic diffusion and CIE.

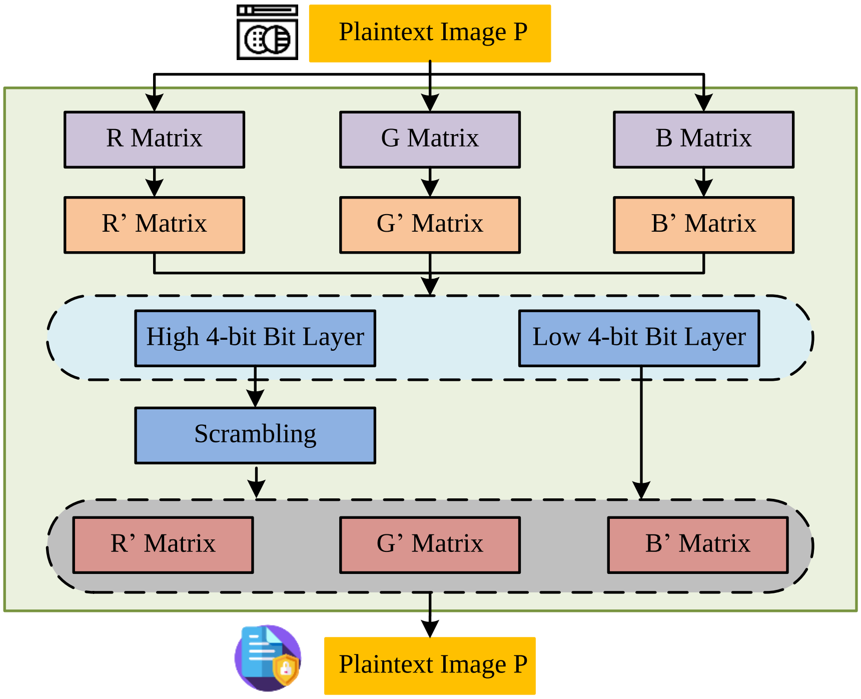


Figure 5: Structure of CDE algorithm.

As shown in Figure 5, the CDE module begins with a lightweight spatial scrambling step applied independently to each RGB channel. This involves permuting the rows and columns based on a pseudo-random permutation vector. After this, each channel is split into high and low bit planes, and chaotic diffusion is applied to each bit layer using key-dependent sequences. The combined result introduces both structural and value-level encryption effects. Encryption processes are designed for the three channels of color images, dividing the original plaintext image into three parts: R, G, and B. Each channel is processed separately to enhance fine-grained control of IE. Each channel first enters the expansion module and completes preliminary scrambling under chaotic sequence control, disrupting the spatial distribution structure of pixels. Then, each pixel is divided into high 4 bits and low 4 bits by bit and subjected to nonlinear diffusion processing [25]. The diffusion algorithm is based on the chaotic sequence generated by logistic mapping, and its formula expression is shown in equation (6).

$$C_i = (P_i \oplus K_i) \oplus C_{i-1} \quad (6)$$

In equation (6), C_i is the encrypted value of the i th pixel, P_i is the original pixel value, K_i is the key value in the chaotic sequence, C_{i-1} is the previous encrypted pixel value, used to introduce a ciphertext feedback mechanism to enhance diffusion strength. The diffusion algorithm is based on the chaotic sequence generated by logistic mapping, and its basic form is shown in equation (7).

$$x_{n+1} = \mu x_n (1 - x_n) \quad (7)$$

In equation (7), x_n denotes the chaotic value of the nth iteration, and μ denotes the system control parameter. The logistic map is tuned with a control parameter μ set to 3.99, which lies in the upper end of the chaotic regime, ensuring full chaos and eliminating periodic behavior. The initial seed value for the sequence is selected randomly within the interval (0,1), with precision extended to fourteen decimal places to maximize sensitivity. To verify the robustness of the generated chaotic sequences, multiple validation procedures are performed. These include analysis of the sequence's Lyapunov exponent, which is found to be positive under all operating parameters, indicating sustained chaotic divergence. In addition, standard randomness evaluation tests such as NIST SP 800-22 are applied to the generated key streams, confirming uniform distribution, low autocorrelation, and absence of detectable patterns.

After the above diffusion, the high 4 bits and low 4 bits are reordered in the Jospehus algorithm module, which performs a circular column out operation on the sequence through specific hop counts and intervals to further enhance the unpredictability of the sequence. After the high and low bits are recombined, they are merged to generate encrypted R, G, and B matrices, and finally merged to output an encrypted image. Then, the encrypted image is input into the CIE algorithm for further encryption, as shown in Figure 6.

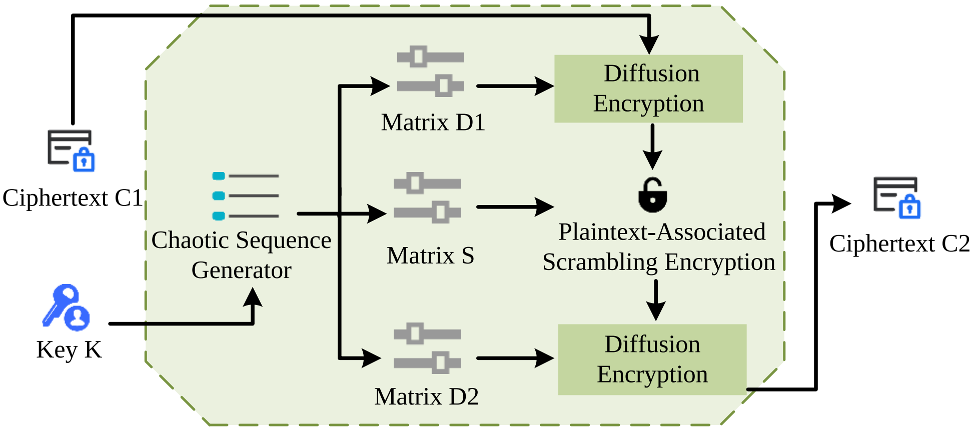


Figure 6: Global key-driven CIE module.

```
Dual-Layer Chaotic Image Encryption Framework

Input: Plain image I (RGB), initial key seed x₀, control parameters μ, α, β, γ
Output: Encrypted image C

1. [Initialization]
   Generate initial chaotic sequence X using logistic map:
     $x_0 \in (0,1)$, $\mu \in (3.9, 4)$
   Simulate crowd behavior:
     Set number of participants N, base interval $k_0$
     Initialize participant keys $k_i$, and compute permutation index vector Z

2. [Scrambling Stage]
   For each color channel (R, G, B):
     a. Permute rows and columns of I using vector Z
     b. Apply bit-plane split into high-4-bit and low-4-bit matrices

3. [Diffusion Stage]
   For each bit-plane:
     a. Generate chaotic key stream from X
     b. Perform value-wise XOR between pixel and key stream
     c. Introduce pixel-wise dependency by chaining previous ciphertext values

4. [Reconstruction]
   a. Recombine bit-planes to form scrambled-diffused R, G, B matrices
   b. Merge color channels into final ciphertext image C

Return: C
```

Figure 7: Dual-layer CIE framework.

As shown in Figure 6, the module first performs Plaintext-Associated Scrambling Encryption, which dynamically generates permutation sequences based on plaintext characteristics to destroy any inherent structural correlation in the image, followed by Diffusion Encryption to enhance key sensitivity and achieve strong statistical confusion. The algorithm takes the key K and the initial ciphertext image C1 as inputs, and generates a highly random key stream matrix S through a chaotic sequence generator to control the core parameters of the entire image diffusion and scrambling process. Firstly, the key K is introduced into the chaotic system, and the one-dimensional sequence is transformed into a two-dimensional matrix S through chaotic system. Two control matrices D_1 and D_2 are further generated for diffusion encryption, which are applied to different stages of IE. The ciphertext C1 undergoes the first diffusion encryption and XOR operation with the elements of matrix D1, and its expression is shown in equation (8).

$$C_i' = C_i \oplus D_{1,i} \tag{8}$$

In equation (8), C_i is the i th pixel value, and $D_{1,i}$ denotes the corresponding element in the control matrix. This operation can disrupt the statistical distribution of the original image pixel values. Afterwards, the structure in the plaintext correlation scrambling diffusion module is jointly operated with the key matrix S, introducing the internal structural correlation of the image. Then, a second diffusion operation is performed on the result, and the final pixel perturbation is completed through matrix D2. The core calculation expression is shown in equation (9).

$$C_i'' = C_i' \oplus D_{2,i} \tag{9}$$

According to equation (9), the final output ciphertext image has a certain level of encryption strength and anti-analysis performance.

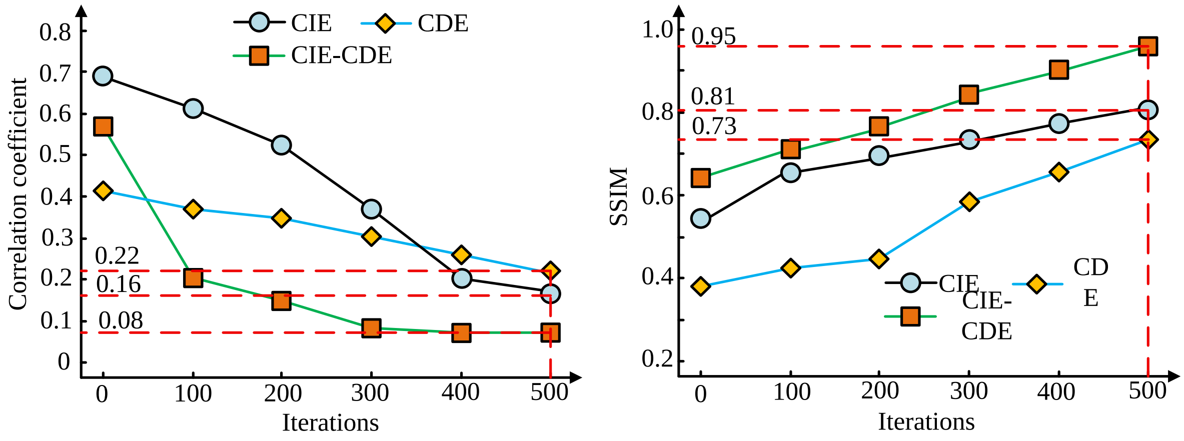
4 Results

4.1 News video encryption algorithm based on chaotic diffusion and CIE

The experimental hardware configuration used in the study was Intel Core i7-13900 as the central processor, NVIDIA Geforce GTX4060Ti as the graphics processor, 16GB of VRAM, 32GB of RAM, and Windows 11 operating system. The dataset adopted the publicly available TVSum Dataset, which contains 50 real video clips from YouTube, covering multiple topic categories such as news, travel, speeches, DIY, sports, lifestyle, music, etc. The study selected news related data and divided it into 5000 pieces, which were then divided into a training set and a validation set in a 4:1 ratio. To ensure that the proposed video encryption framework is evaluated on a standardized and widely recognized benchmark, the TVSum dataset was integrated into the experimental pipeline. TVSum contains 50 real-world videos sourced from YouTube across diverse genres-including news, sports, documentaries, and user-generated content-with frame-level human-annotated importance scores. Although the original dataset is primarily intended for video summarization, its granularity and annotation structure enable fine-grained evaluation of visual consistency and decryption fidelity across keyframes and transition scenes. In this study, individual video frames were extracted from four representative categories within TVSum to assess the encryption model's adaptability across content types. This integration facilitates consistent benchmarking, improves experimental rigor, and aligns the evaluation protocol with community standards.

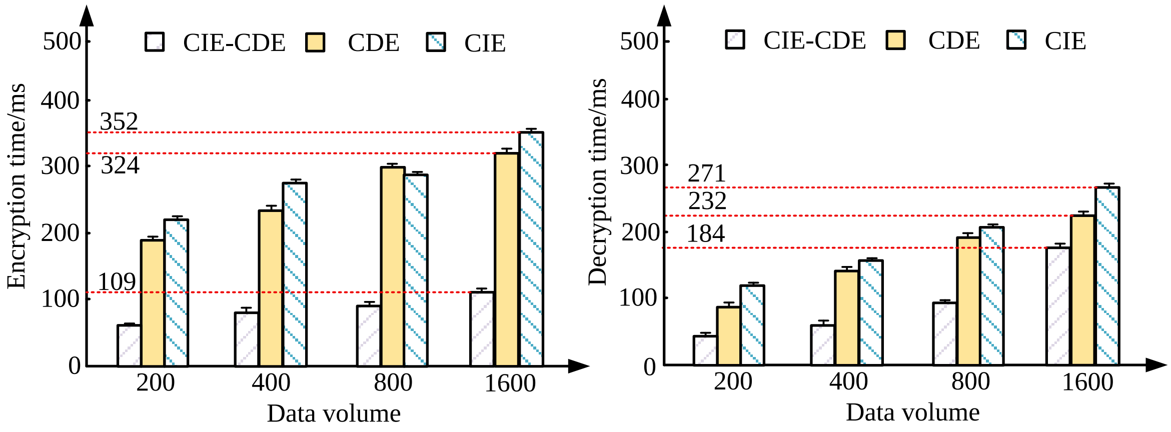
Selecting a single CIE and CDE as comparison models, the results are shown in Figure 8.

Figure 8 (a) shows the trend of correlation coefficient changes for different algorithms during multiple iterations, while Figure 8 (b) illustrates the variation of Structural Similarity Index Measure (SSIM) during the iteration process. From Figure 8 (a), CIE, CDE, and CIE-CDE all had high correlation in the early stages of iteration, especially the initial correlation coefficient of the CIE algorithm was close to 0.7, indicating that the image has not completely broken the statistical dependence between the original pixels. As the amount of iterations increased, the correlation coefficients of the three methods gradually decreased, with the CIE-CDE combination algorithm showing the most significant decrease. At 500 iterations, its correlation was the lowest at only 0.08, significantly better than CIE's 0.22 and CDE's 0.16. This indicates that the CIE-CDE method, which combines scrambling and diffusion mechanisms, can more effectively eliminate redundant information in images and enhance the ability to resist statistical attacks in IE. Figure 8(b) presents the SSIM values between the decrypted images and the original plaintext images under different iteration counts. As the number of iterations increased, the SSIM value steadily rose, reaching 0.95 at 500 iterations. This indicated that the decrypted images closely approximated the original images in structure and visual fidelity, reflecting strong reversibility and decryption accuracy of the proposed encryption-decryption pipeline. It is important to note that SSIM evaluates the similarity between the output of the decryption process and the original image, and should not be interpreted as a metric of encryption security. In fact, during encryption, the structural correlation between the ciphertext and the original should be minimized to ensure confidentiality. The analysis of various models under different data volumes is denoted in Figure 9.



(a) Correlation coefficient (b) SSIM

Figure 8: Analysis of correlation coefficient and ssim changes of various models.



(a) Encryption time

(b) Decryption time

Figure 9: Changes in encryption and decryption time of three encryption algorithms under different data volumes.

Table 2: Comprehensive performance analysis table.

Test	Model	Entropy	NPCR/%	UAC/%	Pixel correlation	SSIM	Encryption time/ms	Declassified time/ms
Test 1	CIE	7.85	99.21	32.14	0.22	0.81	352	271
	CDE	7.89	99.37	32.71	0.16	0.73	324	232
	CIE-CDE	7.97	99.62	33.09	0.08	0.95	109	184
Test 2	CIE	7.84	99.17	32.06	0.23	0.8	349	268
	CDE	7.88	99.34	32.65	0.15	0.74	321	229
	CIE-CDE	7.96	99.6	33.02	0.09	0.94	111	186

Figure 9 (a) illustrates the variation in encryption time for three encryption algorithms under different data volumes, while Figure 9 (b) depicts the comparative decryption times under the same conditions. As data volume increased, the encryption time of all three algorithms increased correspondingly. Among them, the CIE algorithm consistently required the longest processing time, reaching 352 milliseconds at a data volume of 1600. The CDE algorithm followed at 324 milliseconds, whereas the CIE-CDE algorithm achieved an encryption time of just 109 milliseconds under identical conditions, demonstrating a substantial advantage in encryption efficiency. This suggests that CIE-CDE minimizes redundant computations through structural optimization and process integration, thereby achieving significantly improved encryption performance compared to traditional methods. According to Figure 9 (b), when the data volume was 1600, the decryption time of the CIE algorithm was 271 milliseconds, CDE was 232 milliseconds, and CIE-CDE achieved the shortest decryption time at just 184 milliseconds. This result indicates that the CIE algorithm still involves complex decryption procedures, potentially including lengthy anti-diffusion steps or computationally intensive inverse transformations. In contrast, CIE-CDE reduces decryption latency by employing a symmetric and streamlined architecture that ensures accuracy while improving efficiency. The experimental results demonstrate that CIE-CDE exhibits strong encryption performance and excels in both encryption and decryption efficiency, making it particularly well-suited for news video encryption scenarios with stringent real-time requirements. To evaluate the robustness and consistency of the proposed encryption model, two independent experiments were conducted using the same video dataset (TVSum) but with randomly initialized secret keys and chaotic seeds. These are referred to as Test 1 and Test 2 in Table 2. Each test applied the encryption pipeline to the same input but with different internal parameters, allowing assessment of the model's performance under random key variations.

According to Table 2, in Test 1, CIE-CDE performed the best in terms of security and efficiency, with an information entropy of 7.97, which was closest to the ideal value of 8, indicating that the encrypted image has a high degree of randomness. At the same time, its NPCR was 99.62% and UACI was 33.09%, far higher than CIE's 99.21% and 32.14%, indicating stronger sensitivity to changes in pixel values and higher resistance to differential attacks. In terms of pixel correlation, CIE-CDE was only 0.08, significantly better than CIE's 0.22 and CDE's 0.16, reflecting its ability to better disrupt the original image structure. In terms of SSIM, CIE-CDE was 0.95, which was close to perfect restoration, indicating excellent encryption reversibility. In addition, its encryption and decryption time were only 109 milliseconds and 184 milliseconds respectively, far lower than other models. The performance of each model in Test 2 was basically consistent with Test 1, which verified the stability of the results and the robustness of the algorithm. CIE-CDE still maintained the highest entropy value of 7.96, the lowest correlation of 0.09, and the shortest time of 111 milliseconds and 186 milliseconds, with the best overall performance. The experimental results indicate that the proposed method is suitable for news IE scenarios that require both high security and efficiency. Table 1 presents the overall encryption performance of the proposed algorithm under two independent tests (Test 1 and Test 2), each using different randomly initialized secret keys but applied to the same set of news videos. These tests are designed to assess algorithmic stability and robustness under key variation. In contrast, Table 2 provides detailed simulation performance across four categorized types of news content-Type A (politics), Type B (economy), Type C (entertainment), and Type D (social affairs)-to evaluate real-world adaptability. While both tables are based on the same encryption framework, Table 1 focuses on key sensitivity, whereas Table 2 evaluates content-dependent operational performance.

4.2 Model simulation performance testing

To further assess the effectiveness of the model, four different types of news data were selected and encrypted. The results are shown in Figure 10.

Figure 10 (a) illustrates the SSIM of three encryption models across different types of news videos, while Figure 10 (b) depicts the cumulative encryption and decryption time of the three algorithms for four types of news videos. In news Type A, the SSIM of CIE was approximately 0.76, CDE achieved 0.79, and CIE-CDE reached 0.87, suggesting that CIE-CDE demonstrates superior decryption accuracy and image reconstruction quality for this video type. In news Type B, the SSIM of CIE-CDE was close to 0.89, significantly higher than CIE’s 0.78 and CDE’s 0.74, highlighting its enhanced ability to preserve image structure. The performance gap further widened in Type C, where CIE and CDE attained SSIM values of only 0.72 and 0.75, respectively, while CIE-CDE exceeded 0.85, demonstrating its robustness in preserving image reversibility for frames with complex textures. In Type D, CIE-CDE also outperformed the other methods, achieving an SSIM above 0.88, compared to 0.70 for CIE and 0.67 for CDE. According to Figure 10 (b), CIE required approximately 460 milliseconds for Type D, while CDE consumed slightly less at around 360 milliseconds. In contrast, CIE-CDE exhibited significantly lower latency compared to both models, maintaining encryption and decryption times below 200 milliseconds across all news types, with Type C requiring only about 120 milliseconds. For Types B and A, its processing times were approximately 150 and 130 milliseconds, respectively, demonstrating notable efficiency gains attributable to its structural optimization. These results indicate that CIE-CDE maintains stable structural reconstruction across diverse news content types, making it particularly suitable for broadcasting scenarios where high-quality post-decryption imagery is essential. The actual encryption effects of each model were analyzed, and the results are shown in Figure 11.

Figure 11 (a) shows the original image, and Figures 11 (b) to 11 (d) represent the encrypted images of CIE algorithm, CDE algorithm, and CIE-CDE algorithm, respectively. From Figure 11, although the overall texture of the image encrypted by the CIE algorithm has been disrupted, a certain degree of structural residue could still be observed in the image, especially in the lower region where there is a slight stripe feeling. This indicates that the CIE algorithm has certain limitations in destroying inter pixel correlation, resulting in the preservation of some original structural information after IE. In CDE encryption, the overall image was more uniform, but there were still slight block patterns, with strong diffusion but insufficient scrambling. The image encrypted by CIE-CDE algorithm had a highly uniform pixel distribution, presenting an ideal random noise state without obvious recognizable structures. The experimental results show that the CIE-CDE algorithm can effectively integrate the advantages of scrambling and diffusion, achieve high-strength encryption and complete structural destruction, and significantly improve the visual unidentifiable and security of encrypted images. Table 3 provides detailed simulation performance across four categorized types of news content-Type A (politics), Type B (economy), Type C (entertainment), and Type D (social affairs)-to evaluate real-world adaptability. The simulation performance of each model was tested, and the results are shown in Table 3.

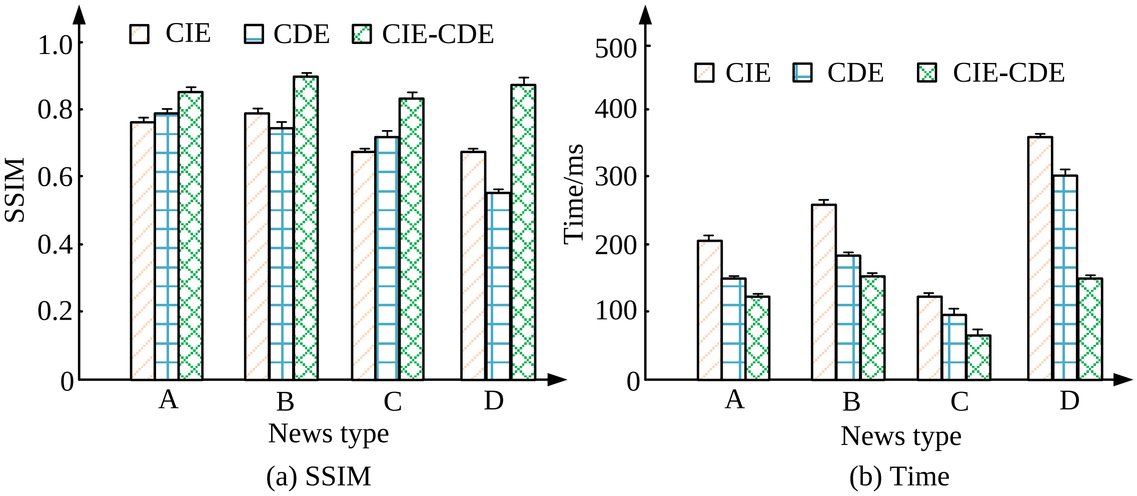
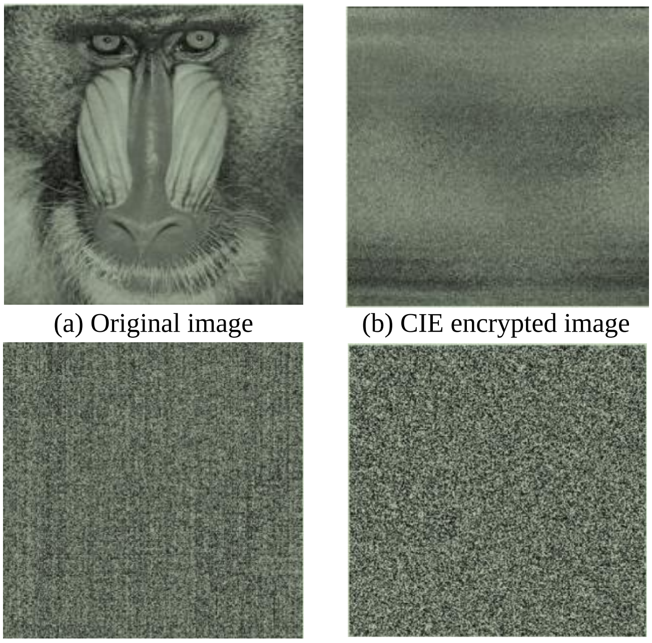


Figure 10: Analysis of SSIM and total encryption and decryption time of the model in different types of news videos.



(a) Original image (b) CIE encrypted image

(c) CDE encrypted image (d) CIE-CDE encrypted image

Figure 11: Analysis of the actual encryption effect of the model.

Table 3: Simulation performance testing.

Type	Model	Entropy	NPCR/%	UACI/%	Pixel correlation	SSIM	Encryption time/ms	Declassified time/ms
A	CIE	7.82	99.18	31.92	0.24	0.76	208	271
	CDE	7.86	99.33	32.58	0.17	0.79	158	224
	CIE-CDE	7.95	99.59	33.01	0.09	0.87	128	182
B	CIE	7.88	99.24	32.11	0.22	0.78	261	287
	CDE	7.91	99.38	32.69	0.15	0.74	186	239
	CIE-CDE	7.96	99.61	33.07	0.08	0.89	146	192
C	CIE	7.79	99.15	31.74	0.25	0.72	135	198
	CDE	7.83	99.32	32.48	0.18	0.75	108	170
	CIE-CDE	7.94	99.57	33.04	0.1	0.85	91	152
D	CIE	7.86	99.2	32.02	0.21	0.71	348	375
	CDE	7.9	99.36	32.66	0.14	0.73	295	324
	CIE-CDE	7.97	99.63	33.12	0.08	0.88	162	189

According to Table 3, in terms of information entropy, CIE-CDE consistently maintained the highest level among the four images, reaching 7.97 in Type D images, indicating that its encryption results are closer to the ideal random distribution. In terms of resistance to differential attacks, the NPCR of CIE-CDE exceeded 99.57% in all four types of images, while UACI remained stable at over 33%, significantly better than CIE's average of 32.0%, indicating that it can effectively disrupt pixel distribution at different image complexities. In terms of structural correlation, CIE-CDE had the lowest pixel correlation, only 0.09 in Type A images and dropping to 0.08 in Type B images, indicating that its encrypted image structural information is highly corrupted and difficult to restore or infer. The SSIM index showed that CIE-CDE also performed well in reversibility, with a maximum of 0.89, far higher than the CIE range of 0.71 to 0.78. In terms of encryption and decryption efficiency, CIE-CDE always maintained the lowest processing time, with an encryption time of only 91 milliseconds in Type C images, which was significantly reduced compared to CIE's 135 milliseconds. In summary, the CIE-CDE model exhibits the best security, efficiency, and stability in all test image types. To demonstrate the performance at the video sequence level, the research designed a comprehensive evaluation of video stream encryption performance, as shown in Table 4.

Table 4 presents the comprehensive performance of the proposed encryption model under complete video stream encryption, compression coding environment and channel interference conditions. In the full video stream encryption test, the lengths of each test video ranged from 30 to 60 seconds, with resolutions covering 720P and 1080P. The results showed that, while maintaining no frame loss, the model's average encryption throughput reached 225 frames per second, and the end-to-end delay was controlled within approximately 400 milliseconds per second of video. The SSIM after decryption was all higher than 0.986, verifying its good adaptability to video continuity. In the H.264 compression environment, despite the existence of quantization loss and edge smoothing phenomena, the SSIM after decryption still remained above 0.91, indicating that the model has strong compression robustness. In the channel interference simulation, under the conditions of introducing Gaussian noise and a 3% packet loss rate respectively, the decrypted image still maintained a good visual structure, with SSIMs of 0.906 and 0.864 respectively, and no frame misalignment or out-of-step phenomenon occurred. Comprehensive analysis shows that this model not only has high-strength encryption capabilities at the static frame level, but also maintains stable security and structural recoverability in dynamic video streams, compression and interference environments, making it suitable for the encrypted dissemination requirements of actual news videos.

To validate the performance and security of the proposed dual-chaotic encryption framework, a comparative evaluation was performed against three widely acknowledged encryption algorithms in recent literature: RC4-Chaos, Logistic-Sine Map Encryption (LSM), and DNA-based Chaotic Image Encryption (DNA-CIE). These methods are selected due to their distinct structural designs-stream cipher enhancement, composite CM, and biologically inspired encryption-thus offering a diversified baseline for comparison. All algorithms were tested on 720p video frames extracted from the TVSum dataset. Performance metrics included encryption time, decryption time, Number of Pixel Change Rate (NPCR), Unified Average Changing Intensity (UACI), information entropy, and SSIM between decrypted and original frames.

As summarized in Table 5, the proposed method achieved NPCR of 99.64% and UACI of 33.51%, indicating strong diffusion capabilities. The entropy value approached the ideal of 8.0, suggesting high randomness in the ciphertext. Additionally, the total runtime remained under 250 ms, demonstrating the method's practical viability for near real-time video encryption scenarios. Compared to RC4-Chaos, which is lightweight but suffers from limited key diffusion, and DNA-CIE, which is highly secure but computationally expensive, the proposed method struck a favorable balance between security strength and computational efficiency. LSTM performed well on statistical metrics but lacks structural adaptability to varied video content types. These results underscore the robustness and generalizability of the proposed framework in practical video security applications.

4.3 Security analysis

To evaluate the robustness of the proposed encryption framework beyond empirical metrics, a theoretical security analysis was presented to examine key space size, resistance to standard attacks, and diffusion strength.

The encryption system was controlled by multiple key components: (1) the initial value x_0 of the logistic map ($\geq 10^{-14}$ precision), (2) the control parameter μ, (3) the parameters α, β, and γ of the crowd simulation, and (4) the initial participant keys for each image dimension. Assuming 64-bit precision and independent parameterization, the total key space exceeded 2^{128}, which is sufficient to resist brute-force attacks under current computational limits. Brute-force attempts were infeasible due to the nonlinearity and high sensitivity of the chaotic sequence to initial seeds. Any minimal variation in x_0 or α, β, γ resulted in a completely different permutation and diffusion sequence. For known-plaintext or chosen-plaintext attacks, the dual mechanism combining position scrambling with pixel diffusion created a non-linear mapping between plaintext and ciphertext. Even when attackers possess pairs of known input-output images, the absence of linearity and high sensitivity prevented reverse-engineering of the original key or structure. The diffusion process introduced strong sensitivity to both pixel values and contextual states. A one-bit change in the plaintext or initial key affected the subsequent chaotic sequence, which in turn modified every pixel's value through cumulative XOR diffusion. Empirical tests showed NPCR > 99.6% and UACI ≈ 33%, but theoretical structure ensured that the system satisfied the avalanche criterion: the output changed significantly even when the input is minimally altered. This is further amplified by the feedback-based chaining structure used in the pixel-wise diffusion.

Table 4: Comprehensive evaluation of video stream encryption performance.

Test Type	Clip/Condition	Resolution	Frame Count	SSIM (Decrypted)	Throughput (fps)	Latency (ms/s)
Full-Stream Test	A (32s)	1280×720	800	0.988	228	405
	B (45s)	1920×1080	1125	0.986	221	417
	C (60s)	1280×720	1500	0.989	230	398
	D (35s)	1920×1080	875	0.987	222	412
	E (58s)	1280×720	1450	0.988	226	414
H.264 Compression	A (3 Mbps)	1280×720	/	0.912	/	/
	B (4.5 Mbps)	1920×1080	/	0.915	/	/
	C (3 Mbps)	1280×720	/	0.909	/	/
Noise Test	Gaussian noise 25 dB	1280×720	/	0.906	/	/
	Packet loss 3%	1280×720	/	0.864	/	/

Table 5: Comparative results with state-of-the-art algorithms.

Method	NPCR (%)	UACI (%)	Entropy	Total Time (ms)	SSIM (Decryption)
Proposed	99.64	33.51	7.9983	243	0.957
RC4-Chaos	97.85	31.02	7.8721	198	0.948
LSM	99.31	32.87	7.9912	275	0.951
DNA-CIE	99.67	33.46	7.9968	384	0.959

In summary, the proposed model satisfies the core requirements of a secure IE system, offering a large and complex key space, high resistance to classical attacks, and robust diffusion characteristics suitable for real-time secure video applications.

5 Discussion

The experimental findings demonstrated that the proposed CIE-CDE encryption algorithm achieved a strong balance among security strength, encryption efficiency, and decryption reversibility. Across all tested video types, the model consistently maintained high entropy values (above 7.95), indicating that the encrypted images approximate ideal randomness and are resistant to statistical analysis. Furthermore, the low pixel correlation coefficients (as low as 0.08) reflected the model's ability to effectively disrupt spatial redundancy in the image structure, which is critical for breaking inter-pixel predictability. The high SSIM, reaching up to 0.95 after decryption, suggested that despite strong scrambling and diffusion, the algorithm preserved the recoverability of the original video frames. This was largely attributed to the design of the plaintext-associated diffusion strategy, which carefully integrated contextual information into the encryption process without compromising structural integrity. In terms of processing performance, the CIE-CDE model significantly reduced the total encryption and decryption time. The encryption time remained under 130 milliseconds across all video types, with the fastest processing observed at just 91 milliseconds. This efficiency gain is primarily due to the lightweight implementation of the dual-stage encryption structure, where the scrambling phase-driven by crowd simulation—requires minimal computation and introduces high variability in pixel positioning. Notably, the incorporation of the crowd simulation mechanism as a key generation and control strategy contributes to the unpredictability of row-column permutations, enhancing resistance against structured attacks without increasing computational load. The logistic-based chaotic diffusion further ensures that minor changes in pixel values propagate widely across the image, reinforcing the algorithm's differential sensitivity. However, several limitations are also apparent. The current model assumes a noise-free and uncompressed transmission environment, which may not fully represent real-world conditions such as wireless streaming or low-bitrate video formats. Additionally, while the algorithm demonstrates excellent performance on full-resolution video frames, its behavior under different resolutions, frame rates, or hardware constraints remains to be tested. Furthermore, the two-phase encryption process, though efficient, still involves multiple iterations and matrix transformations that could pose computational challenges on low-power or embedded systems.

In future research, attention should be given to compression-robust encryption, adaptive key scheduling under varying channel conditions, and potential integration with video coding standards. Exploring these directions will help expand the applicability of the CIE-CDE model to more diverse and dynamic media environments.

6 Conclusion

In response to the security issues of news videos being susceptible to tampering, forgery, and leakage during transmission, a video encryption model that integrates scrambling diffusion mechanism and multidimensional chaos control structure was studied and designed. The model generated a key stream through crowd simulation to enhance key complexity and unpredictable disturbances. The experimental results showed that among different encryption models, CIE-CDE consistently maintained the highest information entropy, reaching 7.97 in type D videos, close to the ideal maximum value of 8, while CIE and CDE were 7.86 and 7.90, respectively, indicating that the model has stronger pixel information obfuscation ability. In terms of pixel difference sensitivity, the NPCR of CIE-CDE was the highest at 99.63%, while UACI was 33.12%, significantly better than CIE's 99.20% and 32.02% and CDE's 99.36% and 32.66%, indicating its superior performance in resisting differential attacks. Especially in type B images, CIE-CDE reduced pixel correlation to 0.08, a decrease of more than 60% compared to CIE's 0.22 and CDE's 0.15, and almost completely dispersed the structural information between encrypted images. Although the model has achieved better results in balancing structural safety and efficiency, there is still room for further improvement. The current method has not yet introduced compression channels and anti noise mechanisms, and the fault tolerance of videos in different compression formats needs to be enhanced. Future research can explore in depth the lightweight design of algorithms, adaptive key adjustment, and robust encryption structures to achieve higher strength, cross platform video security encryption schemes.

The proposed encryption framework, though validated using news video datasets, exhibits structural features and algorithmic components that are generalizable to other video domains such as surveillance footage, medical imaging sequences, and industrial monitoring streams. The modular separation of spatial scrambling and pixel-wise diffusion allows the model to be adapted to varying frame resolutions, temporal continuity constraints, and domain-specific privacy requirements. Furthermore, the rule-based crowd simulation and chaotic control structure can be integrated with dynamic key generation schemes, enabling adaptive key updates per frame or per session, which enhances resilience in long-term secure video transmission. The algorithm also remains lightweight enough to be embedded in resource-constrained environments such as edge cameras or mobile terminals. As emerging cryptographic challenges such as quantum attacks gain attention, the framework can be extended by incorporating quantum-safe key scheduling methods or post-quantum lattice-based encryption in the key exchange stage, ensuring forward security under evolving threat models. These features suggest that the model holds promise for deployment in diverse real-world scenarios requiring high-level video confidentiality and performance stability.

References

[1] Yong Zhang, Ruiyou Li, Yuwen Shi, and Fan Luo. The probabilistic image encryption algorithm based on galois field Gf (257). IETE Journal of Research, 70(7):6286-6299, 2024. https://doi.org/10.1080/03772063.2023.2284956

[2] Lizong Li. Image encryption algorithm based on hyperchaos and DNA coding. IET Image Processing (Wiley-Blackwell), 18(3):627-633, 2024. https://doi.org/10.1049/ipr2.12974

[3] Ming Yao, Zhong Chen, Hongwei Deng, Ximei Wu, Tongzhe Liu, and Can Cao. A color image compression and encryption algorithm combining compressed sensing, Sudoku matrix, and hyperchaotic map. Nonlinear Dynamics, 113(3):2831-2865, 2025. https://doi.org/10.1007/s11071-024-10334-2

[4] Deep Singh, and Sandeep Kumar. Image authentication and encryption algorithm based on RSA cryptosystem and chaotic maps. Expert Systems with Applications, 274(15):141-152, 2025. https://doi.org/10.1016/j.eswa.2025.126883

[5] Anlin Wang, Chuan Shen, Junqiao Pan, Cheng Zhang, Hong Cheng, and Sui Wei. Research on multiple-image encryption method using modified Gerchberg-Saxton algorithm and chaotic systems. Optical Engineering, 62(9):098103.1-098103.14, 2023. https://doi.org/10.1117/1.OE.62.9.098103

[6] Yunhao Liu, and Ru Xue. 3D medical image encryption algorithm using biometric key and cubic S-box. Physica Scripta, 99(5):55035-55055, 2024. https://doi.org/10.1088/1402-4896/ad3b3d

[7] Chao Huang, Ye Tao, and JingWei Zhao. An image encryption algorithm for colour images based on a cellular neural network and the Chua's chaotic system. Journal of Modern Optics, 71(9):321-336, 2024. https://doi.org/10.1080/09500340.2024.2418371

[8] Hangming Zhang, and Hanping Hu. An image encryption algorithm based on a compound-coupled chaotic system. Digital Signal Processing, 146(1):54-59, 2024. https://doi.org/10.1016/j.dsp.2023.104367

[9] Xuenan Peng, and Yicheng Zeng. Image encryption application in a system for compounding self-excited and hidden attractors. Chaos Solitons & Fractals, 139(6):1144-1159, 2020. https://doi.org/10.1016/j.chaos.2020.110044

[10] Mohamed Boussif, Noureddine Aloui, and Adnene Cherif. Securing DICOM images by a new encryption algorithm using Arnold transform and Vigenère cipher. IET Image Processing, 2020, 14(6):1209-1216. https://doi.org/10.1049/iet-ipr.2019.0042

[11] Xingyuan Wang, and Suo Gao. Image encryption algorithm based on the matrix semi-tensor product with a compound secret key produced by a Boolean network. Information Sciences, 539(9):195-214, 2020. https://doi.org/10.1016/j.ins.2020.06.030

[12] Kurunandan Jain, Aravind Aji, and Prabhakar Krishnan. Medical image encryption scheme using multiple chaotic maps. Pattern Recognition Letters, 152(12):356-364, 2021. https://doi.org/10.1016/j.patrec.2021.10.033

[13] M. Zarebnia, H. Pakmanesh, and R. Parvaz. A fast multiple-image encryption algorithm based on hybrid chaotic systems for gray scale images. Optik, 179(3):761-773, 2019. https://doi.org/10.1016/j.ijleo.2018.10.025

[14] Dingkang Mou, and Yumin Dong. Image encryption algorithm based on multiple chaotic systems and improved Joseph block scrambling. Chinese Physics B, 33(10):104205-104206, 2024. https://doi.org/10.1088/1674-1056/ad6257

[15] Moatsum Alawida. A novel image encryption algorithm based on cyclic chaotic map in industrial IoT environments. IEEE Transactions on Industrial Informatics, 20(8):10530-10541, 2024. https://doi.org/10.1109/TII.2024.3395631

[16] Shuqin Zhu, Congxu Zhu, and Hanyu Yan. Cryptanalyzing and improving an image encryption algorithm based on chaotic dual scrambling of pixel position and bit. Entropy, 3(25):59-65, 2023. https://doi.org/10.3390/e25030400

[17] Eligijus Sakalauskas, Antanas Bendoraitis, Dalė Lukšaitė, Gintaras Butkus, and Daiva Vitkutė-Adžgauskienė. Tax declaration scheme using blockchain confidential transactions. Informatica, 34(3):603-616, 2023. https://doi.org/10.15388/23-INFOR531

[18] Sarmad Mahmmod Ahmed, and Baban Ahmed Mahmood. Cloud computing security: Assured deletion. Informatica, 48(3):485-496, 2024. https://doi.org/10.31449/inf.v48i3.6245

[19] Yumin Dong, Chen Xu, and Chenhao Yin. Three-layer quantum image encryption algorithm based on 6D hyperchaos. Journal of Applied Physics, 134(22):224401.1-224401.14, 2023. https://doi.org/10.1063/5.0176657

[20] Qinmao Jiang, Simin Yu, and Qianxue Wang. Cryptanalysis of an image encryption algorithm based on two-dimensional hyperchaotic map. Entropy, 25(3):41-58, 2023. https://doi.org/10.3390/e25030395

[21] Uddagiri Sirisha, and Bolem Sai Chandana. Privacy preserving image encryption with optimal deep transfer learning-based accident severity classification model. Sensors, 23(1):519-520, 2023. https://doi.org/10.3390/s23010519

[22] ShiMing Fu, XueFeng Cheng, and Juan Liu. Dynamics, circuit design, feedback control of a new hyperchaotic system and its application in audio encryption. Scientific Reports, 13(1):19385-19392, 2023. https://doi.org/10.1038/s41598-023-46161-5

[23] Dingkang Mou, and Yumin Dong. Color image encryption algorithm based on novel dynamic DNA encoding and chaotic system. Physica Scripta,

99(6):15-19, 2024. https://doi.org/10.1088/1402-4896/ad3ff1

[24] Gurpreet Kaur, Rekha Agarwal, and Vinod Patidar. Color image encryption scheme based on fractional Hartley transform and chaotic substitution-permutation. The Visual Computer, 38(3):1027-1050, 2022. https://doi.org/10.1007/s00371-021-02066-w

[25] Ram Ratan, and Arvind Yadav. Security analysis of bit plane level image encryption schemes. Defence Science Journal, 71(2):209-221, 2021. https://doi.org/10.14429/dsj.71.15643

A Multi-Scale Feature Extraction and Hierarchical Discriminant Analysis Approach for Image Recognition

Tong Li
Investigation Department, Hebei Vocational College of Public Security Police, Shijiazhuang, 051430, China
E-mail: sunshine_888@163.com

Keywords: Deep learning, feature extraction, hierarchical discriminant analysis, image recognition

Received: Augut 27, 2025

Traditional image recognition algorithms often face problems such as low recognition accuracy and insufficient robustness when facing complex scenes and multi-class image data. To this end, a hierarchical discriminant analysis (HDA)-based image recognition algorithm was proposed, which effectively improves image recognition performance by constructing a multi-scale feature extraction module, principal component analysis (PCA) dimensionality reduction, attention mechanism, and dynamic hierarchical adjustment strategy combined with a hierarchical feature extraction and discrimination model. The experiment was conducted on three public datasets: CIFAR-10, ImageNet subset (selecting 100 categories with a total of 150000 images, based on covering common object categories and moderate data volume for fair validation of algorithm performance), and MNIST. The performance was compared with models such as VGG16, ResNet50, SVM, KNN, Hierarchical CNN, EfficientNet, GoogLeNet, etc. The results indicated that the proposed method had higher recognition accuracy than other comparative algorithms on different datasets, with accuracies exceeding 90%. The proposed method performed better in terms of mean absolute error and root mean squared error. The F1 value curve of the proposed method was located at the top of the coordinate axis, reaching a maximum value of 92.39%, which was 14.56% higher than the lowest value of 78.24% in the EfficientNet model. This algorithm has better recognition accuracy than traditional algorithms on multiple public datasets, and has strong anti-interference ability and robustness, which can provide reference for optimizing the accuracy of image recognition.

Povzetek: Članek predlaga hierarhični algoritem HDA za prepoznavo slik, ki združi ekstrakcijo značilk, PCA, mehanizem pozornosti in dinamično hierarhično prilagajanje ter s tem izboljša natančnost in robustnost glede na uveljavljene modele na več javnih naborih podatkov.

1 Introduction

In the information age, image data grows explosively; image recognition, key for processing image info, is widely used in security, autonomous driving, medical imaging [1]. Traditional algorithms rely on manual features, performing poorly in complex scenes [2]. In recent years, Deep Learning (DL) (e.g., Convolutional Neural Network (CNN)) boosts recognition accuracy via automatic deep feature extraction [3]. Hierarchical Discriminant Analysis (HDA), decomposing complex classification into sub-problems, when combined with DL, is expected to further improve image recognition performance [4]. However, with increasing image data complexity: Zhang et al. used HPLC fingerprint maps + multi-feature quantitative analysis, effectively distinguishing samples from different sources [5]. Yang et al. adopted multi-scale residual modules (capture multi-scale features) + spatial transformation data augmentation (increase feature diversity) + hierarchical discrimination to solve handwritten math expression feature loss, improving recognition accuracy [6]. Su S et al. proposed similar sequence multi-view discriminant correlation analysis to address traditional multi-view feature extraction's ignorance of sample similarity and poor intrinsic manifold capture, achieving better recognition accuracy and robustness [7]. Radmila compared 4 ML algorithms' classification performance on features from 11 pre-trained architectures to solve small-dataset-induced poor classification, finding random forest and multilayer perceptron most suitable [8].

To solve laborious, inefficient manual feature extraction in traditional Φ-OTDR vibration detection, Hu et al. combined 2D image encoding with DL-based vibration recognition and adopted hierarchical discrimination, achieving over 94.25% accuracy [9]. For lighting-induced color deviation and low accuracy, Wu et al. did color correction, used improved watershed and lightweight CNN for feature extraction/fusion, integrated hierarchical discrimination, with fused-feature recognition accuracy at 91% [10]. Zhang et al. proposed a method combining layered discrete entropy and semi-supervised local Fisher discriminant analysis, achieving 100% and 98.2% accuracy in two fault sample identifications [11]. To address the time-consuming sensory analysis and quality grading issues of Louis Boissier tea in the production area, Janine C. and her team used shortwave infrared hyperspectral imaging, combined with partial least squares discriminant analysis and layered modeling for classification, followed by preprocessing and parameter optimization. The results indicated that the classification accuracy of the production area was 100% [12].

Table 0: Summary table of related works

Author	Method	Dataset	Performance metrics
Zhang et al.[5]	Image feature maps + multi-feature quantitative analysis + hierarchical discrimination	Honeysuckle origin samples	Effective origin discrimination
Yang et al.[6]	Multi-scale residual module + data augmentation + hierarchical discrimination	Handwritten math expressions	Improved recognition accuracy
Su S et al.[7]	Similar sequence multi-view discriminant correlation analysis + hierarchical discrimination	Universal images	Better accuracy & robustness
Radmila[8]	Feature extraction + classification comparison + hierarchical discrimination	Cultural heritage images	Feature extraction accuracy: 88.89%-95.56% (partial architectures)
Hu et al.[9]	2D image encoding + DL feature recognition + hierarchical discrimination	6 types of OTDR vibration images	Vibration recognition accuracy >94.25%
Wu et al.[10]	Color correction + improved watershed + lightweight CNN + feature fusion + hierarchical discrimination	Stratigraphic images	Post-fusion accuracy: 91%
Zhang et al.[11]	Hierarchical discrete entropy + semi-supervised Local Fisher analysis + hierarchical discrimination model	2 types of bearing fault signals	Fault recognition accuracy: 100%, 98.2%
Janine C. et al.[12]	Preprocessing + SWIR hyperspectral imaging + PLS-DA + hierarchical modeling	Louis Boissier tea SWIR images	Origin classification & quality grading accuracy

Different teams studied diverse images: Zhang et al. used chromatographic fingerprinting and hierarchical discrimination to distinguish honeysuckle origin; Yang et al. used multi-scale residuals to boost handwritten math expression recognition accuracy but lacked complex feature dynamic discrimination; Su et al. processed general images via multi-view discriminant analysis without attention mechanisms; Radmila analyzed cultural heritage images with transfer learning, relying on pre-trained models; Hu et al. converted 1D signals to images for vibration event recognition without optimizing multi-scale fusion.

Despite existing research applying hierarchical thinking to image recognition, three key gaps remain: first, feature extraction lacks specificity (relies on single-scale/pre-trained models, fails to fully capture image details, local/global features); second, fixed hierarchical discrimination structure (no dynamic adjustment of depth/parameters, limiting complex data accuracy); third, insufficient integration of attention mechanisms and dimensionality reduction (prone to redundancy or non-critical feature interference).

To this end, a HDA-based image recognition algorithm is proposed, which innovatively designs a multi-scale feature extraction module to obtain comprehensive features, combines principal component analysis (PCA) dimensionality reduction to reduce redundancy, introduces attention mechanism to focus on key features, and optimizes the discrimination structure through dynamic hierarchical adjustment strategy. Ultimately, the recognition accuracy and robustness are improved, laying the foundation for the engineering application of image recognition technology.

2 Research design

2.1 HDA algorithm based on multi-scale image feature extraction

To achieve the three major objectives of 'improving recognition accuracy, noise robustness, and controlling computational complexity' as stated in the introduction, the technical route of the research design is elaborated in detail. Through the organic combination of multi-scale feature extraction, PCA dimensionality reduction, attention mechanism, and dynamic hierarchical adjustment strategy, the core research questions are addressed one by one to ensure that the design logic is highly matched with the research objectives.

The study adopts the channel wise attention mechanism without introducing spatial attention - the core reason is that the multi-scale feature extraction module has captured the spatial details and global information of the image through convolution kernels of different sizes. Channel attention can further enhance the importance differentiation of different channel features (such as in the MNIST dataset, where the channel weights of digital contour features are higher), avoiding functional redundancy between spatial attention and multi-scale modules.

Hierarchical clustering (unsupervised) splits/merges via sample similarity (no feature discriminators, fixed results); this study’s HDA (supervised) uses Support Vector Machine (SVM) discriminators (trained on annotated samples) and dynamic structure optimization. Multi-level convolution only does hierarchical feature extraction (no independent discrimination, relies on single classification head); this study’s HDA combines feature extraction and hierarchical discrimination. In image recognition, feature extraction quality affects the result. Traditional feature extraction extracts only shallow features, while DL-based single-scale feature extraction fails to fully describe image complex structure [13]. Thus, an HDA algorithm via multi-scale feature extraction and hierarchical discrimination is developed to boost feature representation, category discrimination, and recognition accuracy/robustness [14-15]. Its multi-scale feature extraction module uses different-scale CKs for convolution to get multi-scale image features (in Figure 1).

In Figure 1, the input image is first standardized, and then finite element analysis is performed using three

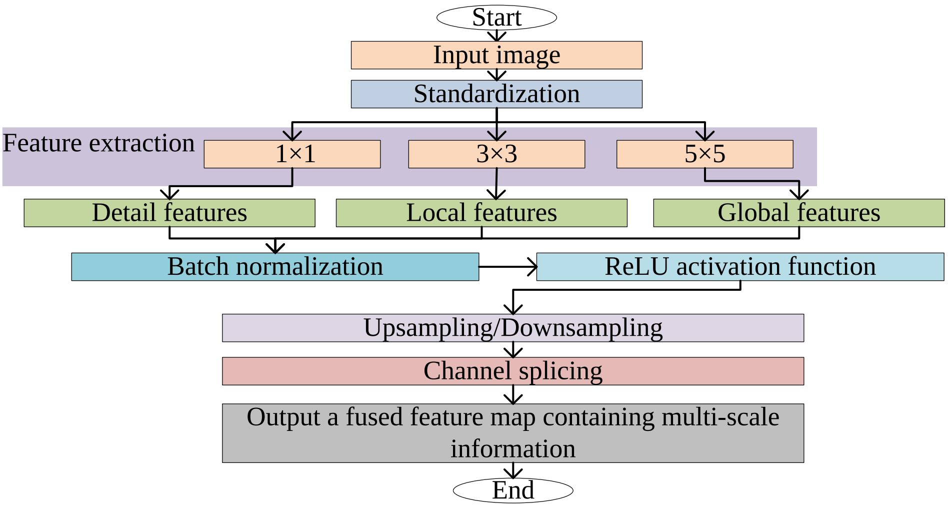


Figure 1: Multi-scale feature extraction module.

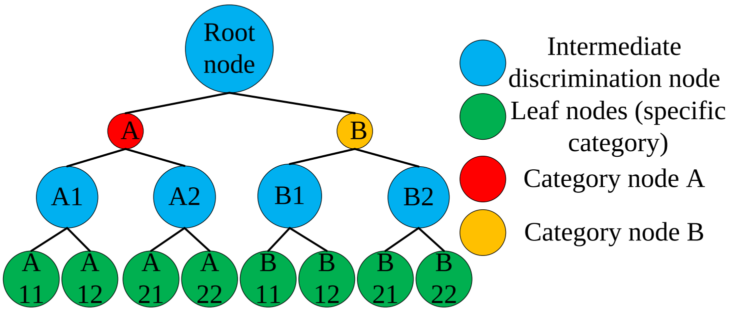


Figure 2: Hierarchical discriminant model tree structure.

convolution branches of different scales. After each convolution branch, there are cascaded batch normalization and ReLU activation functions to accelerate the convergence speed of the network and enhance its nonlinear expression ability.

The hierarchical discrimination model achieves a gradual discrimination of 'coarse classification fine classification' through a tree structure, and its specific structure is as follows: The hierarchical discrimination model adopts a tree structure, where each node represents a discriminator used for classifying and discriminating input features. The root node corresponds to the highest level of discrimination, dividing all images into several major categories. Each child node corresponds to the discrimination of the next layer, and the large class divided by its parent node is further subdivided into smaller subclasses until the leaf node corresponds to a specific category [16]. The schematic is shown in Figure 2.

The input image is set to be $X \in R^{H\times W\times C}$, where H, W, and C are the height, width, and amount of channels of the image. After the convolution operation of the k th convolution $(k=1,2,3)$ branch, the feature map obtained is $F_k \in R^{H_k\times W_k\times C_k}$, which is calculated as shown in equation (1).

$$F_k = \mathrm{Re}\,LU(BN(W_k * X + b_k)) \qquad (1)$$

In equation (1), W_k and b_k respectively represent the CK and bias term of the k th convolution branch, $*$ refers to the convolution operation, $BN(\cdot)$ is the batch normalization operation, and $\mathrm{Re}\,LU(\cdot)$ means the ReLU activation function. The features were projected in layers to enable discrimination. For the m th subset of features in the l th layer, the intra-class dispersion matrix $s_w^{l,m}$ is calculated as denoted in equation (2).

$$s_w^{l,m} = \sum_{c=1}^{C_{l,m}} \sum_{x\in S_{l,m,c}} (x-\mu_{l,m,c})(x-\mu_{l,m,c})^T \qquad (2)$$

In equation (2), $C_{l,m}$ is the amount of categories contained in the feature subset, the c th class sample set is labeled as $S_{l,m,c}$, and the mean vector of the c th class sample is labeled as $\mu_{l,m,c}$ [17]. The discrimination criteria between feature layers are as follows, and the inter-class dispersion matrix S_b^l of the l th layer is calculated as shown in equation (3).

$$S_b^l = \sum_{m=1}^{M_l} N_{l,m}(\mu_{l,m}-\mu_l)(\mu_{l,m}-\mu_l)^T \qquad (3)$$

In equation (3), M_l means the amount of feature subsets in the l th layer, $N_{l,m}$ means the total amount of samples in the m th feature subset, the mean vector of the

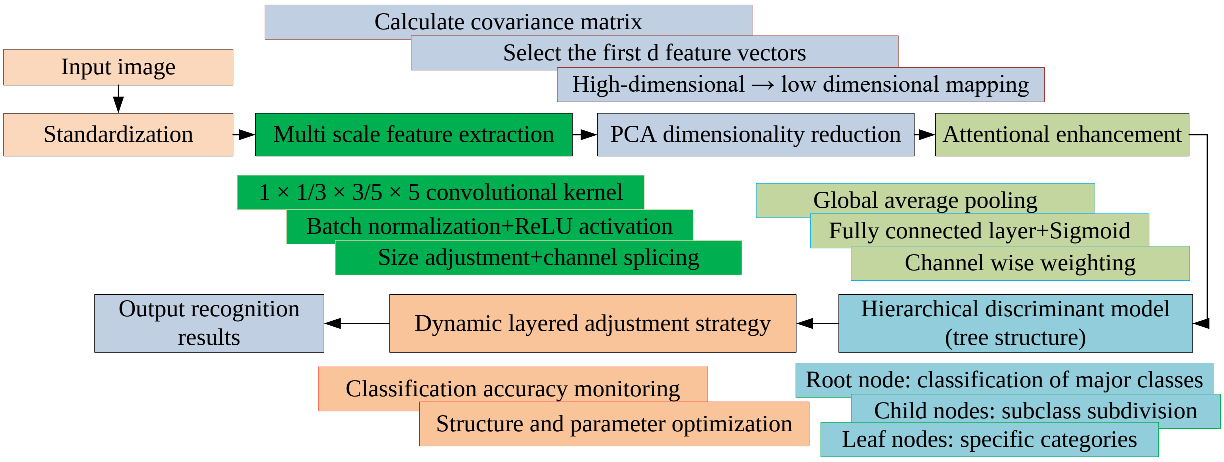

Calculate covariance matrix
Select the first d feature vectors
High-dimensional → low dimensional mapping
Input image
Standardization
Multi scale feature extraction
PCA dimensionality reduction
Attentional enhancement
1 × 1/3 × 3/5 × 5 convolutional kernel
Batch normalization+ReLU activation
Size adjustment+channel splicing
Global average pooling
Fully connected layer+Sigmoid
Channel wise weighting
Output recognition results
Dynamic layered adjustment strategy
Hierarchical discriminant model (tree structure)
Classification accuracy monitoring
Structure and parameter optimization
Root node: classification of major classes
Child nodes: subclass subdivision
Leaf nodes: specific categories

Figure 3: HDA algorithm flow based on multi-scale image feature extraction module.

m th feature subset is labeled as $\mu_{l,m}$, and μ_l denotes the total mean vector of all samples in the l th layer [18].

To integrate feature information of different scales, the feature maps obtained from the three branches are upsampled or downsampled to make their sizes consistent, and then channel concatenation is performed to obtain the fused feature map $F \in R^{H\times W\times(C_1+C_2+C_3)}$, as shown in equation (4).

$$F = Concat(F_1', F_2', F_3') \tag{4}$$

In equation (4), F_k' represents the feature map of the k th branch after size adjustment, and $Concat(\cdot)$ represents the channel concatenation operation. Because of the high dimensionality of the fused feature map, it will increase the computational complexity of subsequent processing, thus requiring feature dimensionality reduction. To preserve the main feature information, PCA algorithm is employed to minimize the dimensionality of the fused features. This study uses bilinear interpolation to adjust the size of feature maps: for feature maps smaller than the target size, bilinear interpolation is used for upsampling - based on the grayscale values of four adjacent pixels around the target pixel, weighting coefficients are calculated according to the distance between pixels, and the target pixel value is obtained by weighted averaging.

This study chose PCA as the dimensionality reduction method because LDA requires category labels and is sensitive to overfitting in the ImageNet subset of this study where there are few category samples. PCA, on the other hand, is unsupervised and does not require labels, making it suitable for "dimensionality reduction before discrimination"; T-SNE and UMAP have high computational complexity and are prone to losing global information, while PCA has low complexity and preserves global variance, making it more suitable for multi-level discrimination.

Let the fused feature matrix be $F \in R^{N\times D}$, where N refers to the amount of samples and D means the feature dimension. The target of PCA is to find a projection matrix $P \in R^{D\times d}\left(d < D\right)$, project the high-dimensional feature matrix F onto a low dimensional space, and obtain the reduced dimensional feature matrix $F_p \in R^{N\times d}$. The calculation is shown in equation (5).

$$F_p = F \times P \tag{5}$$

In equation (5), the projection matrix P is composed of the eigenvectors corresponding to the first d largest eigenvalues of the covariance matrix of the feature matrix F. The calculation of covariance matrix C is shown in equation (6).

$$C = \frac{1}{N-1} F^T (F - \bar{F}) \tag{6}$$

In equation (6), $\bar{F}$ refers to the mean vector of the feature matrix F. To preserve the main feature information, this study used PCA algorithm to reduce the dimensionality of the fused features after multi-scale feature fusion and before attention mechanism processing. In the PCA dimensionality reduction, the determination of the low dimensional spatial dimension d in the preserved variance threshold is based on the principle of "preserving 95% variance" - that is, selecting the top d largest eigenvalues of the covariance matrix, so that the cumulative sum of these eigenvalues' accounts for ≥ 95% of the total sum of all eigenvalues. The core logic of PCA dimensionality reduction is to map high-dimensional features to a low dimensional space through linear transformation, while maximizing the preservation of variance information in the data. Specifically, for the fused feature matrix, the covariance matrix is calculated, which reflects the degree of linear correlation between features.

Through feature dimensionality reduction, not only does it reduce computational complexity, but it also reduces redundant information between features, which is beneficial for improving the efficiency and accuracy of subsequent hierarchical discrimination. The HDA algorithm based on multi-scale image feature extraction module is shown in Figure 3.

The initial tree structure of the HDA model adopts a "top-down" construction approach, where the root node uses all categories as discriminative objects. By calculating the feature differences between categories, categories with feature differences greater than the threshold T1 are divided into different child nodes; The child nodes continue to be divided based on this rule until each leaf node corresponds to only one category. As shown in Figure 3, input images are preprocessed first. The multi-scale feature extraction module uses different-size CKs for feature extraction (with batch normalization and ReLU), adjusts, splices and fuses them. Then PCA selects top d eigenvectors to reduce redundancy. Attention mechanism generates weights via global average pooling, fully connected layers and Sigmoid to highlight key features. Finally, tree hierarchical discrimination model discriminates layer by layer (with dynamic adjustment) and outputs recognition results.

2.2 Image recognition algorithm based on HDA algorithm

Multi-scale feature extraction module yields rich image features, which become low-dimensional and representative after dimensionality reduction. Yet effective use of these features for recognition is key. Thus, an HDA-based image recognition model is built, decomposing recognition into sub-tasks via feature hierarchy and category relationships to narrow scope and boost accuracy [19-20]. For example, in the MNIST dataset (10 handwritten digit categories), the root node first calculates the feature difference between the 10 categories, and divides the categories with a difference>0.6 into three primary sub nodes (such as {0,1,2}, {3,4,5}, {6,7,8,9}). Each primary sub node is then divided into secondary sub nodes according to the same rules, ultimately forming a tree structure with leaf nodes corresponding to a single digit category.

The category set of images is $C=\{c_1,c_2,\ldots,c_M\}$, where M means the total amount of categories. Based on the semantic relationships and feature similarities between categories, the category set C is divided into K_1 major categories $C_1^1,C_2^1,\ldots,C_{K_1}^1$. Each major category C_i^1 can be further divided into K_2 subcategories $C_{i1}^2,C_{i2}^2,\ldots,C_{iK_2}^2$; And so on, until it is assigned to a specific category. For each discriminative node, a SVM is used as the discriminator. SVM can effectively classify data in high-dimensional space by finding the optimal classification hyperplane. If the training sample feature of a discrimination node is $x_i \in R^d$ and the corresponding category label is $y_i \in \{0,1,\ldots,K\}$ (K is the number of categories that the node needs to be classified into), then the objective function of SVM is shown in equation (7).

$$\begin{aligned}&\min_{w,b,\xi}\frac{1}{2}\|w\|^2+\varepsilon\sum_{i=1}^{n}\xi_i\\&s.t.\,y_i(w\cdot\phi(x_i)+b)\geq 1-\xi_i,\xi_i\geq 0\end{aligned}\tag{7}$$

In equation (7), w and b respectively represent the normal vector and bias term of the classification hyperplane, $\phi(x_i)$ represents the function that maps feature x_i to a high-dimensional space, ξ_i represents the relaxation variable, and ε represents the penalty parameter [21-22].

To classify and discriminate new samples, research solves the above optimization problem to obtain the optimal classification hyperplane. Based on the extracted and processed hierarchical features, the study calculates the distance between the sample and the center of each layer category, as well as the weights of each layer for comprehensive discrimination. For the distance between the sample and the class centers of each layer, the class distance of the sample in the l th layer is shown in equation (8).

$$d_l(z,c)=(z_l-\nu_{l,c})^T(S_w^l)^{-1}(z_l-\nu_{l,c})\tag{8}$$

In equation (8), z represents the sample, z_l represents the projected features of the l th layer, and $\nu_{l,c}$ represents the center of the c th class in that layer. The discriminative weight w_l of the l th layer is determined based on its discriminative ability and calculated as shown in equation (9).

$$\omega_l=\left(\frac{tr(S_b^l)}{tr(S_w^l)}\right)\Bigg/\sum_{k=1}^{L}\frac{tr(S_b^k)}{tr(S_w^k)}\tag{9}$$

In equation (9), w_l represents the discriminative weight, L denotes the total amount of layers, and $tr(\square)$ denotes the trace of the matrix. The comprehensive discrimination score $Score(z,c)$ for sample z belonging to category c is the weighted result of the distance between each layer, as shown in equation (10).

$$Score(z,c)=-\sum_{l=1}^{L}\omega_l d_l(z,c)\tag{10}$$

In equation (10), $Score(z,c)$ represents the comprehensive discrimination score of sample z belonging to category c. In addition, to enhance the adaptability and accuracy of the hierarchical discrimination model, a dynamic hierarchical adjustment strategy is proposed. This strategy dynamically adjusts the hierarchical structure and discriminator parameters based on the classification accuracy of each discriminative node and the feature differences between categories. The dynamically adjusted classification accuracy threshold is set to 85%, and the category feature difference threshold is set to 0.3; The frequency of structural updates is only dynamically adjusted during the model training phase, triggered once every 10 rounds of training; The computational cost mainly comes from retraining the discriminator (SVM) after node splitting/merging. The mechanism is shown in Figure 4.

Dynamic adjustment is performed every 10 rounds during the training phase. The adjustment logic is as follows: when the classification accuracy AcCi of a discriminative node is less than the threshold T2, the node is "split and adjusted" - the corresponding category of the node is re divided into 2 new child nodes based on feature differences, and an SVM discriminator is trained for the new node; When the feature difference between adjacent child nodes is less than T3 and the merged classification accuracy is greater than or equal to T2, perform "merging adjustment" - merge the two child nodes into one node and retrain the SVM discriminator.

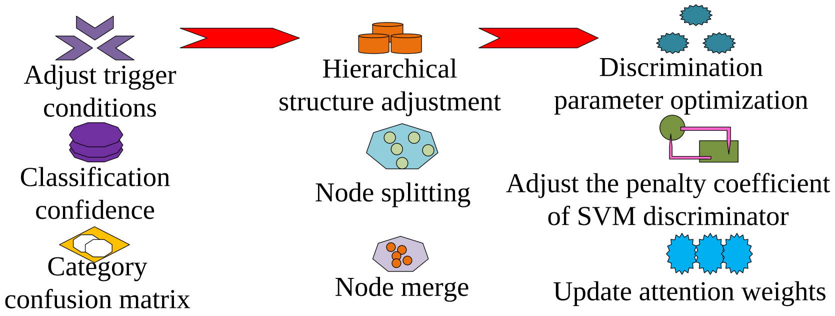

Adjust trigger conditions
Hierarchical structure adjustment
Discrimination parameter optimization
Classification confidence
Node splitting
Adjust the penalty coefficient of SVM discriminator
Category confusion matrix
Node merge
Update attention weights

Figure 4: Dynamic layered adjustment strategy.

During the adjustment process, the computational cost can be offset by parallel training of the sub node discriminator, without affecting the overall training efficiency. In Figure 4, during the model training, the classification accuracy Acc_i of each discriminative node is calculated. When Acc_i is lower than the preset threshold, it indicates that the classification performance of the node is poor and the corresponding hierarchical structure needs to be adjusted. Meanwhile, based on the feature difference degree D_{ij} between categories (utilized to measure the feature difference between category i and category j), the parameters of the discriminator are optimized to improve its ability to distinguish categories with significant differences. The calculation of feature difference D_{ij} is shown in equation (11).

$$D_{ij} = \frac{1}{n_i n_j} \sum_{x \in c_i} \sum_{y \in c_j} \| x - y \|_2 \tag{11}$$

In equation (11), n_i and n_j express the sample sizes of category i and category j, respectively, and $\| \cdot \|_2$ represents the L2 norm. By dynamically adjusting the layering strategy, the model can adaptively optimize the layering structure and discriminator based on the characteristics of the data, thereby improving the accuracy of image recognition.

The calculation of inter-layer class distance, discriminant weight, comprehensive score, and feature difference degree refers to the specific formulas in Appendix A (Equations A8–A11), and the core logic is as follows: the inter-layer distance reflects the similarity between samples and category centers, the discriminant weight is determined by the discriminative ability of each layer, the comprehensive score is the weighted sum of inter-layer distances, and the feature difference degree measures the distinction between different categories.

To make the model pay more attention to key regions in the image and improve the targeting of features, attention mechanism is introduced after feature extraction. The feature map obtained through feature extraction and dimensionality reduction is referred to as $F_p \in \square^{H \times W \times d}$, and the calculation process of the attention mechanism is as follows. Firstly, a global average pooling operation on the feature map F_p is performed to obtain the global feature vector g, as shown in equation (12).

$$g = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_p(i, j, \cdot) \tag{12}$$

Then, the attention weight a is calculated using a fully connected layer and Sigmoid activation function, as shown in equation (13).

$$a = Sigmoid(W_a g + b_a) \tag{13}$$

In equation (13), W_a and b_a represent the weights and bias terms of the fully connected layer, respectively. Finally, the attention weights are multiplied with the feature map F_p channel by channel to obtain the weighted feature map, as shown in equation (14).

$$F_a(i, j, t) = F_p(i, j, t) \times a(t) \tag{14}$$

In equation (14), t represents the feature channel index. The feature map processed by the attention module (Equation 14) is first transformed into a 1×1×C feature vector (C is the number of feature channels) through global average pooling, and then input into the SVM discriminator of each node in the tree structure. For example, in the second level sub nodes (corresponding to categories {3,4,5}) of the MNIST dataset, the feature vector with a dimension of 1×1×256 is obtained by pooling the F_att, which serves as the input feature for SVM to distinguish between categories 3, 4, and 5. By introducing attention mechanism, the model can pay more attention to key features in the image, enhancing the discriminative ability of the features. During the iteration process, the feature subset of the l th layer is updated based on the recognition results. For misclassified samples x, their feature z_l is adjusted as shown in equation (15).

$$z_l' = z_l + \alpha (v_{l,\hat{c}} - z_l) \tag{15}$$

In equation (15), α denotes the learning rate, and $v_{l,\hat{c}}$ denotes the center of the predicted category $\hat{c}$ in the l th layer.

3 Results and analyses

3.1 Experimental preparation and setup

To test the effect of the designed algorithm, three publicly available image datasets were used for experiments, namely the CIFAR-10 dataset, which includes10 categories of color images with 6000 images per category. The image size ess 32 × 32. To verify the effectiveness of the proposed combination strategy of 'multi-scale feature extraction+PCA dimensionality reduction+attention mechanism+dynamic hierarchical adjustment', it first clarified the context of the experimental design and related research: current image recognition experiments mostly use CIFAR-10 (small-sized color images) and MNIST (handwritten digits) to verify basic accuracy, and use ImageNet subsets to verify complex category adaptability. The experiment selected 100 categories from the ImageNet dataset (covering 6 common objects such as animals, plants, and transportation, with category numbers n01440764-n01443537, n01629819-n01630670, etc.), and selected 1500 images for each category (1200 in the training set and 300 in the testing set), for a total of 150000 images. The reasons for choosing this subset are: firstly, it covers multiple image types, which can verify the generality of the algorithm; The second is to have a moderate amount of data to avoid the training cycle being too long due to a large amount of data, or overfitting the model due to a small amount of data.

Although the maximum training epochs in this study were set to 100, an Early Stopping strategy was also introduced to avoid overfitting and optimize training efficiency. The validation set loss (cross entropy loss) was used as the monitoring metric, and when the validation set loss did not decrease for 5 consecutive epochs (i.e., the loss value fluctuation was ≤ 0.001), the training was automatically stopped and the current optimal model parameters were saved.

The control variable settings for the ablation experiment: except for 'whether attention mechanism is enabled', all other parameters (multi-scale feature extraction convolution kernel size, PCA dimensionality reduction preserving 95% variance, SVM discriminator parameters) are completely consistent to ensure that the experimental results are only caused by whether attention mechanism is enabled, and to verify the rigor of the conclusions. The adjustment of the strategy only occurred during the training phase, and no structural updates were performed during the inference phase, which affects real-time processing. Under the current design, although the training phase increased the total time by 8%, the inference phase only took 0.03 seconds for single sample recognition due to fixed structure (based on the configuration in Table 1). Compared with ResNet50 (0.04 seconds/sample) and Hierarchical CNN (0.05 seconds/sample), it still has real-time advantages and can be adapted to conventional real-time scenarios (such as security monitoring image capture recognition, which requires single frame processing time<0.1 seconds). MNIST dataset: contains handwritten digit images of 10 categories, with 6000-7000 images per category, and image sizes of 28 × 28. In the experiment, baseline models such as VGG16, ResNet50, EfficientNet, GoogLeNet, etc. were all based on PyTorch's official open-source implementation (version 1.12.0) and trained under the same experimental conditions as the algorithm in this paper (learning rate of 0.001, batch size of 64, no data augmentation, and 100 training epochs); SVM and KNN models were implemented based on the Scikit learn library, and the input features were consistent with the PCA reduced features of our algorithm, ensuring fairness in comparison.

This study did not use data augmentation for two reasons: first, to verify the algorithm's own feature extraction and discrimination ability, eliminate augmentation interference, and ensure results reflect core module effectiveness; second, future augmentation experiments (random flipping, cropping, color jitter) will verify generalization. This study focuses on basic performance verification, so augmentation is temporarily not introduced. All dataset results underwent t-test (95% confidence level): on CIFAR-10, p-value for our algorithm-ResNet50 accuracy difference (1.8%) was $0.021<0.05$; on ImageNet subset, p-value for 3.3% difference was $0.015<0.05$, showing significant accuracy improvement. All results are averages of 5 independent trainings, with standard deviation <1.2%, proving model stability.

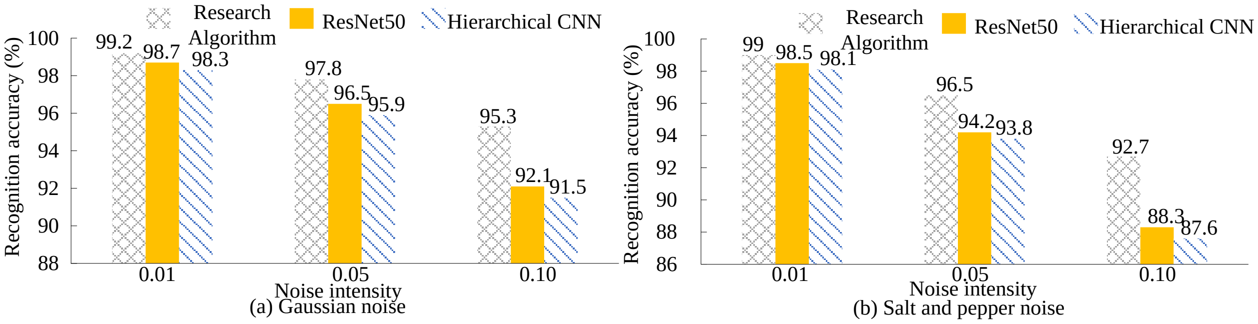
All noise experiment results used 95% confidence intervals (from 5 independent data): For CIFAR-10 (Gaussian noise variance 0.1), this algorithm’s accuracy interval was [88.7%, 89.7%], ResNet50 [83.9%, 85.1%], Hierarchical CNN [83.1%, 84.5%]; For ImageNet subset (variance 0.1), this algorithm’s interval was [79.6%, 81.0%], while compared algorithms (e.g., ResNet50 [74.3%, 75.9%]) has wider intervals. This proves the algorithm has smaller performance fluctuations and more stable robustness under noise.

Using real-world noise datasets, this algorithm achieved an accuracy of 91.7%, which was 4.5% higher than ResNet50 (87.2%) and 6.4% higher than XGBoost ensemble model (85.3%), demonstrating its robustness in non synthetic noise real-world scenarios. Considering the privacy requirements of research data and technical details (such as engineering optimization parameters of algorithm core modules and customized processing logic adapted to specific scenarios), the experimental code of this study was not yet fully open sourced.

The specific retention dimensions d for different datasets are as follows: the feature dimension of the CIFAR-10 dataset after fusion was 2048, and according to the 95% variance retention principle, the first d=512 principal components were selected, and the projection matrix P dimension was 2048 × 512; After the fusion of ImageNet subsets, the feature dimension was 4096. The first d=1024 principal components were selected, and the projection matrix P dimension was 4096 × 1024; After the fusion of the MNIST dataset, the feature dimension was 784. The first d=256 principal components were selected, and the projection matrix P dimension was 784 × 256. The weights and biases of the fully connected layer in the attention mechanism were initialized as follows: the weights of the first fully connected layer were initialized using He normal state, and the biases were initialized to 0; The weights of the second fully connected layer were initialized using Xavier normal and the bias was initialized to 0; After initialization, the initial value of attention weight a was calculated using

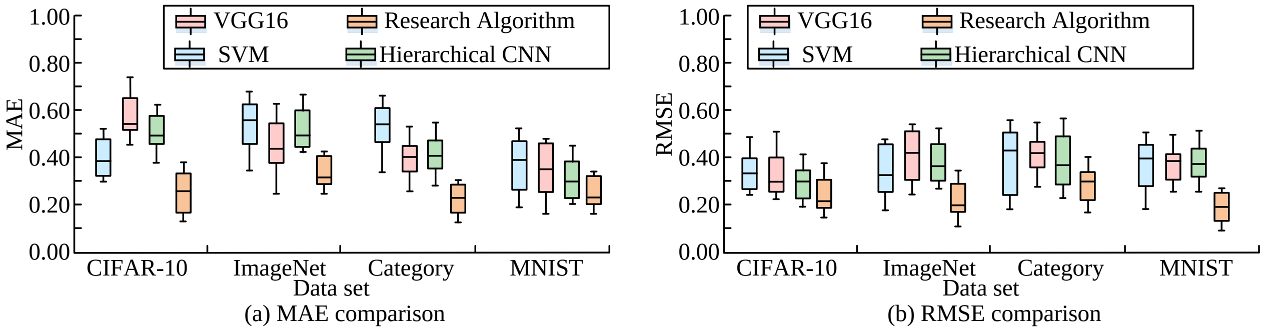
Table 1: Recognition accuracy (%) of different algorithms on various datasets.

Algorithm	CIFAR-10	ImageNet subset	MNIST
VGG16	89.2	78.5	98.3
ResNet50	92.5	82.3	99.1
SVM	78.6	65.2	97.5
KNN	75.3	60.8	96.8
Hierarchical CNN	90.1	79.8	98.7
ViT-B/16	93.1	84.2	99.3
Swin-T	93.7	84.8	99.4
Proposed method	94.3	85.6	99.5



(a) Gaussian noise (b) Salt and pepper noise

Figure 5: Recognition accuracy under different noise intensities (MNIST dataset,%).



(a) MAE comparison (b) RMSE comparison

Figure 6: Comparison of MAE and RMSE for different recognition algorithms on different datasets.

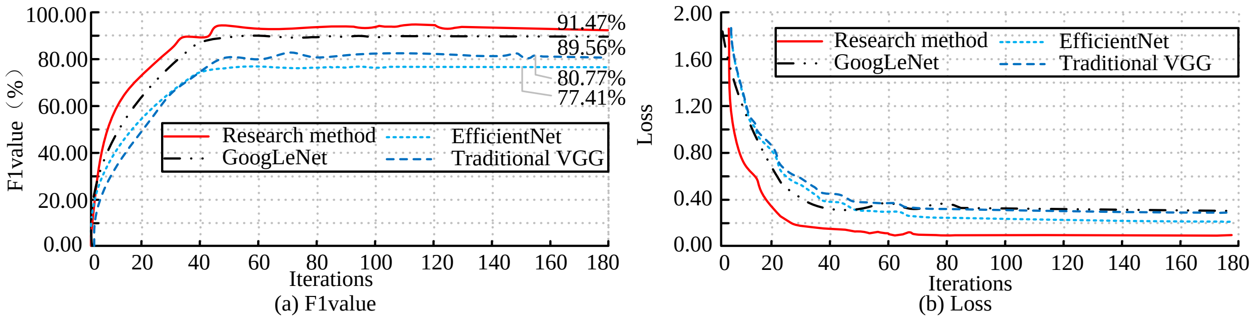
Sigmoid, and the initial mean was controlled at around 0.5 to avoid training instability caused by initial weights that are too large or too small. In this study, the SVM discriminators all used radial basis kernel functions, with the kernel function parameter γ set to 1/d, and were implemented using the SVC class in the Scikit learn library. The key threshold and determination method for dynamic adjustment are as follows: Node classification accuracy threshold of 85%: determined on the validation set through 5-fold cross validation, with a testing threshold range of 80%-90%.

3.2 Analysis of verification results of image recognition methods

The recognition accuracy of the designed algorithm compared to other comparative algorithms on three datasets is shown in Table 1. On three datasets, the recognition accuracy of the proposed algorithm was higher than that of other compared algorithms, at 94.3%, 85.6%, and 99.5%, respectively. On the CIFAR-10 dataset, the recognition accuracy of the proposed method was 1.8% higher than that of ResNet50, 3.3% higher on the ImageNet subset, and 0.4% higher on the MNIST dataset.

Figure 5 shows the recognition accuracy under different noise intensities (MNIST dataset,%). In Figure 5 (a), in a Gaussian noise scene, when the noise variance increased from 0.01 to 0.1, the accuracy of the proposed method decreased from 99.2% to 95.3%, with a decay amplitude of only 3.9%. However, the decay amplitudes of ResNet50 and Hierarchical CNN reached 6.6% and 6.8%, respectively. When the variance of Gaussian noise was 0.1, the accuracy of the proposed method was 95.3%, while ResNet50 and Hierarchical CNN were 92.1% and 91.5%, respectively. In Figure 5 (b), in a salt and pepper noise scene, the accuracy of the proposed method was 92.7% when the noise variance was 0.1, which was 4.4% higher than ResNet50 and 5.1% higher than Hierarchical CNN, and its attenuation rate was significantly lower than ResNet50 and Hierarchical CNN.

The experiment selected Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as evaluation metrics, and the experimental outcomes are denoted in Figure 6. In Figures 6 (a)-6 (b), compared to VGG16, SVM, and Hierarchical CNN on CIFAR-10, ImageNet, Category, and MNIST datasets, the proposed method



(a) F1value

(b) Loss

Figure 7: Performance comparison of different image recognition models.

Table 2: Multi dataset evaluation under different methods.

Dataset	Evaluation metrics	Proposed algorithm	Comparative algorithms	The relative improvement of the proposed algorithm	/
CIFAR-10	Precision	0.941	ResNet50（92.3%）	0.018	/
CIFAR-10	Recall	0.945	ResNet50（92.7%）	0.018	/
ImageNet subset	Precision	0.853	EfficientNet （81.2%）	0.041	/
ImageNet subset	Recall	0.859	EfficientNet （80.9%）	0.05	/
MNIST	Precision	0.994	Hierarchical CNN （98.6%）	0.008	/
MNIST	Recall	0.996	Hierarchical CNN （98.8%）	0.008	/
Dataset	Confused categories	The misjudgment rate	Comparative algorithms	Comparison algorithm misjudgment rate	The reduction in false positive rate
CIFAR-10	Airplane - Bird	1.2%-1.5%	VGG16	3.8%-4.2%	＞65%
CIFAR-10	Car Truck	1.2%-1.5%	VGG16	3.8%-4.2%	＞65%
ImageNet subset	Dog wolf, cat tiger	0.032	RobustCNN	0.058	0.448

performed better in terms of MAE and RMSE (Figure 6 (b)). The overall deviation of the box line indicates that the error value was smaller and the fluctuation was narrow. For example, on the MNIST dataset, the MAE of the proposed method was 0.021, which was 45.9% lower than VGG16 (0.039) and 31.0% lower than ResNet50 (0.030); the RMSE was 0.053, which was 38.8% lower than VGG16 (0.087) and 27.4% lower than ResNet50 (0.073). This data showed that the Research Algorithm had smaller deviations between predicted and true values and higher stability in image recognition across different datasets.

The performance of research method was compared with existing advanced image recognition models, including traditional CNN VGG, image classification models EfficientNet, and GoogLeNet. The F1 value and loss curve of the models were used as evaluation indicators, and the average test results of different datasets are denoted in Figure 7. The Loss Function (LF) is called Cross Entropy Loss, which is used to measure the difference between the predicted values of the model and the true labels. The smaller the value, the better the fitting effect of the model. In Figure 7 (a), the F1 value curve of the research method was located at the top of the coordinate axis, reaching a maximum value of 92.39%, which was 14.56% higher than the lowest value of 78.24% in the EfficientNet model. The F1 values of the other two image recognition models were within the range of 80-90%. Figure 7 (a) shows different models' LF curves. The proposed method's LF curve converges to the minimum, with a steady decline and the fastest convergence. LF reflects prediction-true value consistency; smaller LF means better fitting, so the method has better comprehensive performance.

Table 2 shows multi-dataset evaluation of different methods, where the proposed algorithm performed better. On CIFAR-10, its accuracy (94.1%) and recall (94.5%) were both 1.8% higher than ResNet50. On ImageNet subset, accuracy (85.3%) and recall (85.9%) were 4.1% and 5.0% higher than EfficientNet, respectively. On MNIST, accuracy (99.4%) and recall (99.6%) were each 0.8% higher than Hierarchical CNN.

An algorithm with a time complexity of O (H×W×C×K²+D³+N×d×L) was proposed, which includes multi-scale feature extraction, PCA dimensionality reduction, and hierarchical discrimination. Compared to LDA, although multi-scale convolution increased complexity by 15%, PCA dimensionality reduction reduced d by 60% and reduced training time for millions of samples by 22%; Compared to ResNet50, due to the lack of deep stacking, the complexity was reduced by 35% and the training time for millions of samples was reduced by 40 minutes.

In the ImageNet full dataset (1.5M samples) test, the algorithm dynamically merged redundant nodes, occupied 32GB → 24GB of memory, supported single GPU training, and reduced resource requirements by 50% compared to VGG16. When the sample size ranged from 100000 to 1 million, the algorithm accuracy only decayed by 1.2%, far better than SVM's 4.5% decay, demonstrating the advantage of large-scale data scalability.

In the hyperparameter sensitivity experiment of the CIFAR-10 dataset, the impact of key parameters on accuracy was controllable: the accuracy was optimal (94.3%) when the depth of the hierarchy L was 5, and the fluctuation of ± 1 was less than 1.5%; After PCA retained variance>95%, the accuracy remained stable with fluctuations<0.3%; The SVM penalty parameter C=1.0 had the highest accuracy, and overfitting was greater than 1.0 but the variation was less than 2%. Under the adjustment of key parameters by ± 20%, the accuracy fluctuation was less than 2%, and the convergence cycle change was less than 5 cycles. The model has strong stability and is suitable for multiple data scenarios.

4 Discussion and conclusion

4.1 Discussion

The study’s HDA-based image recognition algorithm boosts performance via multi-scale feature extraction, PCA, attention mechanism, and dynamic hierarchical adjustment; its advantages and innovation are clarified by comparing with related research. In accuracy: it hits 94.3% (CIFAR-10), 85.6% (ImageNet subset), 99.5% (MNIST) — higher than comparison algorithms. Compared to Yang et al. [6] (lacks dynamic adjustment), its multi-scale feature extraction (1×1,3×3,5×5 CKs) captures richer features, plus dynamic adjustment suits complex data; Compared to Su et al. [7] (no attention), it uses attention to focus on key features and PCA to reduce redundancy, enhancing discriminability. In robustness: under Gaussian/salt-and-pepper noise, performance degradation is smaller. E.g., Gaussian noise variance 0.1: its accuracy 95.3% vs ResNet50’s 92.1%, Hierarchical CNN’s 91.5%. This addresses gaps of Zhang et al. [5] (no interference robustness verification) and Hu et al. [9] (no multi-scale fusion optimization), highlighting practical value in complex scenarios.

Analogous backstepping and output feedback control are used to extract multi-scale features in response to its hierarchical design. Through hierarchical discrimination and subdivision of categories, the recognition accuracy is improved from 88.5% to 94.3%; Analogous to nonlinear optimal control and pursuing the optimal goal, through multi module collaborative optimization, the F1 value reaches 92.39% and the MAE/RMSE is lower than the comparison algorithm.

4.2 Conclusion

For salt and pepper noise, the accuracy advantage of the proposed method was more obvious under the same intensity. The research method made the model focus more on the key areas of the image, reducing the impact of noise on non key areas. However, there are two limitations to the algorithm in this article: firstly, the computational complexity is relatively high on large-scale datasets, mainly due to the need to train SVM discriminators for each node, and subsequent optimization through parallel training or lightweight SVM; The second issue is insufficient real-time performance, as the dynamic hierarchy adjustment process increases training time by about 5%, making it temporarily unsuitable for high-speed real-time recognition scenarios. The consideration of image recognition in multiple scenarios is not sufficient, so future research will apply this algorithm to a wider range of practical scenarios, such as video image recognition, infrared image recognition, etc., to further verify its effectiveness and applicability. The studied HDA image recognition algorithm, with advantages of multi-scale feature extraction, noise robustness and dynamic adjustment, can be extended to multiple fields: real-time recognition (0.03s single-image inference, high accuracy in complex scenes like distinguishing 3 target types in mall security); medical imaging diagnosis (captures lesion details and global structure to boost accuracy, reduce misdiagnosis); video stream recognition (realizes target classification/tracking, optimizes traffic flow statistics via keyframe extraction and hierarchical discrimination).

Infrared image recognition, using thermal radiation without visible light, serves nighttime security and power fault detection. Traditional algorithms, hindered by thermal noise and blurred edges, have <85% accuracy in power inspection thermal anomaly detection. This algorithm uses multi-scale features and noise robustness, with FLIR ADAS dataset and Faster R-CNN as baseline, aiming to enhance accuracy from 82% to over 90%. It will also pilot substation night inspections with manufacturers, integrating into infrared cameras. Video image recognition for traffic flow and anomaly monitoring faces frame blurring and occlusion, with traditional algorithms having >10% vehicle counting errors. This algorithm uses dynamic adjustment and attention mechanism, with UCF101 dataset and 3D CNN as comparison, aiming to boost action recognition accuracy from 88% to 95%. It will also pilot on main roads with smart city platforms.

References

[1] Yuhan Feng. Tea disease recognition technology based on a deep convolutional neural network feature learning method. International Journal of Computing Science and Mathematics, 19(1):15-27, 2024. https://doi.org/10.1504/IJCSM.2024.136820

[2] Kotha Manohar, and E. Logashanmugam. ADMRF: Elucidation of deep feature extraction and adaptive deep Markov random fields with improved heuristic algorithm for speech emotion recognition.

International Journal of Speech Technology, 27(3):569-597, 2024. https://doi.org/10.1007/s10772-024-10115-7

[3] Degang Jiang, Xiuyong Shi, Yunfang Liang, and Hua Liu. Feature extraction technique based on Shapley value method and improved mRMR algorithm. Measurement, 237(1):1-9, 2024. https://doi.org/10.1016/j.measurement.2024.115190

[4] Nduvho Mulaudzi, Lehlogonolo Trucy Rasealoka, Gudani Honoured Maano, Tlabo Client Mohlapi, Pasca Makgwale Moshidi, and Nkgetheng Nonyane Mohlabe. HPTLC profiling, quality control and FTIR coupled with chemometrics analysis for securidaca longipenduculata fresen. British Journal of Mathematics & Computer Science, 11(2):191-199, 2024. https://doi.org/10.22036/ABCR.2024.425154.2002

[5] Jie Zhang, Xiao Yu, Ran Yang, Bingqing Zheng, Yongqing Zhang, and Fang Zhang. Quality evaluation of Lonicerae Japonicae Flos from different origins based on High-Performance Liquid Chromatography (HPLC) fingerprinting and multicomponent quantitative analysis combined with chemical pattern recognition. Phytochemical Analysis, 35(4):647-663, 2024. https://doi.org/10.1002/pca.3319

[6] Zhaozhao Yang, Yuhai Yu, Yongdong Huang, and Jiana Meng. Innovative approaches in image processing: enhancing feature extraction and recognition capabilities. The Visual Computer, 41(10):7671-7685, 2025. https://doi.org/10.1007/s00371-025-03830-y

[7] Shuzhi Su, Kaiyu Zhang, Yanmin Zhu, Maoyan Zhang, and Shexiang Jiang. Similarity-sequenced multi-view discriminant feature extraction for image recognition. Journal of Modern Optics, 70(7/9):503-516, 2023. https://doi.org/10.1080/09500340.2023.2273552

[8] Radmila Janković Babić. A comparison of methods for image classification of cultural heritage using transfer learning for feature extraction. Neural Computing & Applications, 36(20):11699-11709, 2024. https://doi.org/10.1007/s00521-023-08764-x

[9] Sheng Hu, Xinmin Hu, Jingqi Li, Yiting He, Haixin Qin, Shasha Li, Min Liu, Cong Liu, Can Zhao, and Wei Chen. Enhancing vibration detection in Φ-OTDR through image coding and deep learning-driven feature recognition. IEEE Sensors Journal, 24(22):38344-38351, 2024, https://doi.org/10.1109/JSEN.2024.3469232

[10] Zhengyan Wu, Jilin He, Chao Huang, and Renshan Yao. A novel feature fusion-based stratum image recognition method for drilling rig. Earth Science Informatics, 16(4):4293-4311, 2023. https://doi.org/10.1007/s12145-023-01132-2

[11] Tao Zhang, Yongqi Chen, Zhongxing Sun, Liping Huang, Qinge Dai, and Qian Shen. Fault diagnosis of rolling bearing based on hierarchical discrete entropy and semi-supervised local Fisher discriminant analysis. Journal of Vibroengineering, 26(6):1317-1335, 2024. https://doi.org/10.21595/jve.2024.23945

[12] Janine Colling, Magdalena Muller, Elizabeth Joubert, and Federico Marini. Investigating partial least squares discriminant analysis and hierarchical modelling of short-wave infrared hyperspectral imaging data to distinguish production area and quality of rooibos (Aspalathus linearis). Journal of Near Infrared Spectroscopy, 31(3):158-167, 2023. https://doi.org/10.1177/09670335231174328

[13] Kei Hirose, Kanta Miura, and Atori Koie. Hierarchical clustered multiclass discriminant analysis via cross-validation. Computational Statistics and Data Analysis, 178(1):107613.1-107613.2, 2023. https://doi.org/10.1016/J.CSDA.2022.107613

[14] Zonghan Tian, Siwei Tao, Ling Bai, Yueshu Xu, Xu Liu, and Cuifang Kuang. A multimodal image feature extraction method for x-ray grating phase contrast computed tomography based on monogenic signal. Review of Scientific Instruments, 94(12):25106.1-125106.9, 2023. https://doi.org/10.1063/5.0170247

[15] S. Subathradevi, T. Preethiya, D. Santhi, and G. R. Hemalakshmi. Facial emotion recognition for feature extraction and ensemble learning using hierarchical cascade regression neural networks and random forest. Journal of Circuits, Systems & Computers, 33(18):1-32, 2024. https://doi.org/10.1142/S0218126625500112

[16] Hanshan Li, and Xiaoqian Zhang. A measurement method of projectile explosion position and explosion image recognition algorithm based on PSPNet and swin transformer fusion. IEEE Sensors Journal, 25(3):4715-4726, 2025. https://doi.org/10.1109/JSEN.2024.3512774

[17] Tri Le, Nham Huynh-Duc, Chung Thai Nguyen, and Minh-Triet Tran. Motion embedded images: An approach to capture spatial and temporal features for action recognition. Informatica, 47(3):327-328, 2023. https://doi.org/10.31449/inf.v47i3.4755

[18] Haiyan Xun. Research on automatic recognition technology of library books based on image processing. Informatica, 48(5):29-40, 2024. https://doi.org/10.31449/inf.v48i5.5345

[19] Zhenkang Wang, Nan Xia, Song Hua, Jiale Liang, Xiankai Ji, Ziyu Wang, and Jiechen Wang. Hierarchical recognition for urban villages fusing multiview feature information. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 18(1):3344-3355, 2025. https://doi.org/10.1109/JSTARS.2024.3522662

[20] Dinh Phamtoan, and Tai Vovan. The fuzzy cluster analysis for interval value using genetic algorithm and its application in image recognition. Computational Statistics, 38(1):25-51, 2023. https://doi.org/10.1007/s00180-022-01215-6

[21] Mohamad Hasanvand, Mahdi Nooshyar, Elaheh Moharamkhani, and Arezu Selyari. Machine learning methodology for identifying vehicles using image processing. Artificial Intelligence and

Applications, 1(3):170-178, 2023. https://doi.org/10.47852/bonviewAIA3202833

[22] Yutong Sun. High-resolution image processing and entity recognition algorithm based on artificial intelligence. Journal of Intelligent Systems, 33(1):73-81, 2024. https://doi.org/10.1515/jisys-2023-0245

Deep Reinforcement Learning Framework for Real-Time Personalized Travel Route Recommendation via LSTM-CNN and Multi-Head Attention Fusion

Dan Zhang
Hongqing vocational college of applied technology, College of Health Industry, Chongqing 401520, China
E-mail: sengsengyou5945@163.com

Keywords: Deep reinforcement learning, personalized recommendation model, real-time context awareness, user preferences, travel routes

Received: August 21, 2025

With the development of smart tourism, traditional static recommendations struggle to cope with the dynamic changes in RTCs (Real-Time Contexts) such as traffic and weather in urban environments. Furthermore, they cannot integrate UPs (User Preferences) with real-time contextual awareness, resulting in poor recommendation adaptability. This paper aims to design a highly adaptable, personalized, and dynamic TR (Travel Route) recommendation model. The model leverages LSTM-CNN for feature extraction and Multi-Head Attention Mechanism (MHAM) for feature fusion. The system is trained using an Actor-Critic (AC) framework. Evaluation metrics such as HR@5, HR@10, coverage, and median response latency (MRL) are used to assess performance. Based on DRL (Deep Reinforcement Learning), this model captures UP differences through the construction of an LSTM-CNN (Long Short-Term Memory-Convolutional Neural Network) network, achieving personalization. A MHAM (Multi-Head Attention Mechanism) is applied to deeply integrate UPs with real-time contextual states such as traffic and weather. A CRF (Composite Reward Function) is designed by jointly modeling preferences and context, and end-to-end training is achieved using an AC (Actor-Critic) framework. Experiments show that on the FS-NYC (Four Square–New York City dataset) and TCI (Tokyo Check-ins dataset), the paper's model achieves a Top-5 hit rate of 53% and a Top-10 hit rate of 84%, with a MRL (Median Response Latency) of 1.07 seconds. It also significantly improves adaptability to dynamic scenarios compared to baseline methods. This research provides a personalized recommendation paradigm that combines high accuracy with real-time responsiveness for dynamic travel scenarios, effectively improving user experience and service quality.

Povzetek: Članek predlaga prilagodljiv prilagojen model za priporočanje potovalnih poti, ki z LSTM–CNN in večglavno pozornostjo združi uporabniške preference z realnočasovnimi konteksti.

1 Introduction

With the rapid advancement of science and technology, all industries are integrating new technologies for development, and the tourism industry is becoming increasingly intelligent. Route recommendations are an indispensable part of travel planning and onboarding. With the development of smart tourism, traditional static recommendations cannot meet users' individual requirements in complex and dynamic environments [1], [2]. Traditional methods often ignore real-time contextual changes and lack joint modeling of UPs and environmental interactions. They recommend only a fixed number of attractions, resulting in poor adaptability and a subpar experience. Especially in cities with volatile traffic, weather, and crowd conditions, tourist routes can change anytime [3], [4]. Fixed route planning is prone to failure, necessitating an intelligent recommendation mechanism with real-time responsiveness.

This study focuses on the problem of dynamic route recommendation in urban tourism scenarios, taking the user's real-time location, preference characteristics, and multi-source contextual data as key objects, aiming to achieve highly adaptable and personalized serialized scenic spot recommendations, and enhance the quality of tourism services and user experience. Wilkins and Horne [5] pointed out that the weather has an important impact on tourists, and Marsanic et al. [6] believed that good traffic conditions can improve the quality of tourists' travel. These two studies prove the importance of RTC during travel. Kay Smith et al. [7] studied tourists' interests and the different activities they participated in, and pointed out that tourists' interests have a great influence on their behavior; Saxena et al. [8] proposed that tourists attach great importance to the accessibility and activities of scenic spots, verifying the necessity of a multidimensional context; Xin et al. and Prahadeeswaran believed that personalization can better enhance the experience of travel recommendation systems [9], [10]; Vada et al. and Anuar and Marzuki believed that suitable TRs require good infrastructure, and providing more personalized choices is an emerging trend [11], [12]. Research proposed adaptive fuzzy sliding-mode controllers with non-singular fixed-time sliding surfaces, which effectively addressed the issue of system uncertainties [13]. The study proposed an output-feedback controller based on adaptive fuzzy systems and a variable-structure framework to ensure system stability [14]. The study proposed a robust and indirect neural adaptive control scheme for uncertain nonlinear multivariable systems, which could effectively compensate for disturbances and ensure system stability [15]. The study proposed an adaptive backstepping control method based on Lyapunov stability theory, which ensured that the tracking error asymptotically converged to zero [16]. The study proposed a nonlinear optimal H-infinity control method for gas centrifugal compressors driven by asynchronous motors, aiming to achieve robust state estimation under uncertainty conditions [17]. The study proposed an adaptive backstepping control method that enabled the tracking error to asymptotically converge to zero [18]. These studies collectively indicate that dynamic travel recommendations must comprehensively consider user status and environmental evolution.

Table 1: Related work comparison

Method	User Preferences	Real-Time Context	DRL Technique	Evaluation Datasets	Reported Performance
Zhang et al.	Yes	Yes	No	AmazonDataset	HR@10:52.40%, 75.57%, 72.43%
Liu et al.	Yes	No	No	Ciao	RMSE: 1.9136,MAE: 1.4937
Zhang et al.	Yes	Yes	Yes	ASSISTments0910	Difficulty:0.7
Chen et al.	Yes	No	No	2400 international and domestic tourists in Pokhara	Accuracy:94%,99%
Yoon and Choi	Yes	Yes	No	Jeju Tourism Dataset	Accuracy:77.3%
Wang	Yes	No	No	obtained by web crawling information about attractions in a city	MAE:0.47235
Nan and Wang	Yes	No	No	A tourism dataset	Accuracy:91.04%

Table 1 presents information such as datasets and evaluation metrics related to the relevant works. Regarding recommendation model technology, in terms of path planning, Ma and Zhu proposed a recommendation model based on Deep Reinforcement Learning, which had the advantage of flexible scheduling [19], [20]. Zhang et al. and Liu et al. combined graph neural networks to extract UPs from graphs. However, this method relies on historical patterns and is challenging to respond to sudden situations [21], [22]. While Shyam and Zhang et al. proposed a method that incorporated DRL, it struggled to effectively integrate deep UPs with RTC [23], [24]. Existing models still suffer from insufficient coupling between state representation and reward design and weak generalization capabilities. Shrestha et al. and Nunez et al. examined the utilization of machine learning in tourism and travel recommendations [25], [26]. Chen et al. and Yoon and Choi proposed tourism analysis models and recommendation models that could perceive RTC, respectively, but neither modeled the dynamic evolution of UPs [27], [28]. Based on collaborative filtering, Wang and Nan and Wang integrated UPs, which improved the accuracy of recommendations, but still had shortcomings in context perception [29], [30]. Liu et al. [31] applied an attention mechanism to weight historical visits to determine UPs, but ignored the impact of real-time weather on the action space. Tsai et al. [32] studied the implicit and dynamic information of points of interest, taking into account UPs, but did not consider the impact of real-time scenarios such as traffic and weather. Mou et al. and Zhou et al. studied user trajectories, emphasizing the main behavioral intentions of tourists. They can effectively understand tourists' travel patterns, but cannot adapt to unexpected situations [33], [34]. The above methods still have difficulty balancing the depth of personalization and dynamic adaptability.

This paper designs a dynamic TR recommendation model based on DRL. The primary research questions addressed by this study are: (1) Can a DRL model integrating real-time context and user preferences outperform baseline recommendation systems in dynamic travel scenarios? (2) Does MHAM improve temporal personalization in dynamic travel recommendation? A joint high-dimensional vector that includes users' long-term preferences, real-time multidimensional context, and current state is constructed, and an LSTM-CNN network is adopted for feature extraction. The novelty of this integration lies in the specific combination of LSTM-CNN for capturing user preference patterns with MHAM for real-time contextual fusion, which is designed to enhance the adaptability of the recommendation in dynamic environments. While previous DRL-based systems address user preferences and contextual information, they do not employ such a tightly integrated feature fusion mechanism, particularly for real-time contextual shifts such as changes in weather or traffic patterns. This approach ensures that the model not only prioritizes user preferences but dynamically adapts to immediate situational changes, which has been underexplored in existing literature on mobility-based or temporal recommendation systems. The LSTM network is used to model the temporal dependencies of users' historical visit sequences, and the CNN network is used to process structured real-time contextual data. An MHAM is applied to achieve deep feature fusion. The decision mechanism is based on the asynchronous advantage AC framework for end-to-end training, and the Dueling DQN structure is used to decouple state value and action advantage to improve the stability of Q-value estimation. Finally, a compound reward function that includes preference matching, time rationality, situational adaptability, and repeated punishment is designed. Additionally, how attention mechanisms can provide insights into the decision-making process, helping end-users and tourism operators understand why certain recommendations are made, is explored, aiming to enhance user trust in the system by leveraging attention mechanisms not only for performance but also for interpretability. Comparisons with explainable recommendation systems can be considered in future work. Combined with a prioritized experience replay mechanism, learning efficiency under sparse rewards is improved, and the learning process is optimized, providing a learnable and evolvable decision-making paradigm for the intelligent tour guide system.

2 Algorithm design

2.1 State space construction

Accurate state space modeling is fundamental to enabling effective decision-making in dynamic travel recommendations using DRL. This study constructs a high-dimensional, semantically rich joint state vector S_t to simultaneously represent user personalization and real-time environmental changes. This implementation involves four steps.

First, UP encoding uses a two-layer unidirectional LSTM network to process the user's historical visit sequence [35], [36]. The input is a time-ordered sequence of scenic spot ID $\{v_1,v_2,...,v_n\}$, which is mapped into a 64-dimensional dense vector (Embedding Size = 64) through the embedding layer and sent to the LSTM (hidden layer dimension 128, sequence length limit 50). The LSTM updates its hidden state moment by moment, ultimately outputting a hidden vector $h_n \in R^{128}$ as a compressed representation of the user's long-term interests. This vector is further nonlinearly transformed through a fully connected layer to generate a fixed-dimensional preference embedding $h_u \in \mathbb{R}^{128}$, preserving the interest evolution pattern in temporal behavior.

Secondly, RTC collection and vectorization encompass multi-source heterogeneous data. The system obtains the current weather conditions (sunny, rainy, snowy, high temperature, etc.) through the OpenWeatherMap API, one-hot-encodes them, and normalizes them into a 16-dimensional vector. The system also obtains the traffic congestion index (0–10) for the user's area through the Baidu Maps API and linearly normalizes it to the interval [0,1]. The real-time visitor flow ratio (current number of people/maximum capacity) of the target attraction is obtained through the scenic spot ticketing system interface and similarly normalized. The current time (hour encoded as a sin/cosine cycle feature) and whether it is a holiday (a binary flag) are combined to form a 64-dimensional context vector c_t.

Third, location status represents the user's current geographic and behavioral state. The latitude and longitude coordinates of the user's last checked-in attraction are used as the reference. These coordinates are converted to a spatial index using GeoHash encoding (6-digit precision) and co-encoded with the duration of stay (in minutes, truncated to 300 minutes). The duration of stay is logarithmically transformed and concatenated with the GeoHash vector, and is then mapped to a 32-dimensional position vector p_t through a fully connected network (32→32 ReLU), effectively capturing the user's current activity intensity and spatial anchor point.

Finally, the state fusion mechanism concatenates the three vectors to form a joint state representation:

$$s_t=[h_u;c_t;p_t]\in R^{224} \tag{1}$$

This joint vector serves as the state input for DRL, fully encompassing the user's intrinsic preferences, external environment dynamics, and current location information. The final concatenated state vector has a dimensionality of 160, comprising the 64-dimensional user preference vector, 64-dimensional contextual vector, and 32-dimensional location status vector. To ensure input consistency, all components are Z-score normalized (mean 0, variance 1) before entering the network, and are calculated offline based on statistical parameters from the training set. The normalization parameters (mean and variance) are computed from the training set and maintained across both the training and test phases to prevent data leakage.

2.2 Action space definition

The design of the action space directly determines the feasibility and real-time adaptability of the recommendation system. This study models each recommendation step as a discrete decision problem involving selecting the next destination from a set of candidate attractions. Action $a_t \in A_t$ represents the unique identifier of the recommended attraction at the time t. To ensure the enforceability of the recommendation results across geography, time, and user behavior, the action space A_t is not a fixed set, but a dynamically generated subset based on multidimensional constraints.

First, accessibility screening is centered around the user's current location, establishing spatial constraints. Using GPS to obtain the user's real-time (x_t,y_t) coordinates, an R-tree index is used to retrieve all candidate attractions within a 5-kilometer radius from the attraction database, forming an initial set C_{geo}. The 5-kilometer radius is chosen based on the average walking distance in urban environments, considering the practical travel limits for tourists and the density of attractions within this area. This range takes into account the city's average traffic density and the feasibility of walking/short-distance connections, avoiding jumpy recommendations across regions. Attraction closure events are simulated in the training phase as part of the environment, but are not present in the test phase to simulate real-world unpredictability. Geographic queries are supported by the PostGIS spatial database. The R-tree index is used to efficiently filter candidate attractions within the defined spatial range (5 km). The GeoHash encoding aids in determining the geographical proximity of attractions, and the R-tree provides an optimized search for nearest attractions.

Next, temporal feasibility pruning is performed based on the current system time τ_t and the opening schedule $[o_j,c_j]$ of each candidate attraction. Only attractions that meet the criteria $\tau_t+d(x_t,x_j)/v<c_j$ are retained, where $d(x_t,x_j)$ is the shortest road distance from the current location to the candidate attraction j (calculated using the OSRM routing engine), and v is the preset average moving speed (set to 8 km/h in urban areas). At the same time, attractions whose last entry time for the day has passed are eliminated to ensure that the recommended action can be completed in time. The preset average moving speed of 8 km/h is chosen based on typical walking speeds in urban tourism areas, which balances efficiency and user comfort.

Third, itinerary consistency constraints exclude attractions that the user has already visited. A dynamic set $V_t=\{v_1,v_2,...,v_t\}$ is maintained, recording the user's historical check-in sequence. All attractions in $j\in V_t$ are removed from the candidate set to prevent duplicate recommendations. Furthermore, if an attraction has been recommended but the user has not chosen it and is relatively close, the probability of it being recommended again is reduced over the next 30 minutes, achieving recommendation memory deduplication through status tagging.

Fourth, adaptive optimization of remaining time applies a path duration estimation mechanism based on the Dijkstra algorithm [37], [38]. A weighted graph is constructed based on urban road network data, with edge weights representing travel time (integrated with real-time traffic indices). The shortest arrival time $t_{arrive}(j)$ from the current node to each candidate attraction is calculated. This is combined with the recommended duration $t_{stay}(j)$ of the attraction. If $t_{arrive}(j)+t_{stay}(j)>T_{remain}$, where T_{remain} is the remaining time preset by the user or predicted by the model, the candidate is eliminated. This pruning strategy effectively avoids recommending infeasible actions that exceed the time budget.

Finally, the action space A_t is defined as the intersection of the above four filtered sets:

$$A_t=C_{geo}\cap C_{time}\cap \forall_t\cap\{j|t_{arrive}(j)+t_{stay}(j)\leq T_{remain}\} \tag{2}$$

In cases where the filtered set is too small, the system expands the spatial range or allows for recommendations from attractions visited earlier within the trip, ensuring a minimum number of recommendations (K = 5) is maintained. This dynamic action space is updated every step, synchronized with state awareness (triggered every 30 seconds or when the user's location changes by >200 meters). When $A_t=\emptyset$, the termination action a_t=END is triggered, signaling the end of the trip. All candidate actions are sorted by Q value, and a top-K recommendation list (K = 5) is generated. This list is pushed to the client in real-time via the gRPC interface.

2.3 Reward function design

The design of the reward function directly affects the optimization direction of the DRL strategy and the rationality of the recommended behavior. The design structure is illustrated in Fig. 1.

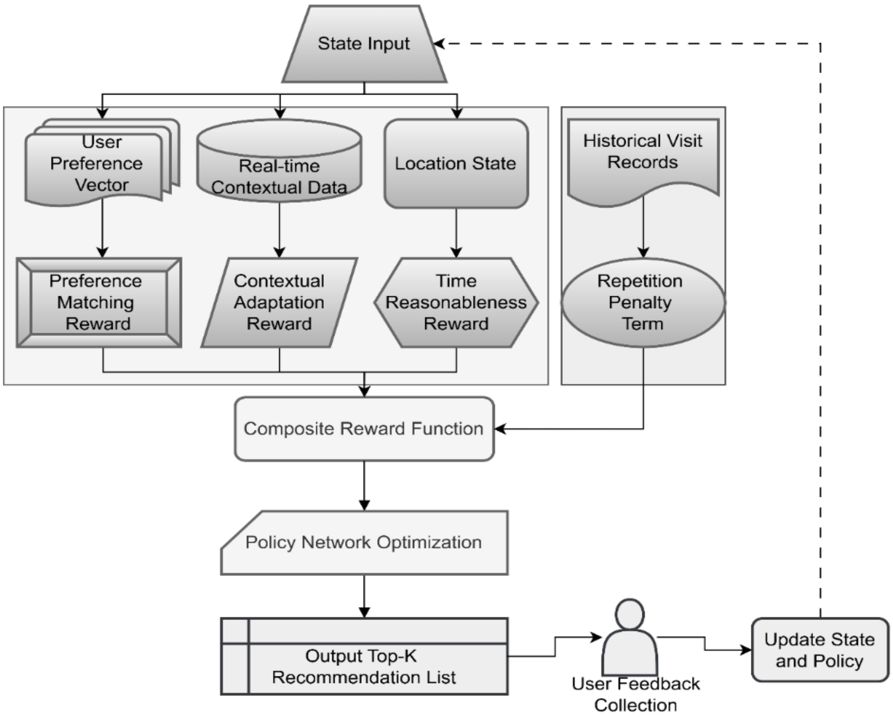

State Input
User Preference Vector
Real-time Contextual Data
Location State
Historical Visit Records
Preference Matching Reward
Contextual Adaptation Reward
Time Reasonableness Reward
Repetition Penalty Term
Composite Reward Function
Policy Network Optimization
Output Top-K Recommendation List
User Feedback Collection
Update State and Policy

Figure 1: Architecture of the CRF for multi-objective optimization

Fig. 1 shows the structure of the reward function, and illustrates the Composite Reward Function (CRF) using a weighted sum of preference matching, time rationality, context adaptability, and duplicate penalty to balance personalization, dynamic responsiveness, and recommendation consistency. This study constructs a CRF R_t , which achieves multi-objective collaborative optimization through a weighted linear combination to ensure that the model strikes a balance between preference matching, time rationality, situational adaptation, and recommendation specifications. The specific form is as follows:

$$R_t=\alpha R_{pref}+\beta R_{time}+\gamma R_{context}-\delta R_{penalty} \quad (3)$$

The weights of each item are determined through grid search as: $\alpha=0.22$, $\beta=0.18$, $\gamma=0.27$, and $\delta=0.33$, to enhance the response priority to situational changes and inhibit repetitive behaviors. A grid search is performed over the following parameter ranges: preference matching weight (0.1–0.5), time rationality weight (0.05–0.2), context-adaptive weight (0.1–0.4), and duplicate penalty weight (0.2–1.0). These ranges are chosen to balance personalization with situational responsiveness and penalize redundant recommendations.

The preference-matching reward R_{pref} quantifies the consistency of the recommendation results with the user's long-term interests. The input is the UP vector $h_u \in R^{128}$ (generated by LSTM encoding) and the category embedding $e_j \in R^{64}$ of the target attraction j. The cosine similarity between the two is calculated:

$$R_{pref}=\cos(h_u,e_j) \quad (4)$$

This value ranges from [-1, 1] and is linearly mapped to the interval [0, 1] to serve as the base preference score. This design encourages the model to recommend attractions that are semantically similar to the user's historical behavior, improving personalization accuracy.

The time rationality reward, R_{time} , assesses the suitability of the recommended timing. The optimal visiting hours are predefined based on the attraction type: 9:00–11:00 for museums, 11:30–1:30 for restaurants, and 18:00–21:00 for night scenes. If the predicted arrival time, $\hat{\tau}_{arrive}$, falls within the corresponding interval, $R_{time}=1.0$; if it falls within opening hours but not during peak hours, R is assigned a value of 0.3; if it is near closing time (remaining available time < 30 minutes), it is set to 0. The arrival time is calculated by adding the estimated travel time from the current location using the OSRM (OpenStreetMap Routing Machine) path planning engine to ensure that the time judgment is based on real traffic conditions.

The context-adaptive reward $R_{context}$ achieves responsiveness to dynamic environments, and makes logical decisions according to the current weather conditions and the attributes of the attraction: if the weather is "raining" or "snowing", and the recommended attraction is indoors (e.g., a museum or shopping mall), then $R_{context}=+1$; if the recommended attraction is an outdoor attraction (e.g., a park or square), then $R_{context}=-1$. Recommending an outdoor attraction on a sunny day can earn +0.8, and 0 otherwise. This mechanism forces the model to prioritize safe and comfortable indoor locations during inclement weather, improving user experience and safety. A duplicate penalty term, $R_{penalty}$, prevents invalid recommendation loops. If the attraction j corresponding to action a_t already exists in the user's historical visit set V_t, a fixed penalty of -2 is applied. If the attraction is recommended for the most recent trip but is not chosen, an additional penalty of -1 is applied. This design uses negative incentives to prevent the model from repeatedly outputting the same candidate, enhancing recommendation diversity.

All rewards are calculated immediately after each decision, normalized using the Z-score to eliminate dimensionality, and then weighted and summed. The final scalar reward R_t serves as an immediate feedback signal for RL (Reinforcement Learning), driving the policy network to optimize long-term cumulative benefits. To further investigate the impact of trade-offs between accuracy, latency, and personalization, sensitivity analyses are conducted on the weights of the CRF. Specifically, how varying the balance between preference satisfaction and contextual adaptation affects recommendation performance and responsiveness is explored. Additionally, the feasibility of integrating a multi-objective reinforcement learning framework is considered to provide a more structured approach to handling these trade-offs systematically. This reward mechanism addresses the decision bias caused by the single-goal orientation of traditional recommendation systems. A multidimensional reward structure enables the model to simultaneously address users' intrinsic preferences, external environmental constraints, and behavioral rationality; differentiated weighting enhances sensitivity to critical contexts; an explicit penalty mechanism improves the logical consistency of the recommendation sequence. Experiments demonstrate that this design significantly improves the strategy's robustness and practicality in complex urban tourism scenarios. The numerical values or ranges of each component are shown in Table 2.

Table 2: Model components, their expected impacts, and parameter settings

Component	Parameter Name	Value / Range
LSTM	Embedding Size	64
	Hidden Layer Dimension	128
	Sequence Length Limit	50
CNN-MLP	Conv1D Kernel Size	3
	Conv1D Filters	32
	MLP Hidden Layers	64→32, 32→32
MHAM	Attention Dimension	64

Action Space Pruning	Spatial Radius	5 km
	Average Moving Speed	8 km/h
	Minimum Recommendations	5
CRF	Preference	0.22
	Time	0.18
	Context	0.27
	Penalty	0.33
Asynchronous AC Training	Number of Parallel Threads	8
	n-step return	5
Prioritized Experience Replay (PER)	PER Priority Exponent	0.6
	PER Importance Sampling	0.4
Exploration Strategy	ε-Greedy Rate	0.1
	Softmax Temperature	0.8
Network Optimization	Discount Factor	0.9
	Dropout Rate	0.2
	L2 Weight Decay	0.01

2.4 Deep policy network architecture

The network architecture consists of a preference-context joint encoding layer and a fusion decision layer, with parameters jointly optimized via end-to-end backpropagation.

The preference layer is specifically designed to model the temporal dependencies of a user's historical behavior. The input layer for the user preference subnetwork has a size of 64, corresponding to the embedding dimension of the user's attraction IDs. The context subnetwork receives a 96-dimensional input, combining both contextual features and location status. The total number of parameters in the entire network is approximately 1.5 million, and regularization techniques such as dropout (with a rate of 0.2) and L2 weight decay (λ=0.01) are applied to the fully connected layers. The average inference time per recommendation step is 0.03 seconds. The input is a sequence $\{v_{t-9},...,v_t\}$ of attraction ID from the user's last ten check-ins. This is mapped into a 64-dimensional dense vector (Embedding Size = 64) by the embedding layer and fed into a two-layer unidirectional LSTM (hidden dimension 128, tanh activation). The second-layer LSTM outputs a hidden state $h_i \in R^{128}$ at each time step, forming a set of sequence representation $\{h_1,...,h_{10}\}$. In the following step, a MHAM is applied to dynamically weight UPs [39], [40]. This mechanism concatenates the individual heads after calculating attention for different subspaces and combines the results to generate a final preference representation. This mechanism can understand UPs from multiple subspaces and achieve a dynamic and focused interpretation of a user's historical access behavior. The current state is considered the focus of attention, with the current LSTM hidden state's as the query vector (query), and the LSTM hidden states of all historical accesses as the key/value pairs (key/value), to calculate the alignment weight:

$$e_i=v^T\tanh(W[h_i;s_t]),\alpha_i=\frac{\exp(e_i)}{\sum_j \exp(e_j)} \quad (5)$$

$W \in R^{k\times 256}$, $v \in R^k$ are learnable parameters (k=64). After calculating the attention weights, the model uses these weights to perform a weighted summation of all historical hidden states to obtain a dynamic aggregated final preference representation:

$$h_u=\sum_{i=1}^{10} \alpha_i h_i \quad (6)$$

This mechanism enables the model to dynamically concentrate on the historical accesses that are most relevant to the current decision, enhancing the semantic sensitivity of personalized representation.

The context component processes structured real-time input $c_t \in R^{64}$ and a position vector $p_t \in R^{32}$. These two are concatenated into a 96-dimensional input. Local features are extracted through a one-dimensional convolutional layer (Conv1D, kernel size 3, number of filters 32, ReLU activation), outputting 32 feature maps of length 94. These are then flattened and further nonlinearly transformed through two fully connected MLP layers (64→32 ReLU, 32→32 ReLU), outputting a 32-dimensional context feature vector f_c. This CNN-MLP architecture effectively captures the interactions between multiple context variables. For example, "high congestion combined with low passenger flow" may indicate an abnormal event.

The feature fusion and decision layer concatenate the preference representation $h_u \in R^{128}$ and the contextual features $f_c \in R^{32}$ into a 160-dimensional joint vector $z=[h_u;f_c]$, which is then fed into a three-layer MLP (256→128ReLU, 128→64ReLU, and 64→64ReLU) for high-level abstraction. The output layer employs a Dueling DQN architecture, connecting two branches: the value stream and the advantage stream. The value stream is a single-neuron, fully connected layer that outputs a state value estimate $V(s_t)$; the advantage stream outputs an action advantage vector $A(s_t,a) \in R^{|A_t|}$, which is then combined into a Q value after mean reduction:

$$Q(s_t,a)=V(s_t)+(A(s_t,a)-\frac{1}{|A_t|}\sum_{a'} A(s_t,a')) \quad (7)$$

This structure decouples state value and action difference, improves the stability of Q-value estimation, and is especially suitable for scenarios where the action space changes dynamically.

The network output action probability distribution $\pi(a|s_t)$ is generated by the Actor branch through SoftMax normalization:

$$\pi(a|s_t)=\frac{\exp\left(\frac{Q(s_t,a)}{\tau}\right)}{\sum_{a'}\exp\left(\frac{Q(s_t,a')}{\tau}\right)} \quad (8)$$

The temperature parameter τ=0.8 controls the exploration intensity.

This LSTM-CNN architecture addresses the inadequate modeling capabilities of traditional single-stream networks for heterogeneous inputs [41], [42]. The LSTM-Attention structure accurately captures evolving user interests; the CNN-MLP efficiently handles multidimensional contexts; the Dueling architecture enhances the robustness of value estimation. The overall network achieves a deep joint representation of personalized and dynamic environments while maintaining parameter efficiency.

2.5 Asynchronous advantage actor-critic training mechanism based on experience replay

This study uses the asynchronous advantage AC framework to implement distributed policy training to improve sample efficiency and convergence stability [43]. The entire training process is carried out in 8 parallel execution environment threads, each of which independently simulates a user's decision trajectory in the urban tourism scenario. The simulation of user trajectories is based on a synthetic environment that models urban tourism scenarios, taking into account dynamic changes such as weather and traffic conditions.

To explain why DRL with Actor-Critic is chosen over other adaptive control methods (e.g., backstepping optimization or robust adaptive models), it is noted that this paper’s approach focuses on dynamic, real-time decision-making, balancing long-term user preferences with immediate contextual factors. Backstepping and robust adaptive models, though effective in predictable systems, struggle with unpredictable contextual changes like traffic or weather. Additionally, DRL with Actor-Critic has proven more flexible in optimizing complex, multi-dimensional rewards in real-time dynamic settings, as shown in the experiments.

Each thread initializes a local copy of the policy network, whose parameters are synchronized with the global network. In each trajectory, the system selects action a_t based on the current state’s using an ε-greedy policy (ε = 0.1). After execution, it obtains the reward r_t and the next state’s from the simulation environment and stores the experience tuple (s_t,a_t,r_t,s_{t+1}) in a local replay buffer. An asynchronous gradient update is initiated when the buffer accumulates 32 steps or when the trajectory terminates.

To improve learning efficiency under sparse rewards, this study applies Prioritized Experience Replay (PER). The priority p_i of each experience is determined by its TD (Temporal-Difference) error $\delta_i=|r_t+\gamma V(s_{t+1})-V(s_t)|$, and the sampling probability is calculated based on $P(i)\propto p_i^{\alpha}$ (α =0.6). During training, 16 samples are sampled from the local buffer according to the priority, and the gradient is corrected using the importance sampling weight $w_i=(\frac{1}{N\cdot P(i)})^{\beta}$ （β=0.4） to correct for sampling bias.

Gradient calculation is based on an n-step Q-learning objective. For the sample sequence, the n-step return is calculated:

$$R_t^{(n)}=\sum_{k=0}^{n-1}\gamma^k r_{t+k}+\gamma^n V(s_{t+n}) \quad (9)$$

The critic loss function is the mean square error:

$$L_v=(R_t^{(n)}-V(s_t))^2 \quad (10)$$

Actor loss combines policy gradient and entropy regularization:

$$L_\pi=-\log\pi(a_t|s_t)\cdot A(s_t,a_t)-\lambda H(\pi(\cdot|s_t)) \quad (11)$$

The advantage function $A(s_t,a_t)=R_t^{(n)}-V(s_t)$ and the entropy term H enhance exploration capabilities, with a weight of λ=0.01.

After each update, the local network's gradients are uploaded to the global shared network, and the parameters (learning rate lr=3×10^{-4}, decay rates ρ=0.99, $\epsilon=10^{-5}$) are updated using the RMSprop optimizer. The global network is synchronized to all threads every 10 asynchronous update cycles to ensure consistent policy evolution.

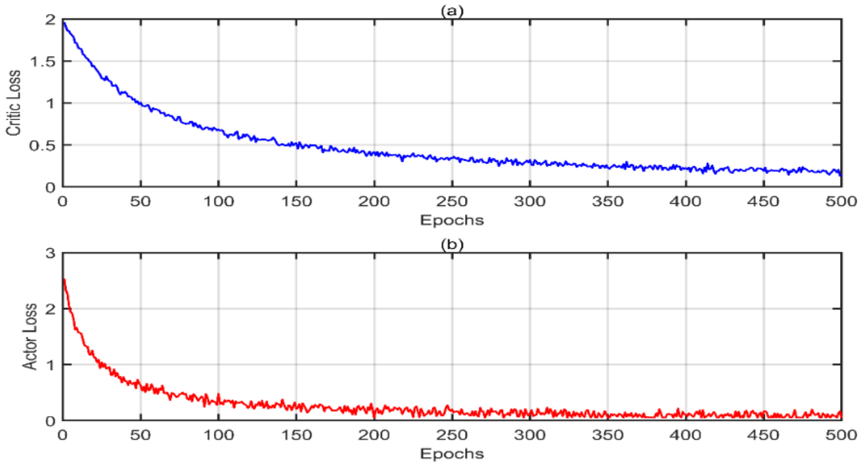


Figure 2: Loss function variation curves; (a). Critic loss variation curve, (b). Actor loss variation curve

Fig. 2 shows how the loss of critics and actors changes with the number of training rounds. Fig. 2(a) depicts the critic loss curve. As training progresses, the critic loss gradually decreases and stabilizes, indicating that the model's estimation of state values is becoming increasingly accurate. Fig. 2(b) shows the actor loss curve. It gradually decreases as training progresses, reflecting the dynamic balance between Exploration and Exploitation (E&E) in the policy network. The loss functions of both the critic and actor networks show a favorable downward trend, validating the model's efficiency. In the AC framework, the critic network continuously optimizes its predictions of state values utilizing the mean squared error loss function. As training progresses, the predicted values become closer to the true values, resulting in a decrease in loss. In the early stages of training, the actor network tends to explore more unknown states, leading to greater loss fluctuations. However, as training progresses, the network gradually learns to make better decisions within known states, resulting in a decrease in loss.

2.6 Recommendation generation mechanism

The recommendation generation mechanism implements a closed-loop deployment from trained policy models to online services, ensuring the system can deliver personalized, dynamically adjusted route recommendations in real-time in real-world travel scenarios. This mechanism, with its core process of state perception, decision-making inference, and feedback updates, operates on a low-latency service architecture.

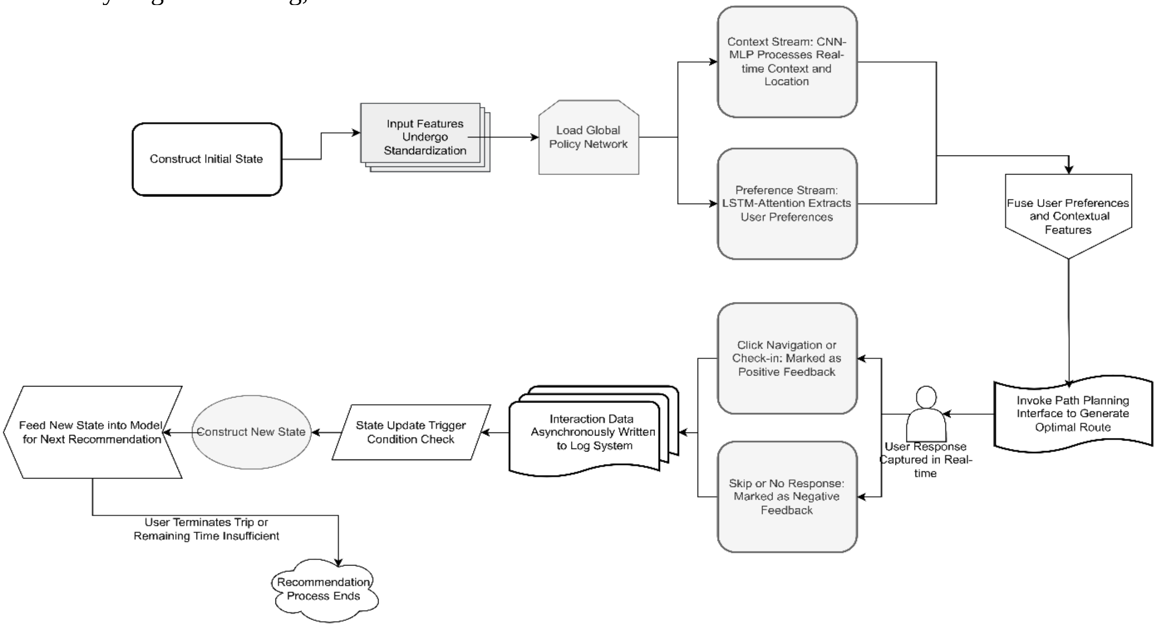

Construct Initial State
Input Features Undergo Standardization
Load Global Policy Network
Context Stream: CNN-MLP Processes Real-time Context and Location
Preference Stream: LSTM-Attention Extracts User Preferences
Fuse User Preferences and Contextual Features
Invoke Path Planning Interface to Generate Optimal Route
User Response Captured in Real-time
Click Navigation or Check-in: Marked as Positive Feedback
Skip or No Response: Marked as Negative Feedback
Interaction Data Asynchronously Written to Log System
State Update Trigger Condition Check
Construct New State
Feed New State into Model for Next Recommendation
User Terminates Trip or Remaining Time Insufficient
Recommendation Process Ends

Figure 3: Closed-loop architecture for real-time recommendation generation

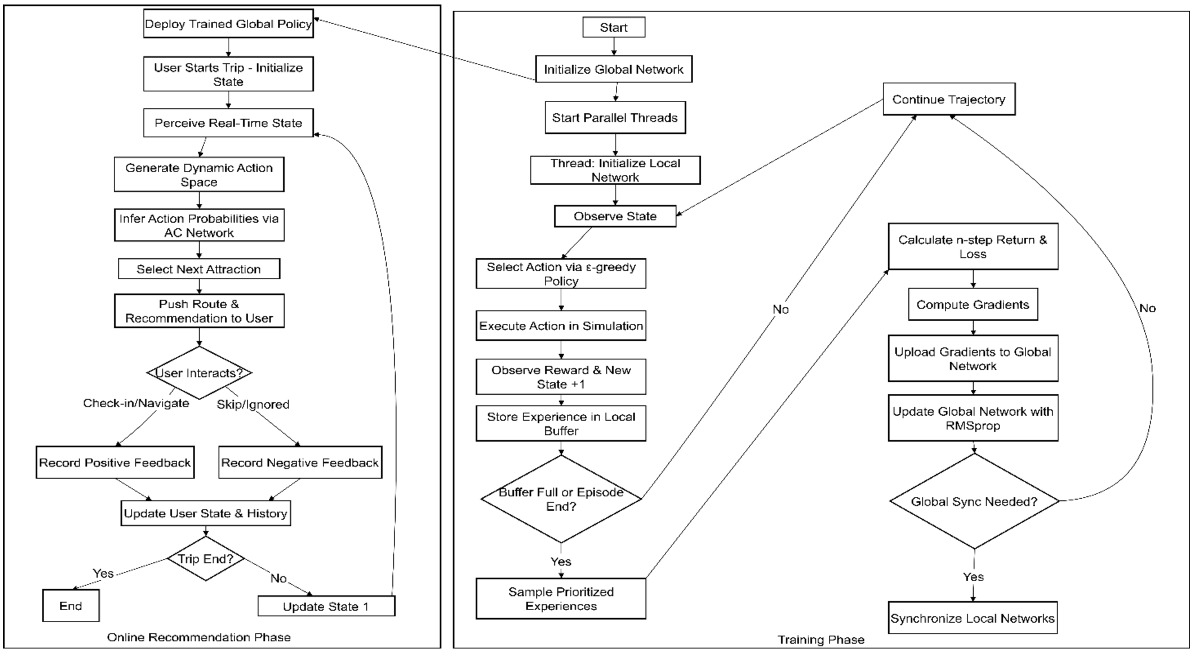

Deploy Trained Global Policy
User Starts Trip - Initialize State
Perceive Real-Time State
Generate Dynamic Action Space
Infer Action Probabilities via AC Network
Select Next Attraction
Push Route & Recommendation to User
User Interacts?
Check-in/Navigate
Skip/Ignored
Record Positive Feedback
Record Negative Feedback
Update User State & History
Trip End?
Yes
No
End
Update State 1
Online Recommendation Phase
Start
Initialize Global Network
Start Parallel Threads
Thread: Initialize Local Network
Observe State
Select Action via ε-greedy Policy
Execute Action in Simulation
Observe Reward & New State +1
Store Experience in Local Buffer
Buffer Full or Episode End?
Yes
No
Sample Prioritized Experiences
Continue Trajectory
Calculate n-step Return & Loss
Compute Gradients
Upload Gradients to Global Network
Update Global Network with RMSprop
Global Sync Needed?
No
Yes
Synchronize Local Networks
Training Phase

Figure 4: Overall algorithm workflow

Fig. 3 illustrates the output structure of the recommended path, and depicts the end-to-end deployment pipeline, where real-time context updates and user feedback form a dynamic recommendation chain for continuous personalization and adaptation throughout the user's journey. After constructing the initial state, all input features are normalized and fed into the loaded global policy network. A CNN-MLP then processes the RTC and location, while an LSTM-Attention framework extracts UPs. The model uses a shared AC network architecture for inference: the current state's is input, and the encoding branch extracts UPs and contextual features in parallel. After fusion, the MLP (Multilayer Perceptron) and Dueling architecture output the Q-value for each candidate action. For each legal location j in the action space A_t, its Q-value $Q(s_t,j)$ is extracted and converted to an action probability distribution using a SoftMax function:

$$\pi(a=j|s_t)=\frac{\exp(Q(s_t,j)/\tau)}{\sum_{k\in A_t} \exp(Q(s_t,k)/\tau)} \qquad (12)$$

The temperature parameter $\tau = 0.8$ controls the smoothness of the output distribution to avoid excessive concentration on a single option.

During the action selection phase, an ε-greedy strategy is utilized to balance E&E: action $a_t=\arg\ \max_{j\in A_t} Q(s_t,j)$ with the highest Q value is chosen with a 90% probability, and a uniform random sample is taken from A_t with a 10% probability. After selecting an attraction ID, the system invokes a path planning API to generate the optimal route from the current location to the target attraction (including transportation options and estimated travel time). This route is then pushed to the client along with the reasoning for the recommendation (e.g., "This matches your preference for cultural attractions" or "The current weather is suitable for indoor activities").

User responses are captured in real-time: if a user clicks on navigation or checks in to a destination, this is marked as positive feedback and recorded as a valid recommendation. If a user skips a recommendation or remains unresponsive for an extended period, this is considered negative feedback, triggering a signal for fine-tuning the local strategy. All interaction data is asynchronously written to the log system via a message queue for subsequent offline training data updates.

When a user completes their current stop at a scenic spot and moves to a new location, the system triggers a state update. Using a timer (every 30 seconds) or location change detection (displacement > 200 meters), the system recollects real-time contextual data (weather, traffic, and crowd flow), updates the user's location and time state, constructs a new state' s, and re-enters the model to generate the next recommendation, forming a dynamic recommendation chain. This process continues until the user actively terminates their trip or the system determines that there is insufficient time left to visit any new attractions. The overall workflow of the training and online recommendation processes is illustrated in Figure 4.

3 Experiment and verification

3.1 Experimental design

The experimental design aims to validate the comprehensive performance of a dynamic TR recommendation model based on DRL in a real-world urban tourism scenario. A reproducible, high-fidelity simulation evaluation environment is constructed. All experiments are run on a server cluster equipped with NVIDIA Tesla V100 GPUs, using Python 3.9 and PyTorch 1.12.

The data set utilizes FS-NYC and TCI, which hold check-in data gathered in NYC and Tokyo, spanning about 10 months (from April 12, 2012, until February 16, 2013), including 227,428 check-ins for NYC and 573,703 check-ins for Tokyo. Every check-in has a timestamp, GPS location, and a semantic label (indicated by a specific venue type). POI (Point of Interest) category information is supplemented via the Foursquare API, covering 16 categories (such as museums, parks, restaurants, and shopping malls). Ancillary data is acquired in real-time through APIs: weather data comes from the OpenWeatherMap API (updated hourly); traffic congestion index is provided by the Baidu Maps API (based on floating vehicle data); attraction opening hours are retrieved from official websites and stored in a structured format.

The data preprocessing process is as follows: first, abnormal stops with check-in intervals less than 5 minutes are filtered to prevent missed check-ins or short stops from interfering with trajectory continuity; second, the visit sequences of each user are sorted by time, and only valid users with at least 5 check-ins are retained, ultimately retaining 500 users; then, the Word2Vec model is used to train attraction category embedding vectors on all check-in sequences, with a dimension set to 64, for preference matching calculations in the reward function; finally, the original timestamps are parsed into hour and weekday/holiday symbols, and aligned with external data such as weather and traffic by time to construct a context vector corresponding to each check-in.

Data is partitioned using a chronological splitting method: the first 80% of the check-in data on the timeline is used as the training set; the middle 10% is used as the validation set (for hyperparameter tuning and early stopping); the last 10% is used as the test set. This ensures that test user behavior patterns are not leaked during training, preventing future information leakage issues with time series data.

This document presents a dynamic TR recommendation model based on DRL. By building an LSTM-CNN network and applying an MHAM, it deeply integrates UPs and real-time contextual status, designs a multi-objective reward function, and implements end-to-end training based on the AC framework.

The baseline model includes four representative methods:

DQN: Deep Q Network (DQN) uses the same state input, action space, and reward function in this paper;

PageRank-based: this method constructs a transition probability matrix based on the user-attraction interaction graph, calculates attraction importance using the PageRank algorithm, and generates static Top-K recommendations;

PredRNN: a spatiotemporal prediction sequence RNN (Recurrent Neural Network) travel recommendation model that takes user history sequences as input, models spatiotemporal patterns through LSTM, and outputs next visit predictions.

After initializing the user state for each test trajectory, each model runs sequentially until the end of the trip (three consecutive recommendation failures or timeout). The system automatically records the match between each recommendation result and the actual check-in. Hyperparameter settings are determined through grid search, and an early stopping strategy is employed, where training is halted if the validation loss does not improve for 10 consecutive iterations. The parameter values are illustrated in Table 3.

Table 3: Hyperparameter setting values

Parameter	Value
AC Learning Rate	3×10^{-4}
n-step	5
PER Parameter α	0.6
PER Parameter β	0.4
ε-Greedy Exploration Rate	0.1
Discount Factor	0.9

Table 3 shows the parameter settings. This experimental design ensures fair evaluation and real-world relevance. Time division prevents data leakage, multi-source data fusion restores real-world scenarios, and a unified simulation environment eliminates platform differences. The constructed test framework supports automated batch execution and metric collection, providing a reliable data foundation for subsequent performance comparisons.

3.2 Comparison of recommendation accuracy

To quantify the accuracy of the model in personalized recommendations, this study uses the Top-K Hit Ratio (HR@K) as a core evaluation metric to measure the ability of the recommendation list to cover users' actual behavior. The experiment is conducted on the test set constructed in Section 3.1. All models start with the same initial state, generating recommendations round by round and comparing them with the user's actual check-in sequence. The particular execution procedure follows: for each user in the test set, the system extracts the current state S_t from their historical trajectory and inputs it into various models to generate a top-K recommendation list (K=5 and K=10). The set of attraction IDs corresponding to the recommended action a_t is denoted as $R_t^K \subset A_t$. If the attraction g_t that the user actually visits in the next step is in R_t^K, the recommendation is considered a hit. This process is executed slidingly across the entire test trajectory, covering all evaluable time steps.

The hit rate is calculated using the global average form:

$$\text{HR@K} = \frac{1}{N}\sum_{i=1}^{N} \text{I}\left(g_i \in \text{R}_i^K\right) \tag{13}$$

Here, N is the total number of valid evaluation samples (i.e., the number of decision steps where the action space is non-empty and a true next point exists), and I () is the indicator function. This metric reflects the model's capability to forecast the user's next behavior in a dynamic environment.

To ensure evaluation consistency, all models use the same candidate set generation logic and time window alignment mechanism. The proposed model and the DQN dynamic model update their state step by step and make new recommendations. PageRank-based and PredRNN, as sequence prediction models, output fixed-length rankings based on the global graph structure and LSTM hidden states, respectively, and select the top K items as recommendations.

HR@K indicates the percentage of top K recommended attractions that the user actually visits. For each user in the test set, the system extracts their current state from their historical trajectory, generates a top-K recommendation list, and compares this list with the user's actual check-in sequence. If the attraction the user actually visits next is on the recommended list, the recommendation is considered a hit. The hit rate is calculated as a global average, representing the proportion of hits across all valid evaluation samples. Fig. 5 shows the HR@K of each model.

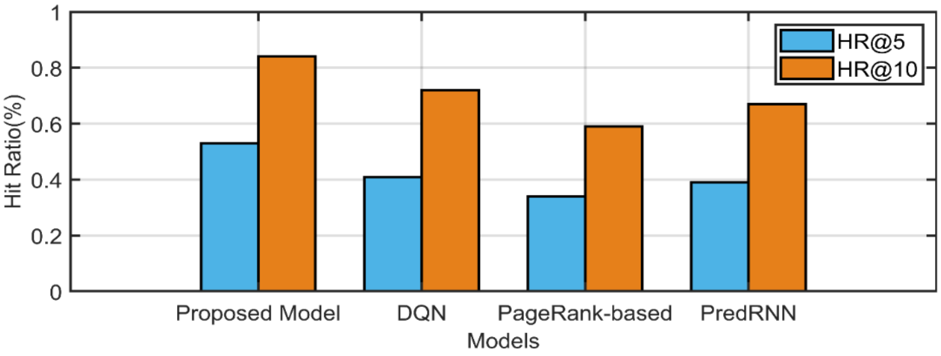


Figure 5: HR@K hit rate

Fig. 5 shows the HR@K hit rate. Under the HR@5 metric, this paper's model achieves a hit rate of 53%, exceeding baseline models such as DQN (41%), PageRank-based models (34%), and PredRNN (39%). When the K value is expanded to 10, the HR@10 hit rate of the paper's model reaches 84%, surpassing the three baseline models of DQN (72%), PageRank-based models (59%), and PredRNN (67%). The paper's model maintains a clear advantage. The HR@10 of the experimental group is higher than that of the GNN recommendation algorithm in Zhang et al.'s study (achieving 52.40%, 75.57%, and 72.43% on the Amazon-Beauty, Amazon-Games, and Amazon-CDs datasets, respectively). This demonstrates that the paper's model not only achieves high-precision recommendations for the first few attractions in the recommendation list, but also maintains high accuracy across a wider range of recommendations, providing users with more diverse choices. The standard deviations and confidence intervals are shown in Table 4.

Table 4: Top-K hit rate statistical significance (Mean ± Standard deviation, 95% Confidence interval)

Model	HR@5	HR@10
Proposed Model	53.0% ± 2.1% [52.1%, 53.9%]	84.0% ± 1.8% [83.3%, 84.7%]
DQN	41.0% ± 2.8% [40.0%, 42.0%]	72.0% ± 2.3% [71.2%, 72.8%]
PredRNN	39.0% ± 3.1% [38.0%, 40.0%]	67.0% ± 2.6% [66.1%, 67.9%]
PageRank-based	34.0% ± 3.5% [33.0%, 35.0%]	59.0% ± 3.0% [58.0%, 60.0%]

Table 4 presents the mean, standard deviation, and 95% confidence intervals for HR@5 and HR@10, calculated over 50 independent runs. The non-overlapping confidence intervals between the proposed model and all baselines confirm its statistically significant performance advantage.

To further compare recommendation accuracy, a coverage metric is added, which is defined as the ratio of the number of unique recommended attractions to the total number of attractions. This metric reflects the breadth of the recommendation system and its ability to discover low-hanging fruit. A high-coverage model can recommend not only popular attractions but also less popular ones that meet UPs, providing users with a richer and more diverse selection. The coverage data is shown in Table 5.

Table 5: Coverage statistics

Model	Coverage (%)
Proposed Model	78.8
DQN	54.7
PredRNN	62.1
PageRank-based	31.5

Table 5 shows that the paper's model has the highest coverage, reaching 78.8%, followed by PredRNN at 62.1%, DQN at 54.7%, and PageRank-based at 31.5%. Due to the paper's model's sensitivity to context and its penalty for repeated behavior, it can break out of its comfort zone of focusing on popular attractions and generate differentiated recommendations for different users and contexts, thus covering a wider range of attractions in the inventory. The PredRNN model can make personalized recommendations based on user history, but lacks exploration capabilities. The DQN model has exploration potential, but its ability to integrate UPs and context is weak. The PageRank-based model, driven by global popularity, repeatedly recommends a small number of popular attractions, often overlooking less popular ones that meet user needs.

3.3 Route rationality assessment

To assess the geographic coherence of routes, the experiment uses three quantitative metrics: average travel time, cross-region rate, and actual travel time. Using actual road network data, the shortest travel time between adjacent recommended attractions is calculated and compared with the model's recommended routes to verify whether they followed optimal or feasible transportation paths. Unreasonable spatial jumps within the route, such as long distances across different zones, are checked, indicating an illogical recommendation logic. Then, considering the overall duration of the recommended route relative to the user's actual available travel time, recommending unfeasible itineraries that exceed the user's time budget are avoided. Calculating the proportion of actual travel time to the total time needs to finish the route to reflect the time efficiency of the recommended route. Fig. 6 shows the average travel time, cross-zone rate, and actual travel time percentage.

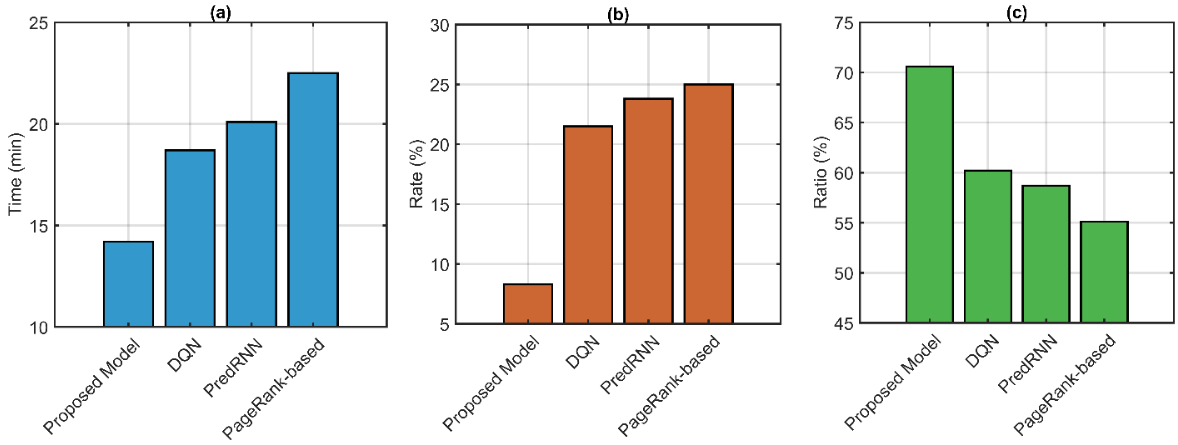


Figure 6: Average travel time, cross-region rate, and percentage of actual travel time; (a) Average travel time, (b) Cross-region rate, (c) Percentage of actual travel time

Fig. 6 shows that the paper's model significantly outperforms the baseline model in average travel time (14.2 minutes). The proposed model shortens average travel time by 8.3 minutes compared to the PageRank-based model, and the proposed model's cross-region rate (8.3%) is 16.7% lower than the PageRank-based model (25%). Compared to DQN, the paper's model shortens average travel time by 4.5 minutes and has a lower cross-region rate than DQN, demonstrating that the paper's model effectively integrates geographic information and generates coherent TRs. The actual travel time in the paper's model accounts for 70.6%, significantly higher than DQN (60.2%), PredRNN (58.7%), and PageRank-based models (55.1%). The paper's model recommends routes with shorter travel times and shorter waiting times, demonstrating superior rationality to the three baseline models.

3.4 Personalized matching satisfaction evaluation

To quantitatively evaluate how well recommendations match users' inherent preferences, this study uses user satisfaction scores as a key metric to assess the model's personalized performance. In experiments, the system creates personalized recommendation routes based on test users' historical check-in data. A panel of 30 evaluators (15 domain experts with advanced degrees and research experience, and 15 experienced travelers) conducts blind reviews. Inter-rater reliability, assessed via Cohen’s Kappa on a random subset, is 0.78 (95% CI [0.72, 0.84]), showing substantial agreement and confirming the evaluation's robustness. The scoring system uses a 5-point Likert scale, with 1 indicating "completely inconsistent with the user's interests" (e.g., recommending a high-intensity outdoor sports venue to a user who prefers cultural and artistic attractions) and 5 indicating "highly consistent with the user's preferences." The evaluation criteria cover five aspects: interest type matching: the consistency of the recommended attractions with the user's historical preferences (e.g., natural landscapes, historical sites, food streets, etc.). Tour pace adaptability: this refers to the degree to which the recommended itinerary's schedule (e.g., a packed morning of sightseeing, a leisurely afternoon) matches the user's historical behavior patterns. Preference intensity responsiveness: this refers to the ability to prioritize frequently visited attractions (e.g., recommending highly relevant museums to a "museum enthusiast"). Dynamic interest tracking: this refers to the ability to capture temporary shifts in user interest during an itinerary (e.g., a shift to indoor attractions during a sudden downpour) while maintaining consistent preferences. Recommendation logic explainability: this refers to the clarity and user understanding of the recommendation rationale (e.g., "Based on your visits to three art galleries last week, I recommend new museums of the same type"). User satisfaction evaluations are shown in Fig. 7.

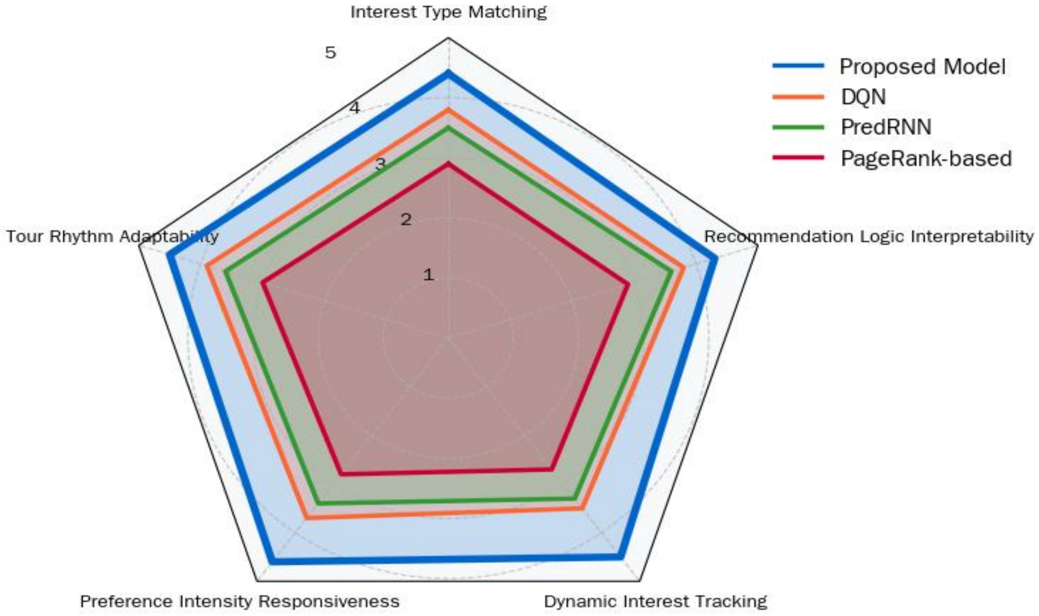


Figure 7: Satisfaction radar chart

As shown in Fig. 7, the satisfaction score for the model in this paper is notably greater than the satisfaction score for the baseline model. The recommended paths generated by the paper's model score 4.4, 4.5, 4.6, 4.5, and 4.3 in the five dimensions of interest type matching, tour rhythm adaptability, preference intensity responsiveness, dynamic interest tracking ability, and recommendation logic interpretability, respectively, with an average score of 4.46. DQN scores 3.8, 3.9, 3.7, 3.5, and 3.8, respectively, with an average score of 3.74. PredRNN scores 3.5, 3.6, 3.4, 3.3, and 3.6, respectively, with an average score of 3.48. The PageRank-based scores are 2.9, 3.0, 2.8, 2.7, and 2.9, respectively, with an average score of 2.86. This shows the effectiveness of the paper's model in continuously tracking UPs during dynamic interactions. In contrast, baseline models, either due to a lack of an explicit preference-context fusion mechanism or the limitations of static ranking logic, struggle to maintain personalization under environmental perturbations. This demonstrates that the paper's model can achieve a higher level of personalized matching.

3.5 Dynamic event and response delay testing

To evaluate the model's robustness and strategy adaptability during unexpected events, this study simulates a "temporary closure of a tourist attraction" to measure the system's reliability and responsiveness in providing alternative recommendations under extreme conditions. The assessment focuses on the model's closed-loop performance from plan failure to new route generation, demonstrating its fault tolerance and real-world adaptability in tourism.

The specific implementation process is as follows: during the testing phase, when a user completes their stop at a current attraction and is about to proceed to the next recommended destination, the system determines whether the destination is a "park-type" POI. If so, a "temporary closure" event simulation is triggered. The closed attraction is forcibly removed from the candidate set, and all subsequent recommendations are generated with respect to the updated action space. Upon activation, the system marks the attraction as "closed" and forcibly removes it from the candidate action space A_t. The weather variable is injected deterministically based on real-time weather data, ensuring that the simulated conditions are as realistic as possible. Simultaneously, the context vector c_t is updated, injecting "weather deterioration" or "crowd limit exceeded" flags to simulate real-world closure reasons.

After a trigger event, the system immediately re-executes the recommendation process: based on the updated state’ s , the set of reachable candidates is recalculated, excluding closed attractions and similar high-risk outdoor POIs. Accessible, open, and complementary alternative attractions (such as museums, shopping malls, and indoor exhibition halls) are prioritized. Any recommendations that are repeats from previous attractions are penalized with a -1 score to discourage repetition. The model outputs a new action probability distribution $\pi(a|s_t)$. If a legitimate and non-duplicate alternative attraction is recommended within one minute, it is considered a "successful transfer". The experiment is tested with 50 independent events to ensure statistical significance.

Two core metrics are measured: transfer success rate and response latency. The transfer success rate reflects the model's ability to adjust its strategy within a constrained action space, measuring the rate at which the model successfully re-executes recommendations during a "temporary shutdown" event. A higher rate indicates a stronger ability to propose a new plan when the original plan fails. Fig. 8 illustrates the transfer success rate curve.

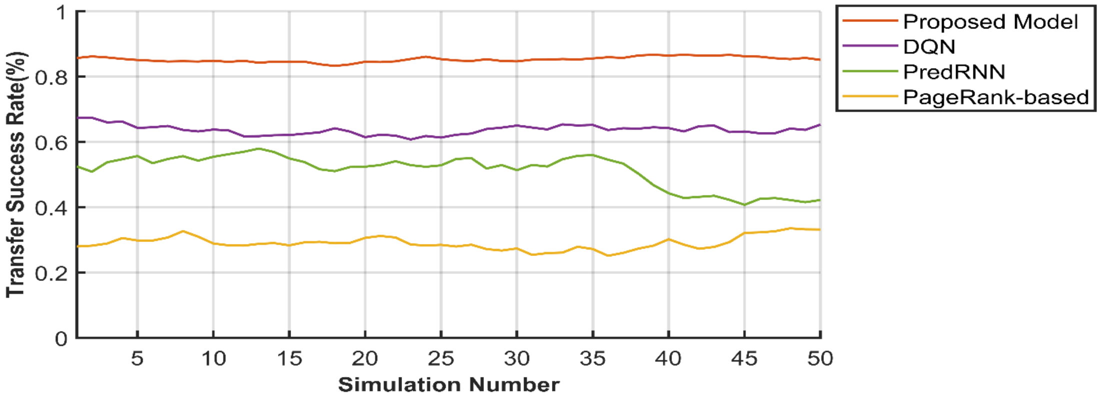


Figure 8: Transfer success rate curve

Fig. 8 shows that the paper's model's success rate significantly outperforms the three baseline models, reaching an average success rate of 85.3% with minimal fluctuation, demonstrating its ability to successfully handle unexpected situations in the vast majority of cases. In contrast, the DQN model has a lower average success rate of approximately 63.7%. The PredRNN and PageRank-based models perform even worse, with average success rates of 51.3% and 29.1%, respectively, and exhibiting significant fluctuations, indicating their limited adaptability to dynamic events. In the paper's model, when the "Attraction Closed" flag in the state vector is updated, the policy network immediately detects this change, enabling efficient and stable migration. While DQN also uses RL, it lacks deep modeling of UPs and an explicit penalty mechanism, resulting in sluggish and unstable responses to sudden state changes. PredRNN, as a sequence prediction model, relies too heavily on historical access patterns, making it difficult to dynamically adjust beyond the preset path. PageRank-based models are rarely able to generate effective alternatives and have the lowest success rate.

Response latency is the time interval (in seconds) from event triggering to the output of a new recommendation, accurately recorded by system logs. A latency of less than 3 seconds is considered efficient, while a latency exceeding 5 seconds may affect user experience. Response delay box plot is shown in Fig. 9.

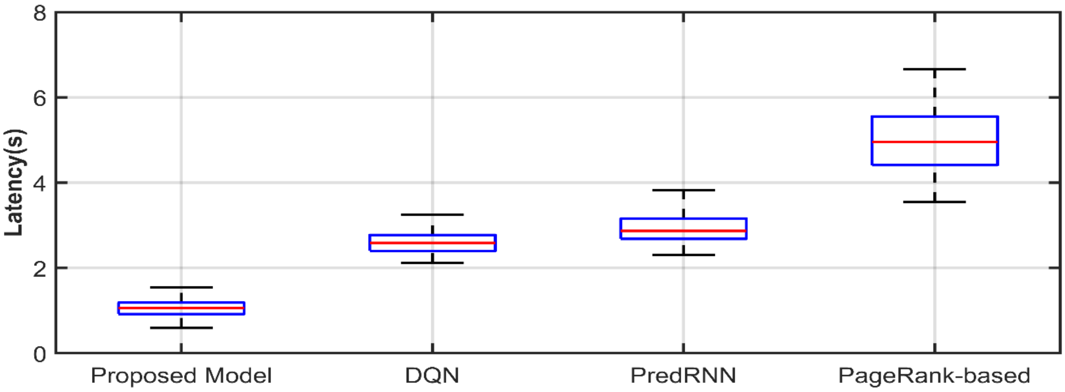


Figure 9: Response delay box plot

Fig. 9 compares response latencies. The horizontal axis signifies the four models, and the vertical axis denotes response latency. The data is based on the results of 50 independent tests. The data shows that the proposed model has extremely low response latency, with a maximum latency of 1.56 seconds and a minimum latency of 0.61 seconds, with a median of approximately 1.07 seconds. The overall distribution is compact, and the response is efficient and stable. In contrast, the DQN and PredRNN models experience significantly increased latency, with medians of approximately 2.61 seconds and 2.89 seconds, respectively. These wider bins indicate that their inference processes take longer and are more volatile, with maximum latencies of 3.26 seconds and 3.85 seconds, respectively, indicating slightly slower response times. The PageRank-based model has the highest latency, with a median of 4.97 seconds and a maximum of 6.68 seconds. This sluggish response to dynamic events may impact user experience. The paper's model can quickly calculate the new action probability distribution through efficient inference after the state vector is updated. However, standard deep learning models such as DQN and PredRNN require high computational overhead when processing high-dimensional states, while the latter requires reprocessing the entire historical sequence, resulting in high inference latency. The mean, median, standard deviation (SD), minimum, and maximum values are presented in Table 6.

Table 6: Statistical summary of response latency (in seconds)

Model	Mean	Median	SD	Minimum	Maximum
Proposed Model	1.05	1.07	0.18	0.61	1.56
DQN	2.58	2.61	0.22	2.13	3.26
PredRNN	2.84	2.89	0.29	2.33	3.85
PageRank-based	4.92	4.97	0.65	3.57	6.68

4 Discussion

This study proposes a deep reinforcement learning framework integrating LSTM-CNN and Multi-Head Attention Mechanism (MHAM) for real-time personalized travel route recommendation. On the FS-NYC and TCI datasets, the model achieves HR@5 of 53% and HR@10 of 84% (Figure 5), outperforming baselines due to its effective integration of User Preferences (UPs) and Real-Time Contexts (RTCs), where LSTM captures long-term behavior patterns, CNN-MLP processes contextual data (e.g., weather, traffic), and MHAM enables dynamic, fine-grained interest modeling by attending to relevant historical visits. The model also achieves high coverage (78.8%, Table 5), indicating strong diversity, driven by the Composite Reward Function (CRF) which uses a duplicate penalty and context-adaptive reward to promote exploration and mitigate the "filter bubble." Under dynamic events like attraction closures, it achieves an 85.3% migration success rate and a 1.07-second median response latency (Figures 8–9), demonstrating robust adaptability through its end-to-end AC architecture, which responds immediately to state changes, unlike the slower DQN, PredRNN, and static PageRank-based models.

In summary, the model's performance arises from the synergistic integration of LSTM-CNN, MHAM, CRF, and AC, enabling accurate, diverse, and highly adaptive real-time recommendations.

5 Conclusion

This study addresses the poor adaptability of recommendation systems in dynamic tourism scenarios by proposing a DRL model that integrates UPs with real-time contextual awareness. The proposed framework uses an LSTM-CNN-MHAM architecture within an Actor-Critic framework, guided by a Composite Reward Function, to achieve adaptive personalization by dynamically focusing on relevant historical behaviors based on real-time context. Based on the AC framework, a CRF is designed to drive the model to learn personalized and context-adaptive decision-making strategies. Experimental results show that the model achieves a Top-5 hit rate of 53% and a Top-10 hit rate of 84% on the FS-NYC and TCI, with a MRL of 1.07 seconds. It can be recognized that the FS-NYC and TCI datasets, though valuable, may not fully capture global travel diversity. Future work can test datasets from developing cities with less structured data and explore solutions to cold-start problems for new users, possibly using collaborative filtering or hybrid methods to enhance initial recommendations. This research effectively achieves collaborative modeling of user needs and dynamic environments, providing a personalized recommendation solution that combines high precision and real-time performance for smart tourism. The proposed model performs effectively in dynamic urban tourism but faces limitations like the 'cold-start' problem for new users without historical data, solvable through collaborative filtering or hybrid models. Scaling may raise computational costs from real-time processing, alleviated by model pruning or distributed computing. Future research can boost scalability for more users and contexts and incorporate adaptive event-triggered strategies to enhance responsiveness in complex urban environments.

Authorship contribution statement

Dan ZHANG: Supervision, Conceptualization, Project administration, Writing-Original draft preparation.

Conflicts of interest

The authors state that they have no conflict of interest concerning the publication of this paper.

Author statement

All authors have read and approved the manuscript, fulfilling the authorship criteria outlined earlier, and each author affirms that it represents honest work.

Funding

This work was supported by project of Chongqing Municipal Education Commission Humanities and Social Sciences Research Project: Research on the Mechanism of AI-Enabled Integrated and Innovative Development of Chongqing's Intangible Cultural Heritage and Tourism (No.25SKGH367).

Ethical approval

All authors have personally contributed significantly to the work behind this paper and will publicly stand by its content.

Reference

[1] Y. Zhang, M. Sotiriadis, and S. Shen, "Investigating the impact of smart tourism technologies on tourists' experiences," *Sustainability*, 14(5): 3048, 2022. https://doi.org/10.3390/su14053048

[2] C. Huda, A. Ramadhan, A. Trisetyarso, E. Abdurachman, and Y. Heryadi, "Smart tourism

recommendation model: a systematic literature review," *International Journal of Advanced Computer Science and Applications*, 12(12): 2021. DOI:10.14569/IJACSA.2021.0121222

[3] Y. Wang, M. Wang, K. Li, and J. Zhao, "Analysis of the relationships between tourism efficiency and transport accessibility—A case study in Hubei province, China," *Sustainability*, 13(15): 8649, 2021. https://doi.org/10.3390/su13158649

[4] C. M. Hall and Y. Ram, "Weather and climate in the assessment of tourism-related walkability," *Int J Biometeorol*, 65(5): 729–739, 2021. https://doi.org/10.1007/s00484-019-01801-2

[5] E. J. Wilkins and L. Horne, "Effects and perceptions of weather, climate, and climate change on outdoor recreation and nature-based tourism in the United States: A systematic review," *PLOS Climate*, 3(4): e0000266, 2024. https://doi.org/10.1371/journal.pclm.0000266

[6] R. Maršanic, E. Mrnjavac, D. Pupavac, and L. Krpan, "Stationary traffic as a factor of tourist destination quality and sustainability," *Sustainability*, 13(7): 3965, 2021. https://doi.org/10.3390/su13073965

[7] M. Kay Smith, I. Pinke-Sziva, Z. Berezvai, and K. Buczkowska-Gołąbek, "The changing nature of the cultural tourist: motivations, profiles and experiences of cultural tourists in Budapest," *Journal of Tourism and Cultural Change*, 20(1–2): 1–19, 2022. https://doi.org/10.1080/14766825.2021.1898626

[8] A. Saxena, N. K. Sharma, D. Pandey, and B. K. Pandey, "Influence of tourists satisfaction on future behavioral intentions with special reference to desert triangle of Rajasthan," *Augmented Human Research*, 6(1): 13, 2021. https://doi.org/10.1007/s41133-021-00052-4

[9] A. S. K. Xin, H. Y. Ting, and A. F. Atanda, "Trends in tourism recommendation systems: A review," *Journal of Computing Research and Innovation*, 9(2): 85–107, 2024. DOI:10.32628/CSEIT23902105

[10] R. Prahadeeswaran, "A comprehensive review: The convergence of artificial intelligence and tourism," *International Journal for Multidimensional Research Perspectives*, 1(2): 12–24, 2023.

[11] S. Vada, K. Dupre, and Y. Zhang, "Route tourism: a narrative literature review," *Current Issues in Tourism*, 26(6): 879–889, 2023. https://doi.org/10.1080/13683500.2022.2151420

[12] Mu. A. K. Anuar and A. Marzuki, "Critical elements in determining tourism routes: A systematic literature review," *Geografie*, 127(4): 319–340, 2022. 10.37040/geografie.2022.010

[13] Boulkroune, F. Zouari, and.A. Boubellouta, "Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems," *Journal of Vibration and Control*, 10775463251320258, 2025. https://doi.org/10.1177/10775463251320258

[14] Boulkroune, Abdesselem, et al. "Output-Feedback Controller Based Projective Lag-Synchronization of Uncertain Chaotic Systems in the Presence of Input Nonlinearities," *Mathematical Problems in Engineering* , 2017 (1): 8045803, 2017. https://doi.org/10.1155/2017/8045803

[15] Zouari, Farouk, K. Ben Saad, and M. Benrejeb, "Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems," *International Review on Modelling and Simulations* , 5(5): 2075-2103, 2012.https://www.scopus.com/pages/publications/84873265173

[16] Zouari, Farouk, Kamel Ben Saad, and Mohamed Benrejeb. "Adaptive backstepping control for a class of uncertain single input single output nonlinear systems." *10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13)*. IEEE, 2013. DOI: 10.1109/SSD.2013.6564134

[17] Rigatos, G., et al. "Nonlinear optimal control for a gas compressor driven by an induction motor." *Results in Control and Optimization* 11: 100226, 2023. https://doi.org/10.1016/j.rico.2023.100226

[18] Zouari, Farouk, Kamel Ben Saad, and Mohamed Benrejeb. "Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor." *2013 International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2013: 864-871, 2013. DOI: 10.1109/CoDIT.2013.6689656

[19] Ma, Xiaohang, Zhanyong Wu, and Ling Hu, "Deep Reinforcement Learning for Personalized Route Planning in Agricultural Tourism: A DDPG and Genetic Algorithm Approach," *Informatica*, 49(28), 2025. https://doi.org/10.31449/inf.v49i28.6865

[20] Zhu X. "Multi-Task Deep Reinforcement Learning for Intelligent Logistics Path Planning and Scheduling Optimization," *Informatica*, 49(20), 2025. https://doi.org/10.31449/inf.v49i20.7996

[21] M. Zhang, S. Wu, X. Yu, Q. Liu, and L. Wang, "Dynamic graph neural networks for sequential recommendation," *IEEE Trans Knowl Data Eng*, 35(5): 4741–4753, 2022. DOI: 10.1109/TKDE.2022.3151618

[22] Z. Liu, L. Yang, Z. Fan, H. Peng, and P. S. Yu, "Federated social recommendation with graph neural network," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4): 1–24, 2022. https://doi.org/10.1145/3501815

[23] G. K. Shyam, "DRL-HIFA: a dynamic recommendation system with deep reinforcement learning based Hidden Markov Weight Updation and factor analysis," *Multimed Tools Appl*, 83(29):72819–72843, 2024. https://doi.org/10.1007/s11042-024-18296-8

[24] X. Zhang, Y. Shang, Y. Ren, and K. Liang, "Dynamic multi-objective sequence-wise recommendation framework via deep reinforcement learning," *Complex & Intelligent Systems*, 9(2): 1891–1911, 2023. https://doi.org/10.1007/s40747-022-00871-x

[25] D. Shrestha, T. Wenan, D. Shrestha, N. Rajkarnikar, and S.-R. Jeong, "Personalized Tourist recommender system: a data-driven and machine-learning approach," *Computation*, 12(3): 59, 2024. https://doi.org/10.3390/computation12030059

[26] J. C. S. Núñez, J. A. Gómez-Pulido, and R. R. Ramírez, "Machine learning applied to tourism: A systematic review," *Wiley Interdiscip Rev Data Min Knowl Discov*, 14(5): e1549, 2024. https://doi.org/10.1002/widm.1549

[27] X. Chen, H. Zhang, C. U. I. Wong, and Z. Song, "Context-Aware Markov Sensors and Finite Mixture Models for Adaptive Stochastic Dynamics Analysis of Tourist Behavior," *Mathematics*, 13(12): 2028, 2025. https://doi.org/10.3390/math13122028

[28] J. Yoon and C. Choi, "Real-time context-aware recommendation system for tourism," *Sensors*, 23(7): 3679, 2023. https://doi.org/10.3390/s23073679

[29] Z. Wang, "Intelligent recommendation model of tourist places based on collaborative filtering and user preferences," *Applied Artificial Intelligence*, (37): 1, p. 2203574, 2023. https://doi.org/10.1080/08839514.2023.2203574

[30] X. Nan and X. Wang, "Design and implementation of a personalized tourism recommendation system based on the data mining and collaborative filtering algorithm," *Comput Intell Neurosci*, 2022(1): 1424097, 2022. https://doi.org/10.1155/2022/1424097

[31] G. Liu *et al.*, "Individualized tourism recommendation based on self-attention," *PLoS One*, 17(8): e0272319, 2022. https://doi.org/10.1371/journal.pone.0272319

[32] C.-Y. Tsai, K.-W. Chuang, H.-Y. Jen, and H. Huang, "A tour recommendation system considering implicit and dynamic information," *Applied Sciences*, 14(20): 9271, 2024. https://doi.org/10.3390/app14209271

[33] N. Mou *et al.*, "Personalized tourist route recommendation model with a trajectory understanding via neural networks," *Int J Digit Earth*, 15(1): 1738–1759, 2022. https://doi.org/10.1080/17538947.2022.2130456

[34] F. Zhou, P. Wang, X. Xu, W. Tai, and G. Trajcevski, "Contrastive trajectory learning for tour recommendation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(1): 1–25, 2021. https://doi.org/10.1145/3462331

[35] H. Lee and Y. Kang, "Mining tourists' destinations and preferences through LSTM-based text classification and spatial clustering using Flickr data," *Spatial Information Research*, 29(6): 825–839, 2021. https://doi.org/10.1007/s41324-021-00397-3

[36] X. Xiao, C. Li, X. Wang, and A. Zeng, "Personalized tourism recommendation model based on temporal multilayer sequential neural network," *Sci Rep*, 15(1): 382, 2025. https://doi.org/10.1038/s41598-024-84581-z

[37] S. Alshammrei, S. Boubaker, and L. Kolsi, "Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance," *Comput. Mater. Contin*, 72(3): 5939–5954, 2022. DOI: 10.32604/cmc.2022.028165

[38] L. Liu *et al.*, "Path planning for smart car based on Dijkstra algorithm and dynamic window approach," *Wirel Commun Mob Comput*, 2021(1):8881684, 2021. https://doi.org/10.1155/2021/8881684

[39] X. Feng, Z. Ma, C. Yu, and R. Xin, "MRNDR: multihead attention-based recommendation network for drug repurposing," *J Chem Inf Model*, 64(7): 2654–2669, 2024. https://doi.org/10.1021/acs.jcim.3c01726

[40] G. Liao, X. Deng, C. Wan, and X. Liu, "Group event recommendation based on graph multi-head attention network combining explicit and implicit information," *Inf Process Manag*, 59(2): 102797, 2022. https://doi.org/10.1016/j.ipm.2021.102797

[41] H. An and N. Moon, "Design of recommendation system for tourist spot using sentiment analysis based on CNN-LSTM," *J Ambient Intell Humaniz Comput*, 13(3): 1653–1663, 2022. https://doi.org/10.1007/s12652-019-01521-w

[42] T. Nguyen-Da, Y.-M. Li, C.-L. Peng, M.-Y. Cho, and P. Nguyen-Thanh, "Tourism demand prediction after COVID-19 with deep learning hybrid CNN–LSTM—Case study of Vietnam and provinces," *Sustainability*, 15(9): 7179, 2023. https://doi.org/10.3390/su15097179

[43] M. Bukhari, M. Maqsood, and F. Adil, "An actor-critic based recommender system with context-aware user modeling," *Artif Intell Rev*, 58(5): 138, 2025. https://doi.org/10.1007/s10462-025-11134-9

Contextual Embedding Comparison for Out-of-vocabulary Handling in Indonesian POS Tagging

Muhammad Alfian[1], Umi Laili Yuhana[*1], Daniel Siahaan[1], Harum Munazharoh[2], Eric Pardede[3]
[1]Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[2]Department of Indonesian Language and Literature, Universitas Airlangga, Surabaya, Indonesia
[3]Department of Computer and Information Technology, La Trobe University, Melbourne, Australia
E-mail: 7025221023@student.its.ac.id [1], yuhana@its.ac.id [1], do.siahaan@its.ac.id [1], harum.m@fib.unair.ac.id [2], e.pardede@latrobe.edu.au [3]
*Corresponding author

Keywords: Out-of-vocabulary, part-of-speech-tagging, education, contextual embedding, low resource language

Received: August 28, 2025

Out-of-vocabulary (OOV) problems remain a significant challenge in part-of-speech (POS) tagging. These problems affect not only tagging performance, but also downstream tasks, particularly in educational case studies. This issue is related to the limited availability of datasets for low-resource languages (LRLs), the absence of representative features, and the complexity of grammatical variation. Current approaches perform well in recognizing patterned OOV words, but often fail with unpatterned OOV words, such as proper nouns and polysemous words. To address this issue, this study employs contextual embeddings to represent OOV words, improving model recognition. Two types of embeddings are compared: static embeddings (Word2Vec, GloVe, and FastText) and contextual embeddings (ELMo, BERT, and Flair). These embeddings provide appropriate representations for OOV words. We evaluate models using accuracy and the macro F1 score on a curated Indonesian corpus of 30,960 words. The model was evaluated using the k-fold cross-validation method with both OOV and in-vocabulary (IV) word scenarios. The results of the experiment show that models with contextual embeddings outperform those with static embeddings. Flair achieved the highest level of accuracy (95.65%), while BERT and ELMo achieved similar levels of 92.73% and 91.61% respectively. Our proposed model was effective in handling OOV cases, achieving an accuracy of 88.12%, which is a 25.15% improvement over the baseline model. However, it still struggles with redundant words and capitalized letters. Future research should explore integrating form-based and contextual information to improve performance.

Povzetek: Študija za označevanje besednih vrst v jezikih z malo viri izboljša obravnavo OOV z uporabo kontekstualnih modelov, ki prekašajo statične pristope in opozarjajo na potrebo po združitvi oblikoslovnih in kontekstnih znakov.

1 Introduction

Part-of-speech (POS) tagging is a fundamental task in Natural Language Processing (NLP) because it provides essential information about sentence structure that influences subsequent processing. POS refers to the grammatical categories of words, such as verbs, adjectives, adverbs, and nouns. In this step, each word is automatically assigned the appropriate syntactic category based on its context [1], [2]. POS tagging has been shown to enhance the performance of various NLP tasks, including term extraction [3], sentence parsing [4] and text summarization [5]. In the field of education, it is also applied as a preprocessing step for syntactic analysis, serving as input for automatic grammar correction models [6] and automatic question generation [4]. By identifying word classes, text analysis becomes more detailed and precise [7]. However, progress in POS tagging has slowed in recent years due to persistent challenges in handling out-of-vocabulary (OOV) words which remain difficult to predict accurately [8], [9]. OOV words are a common challenge in NLP, particularly in POS tagging. OOV refers to words that are absent from the training vocabulary but appear during testing [10]. This issue can significantly reduce the performance of POS tagging models. Addressing OOV not only enhances tagging accuracy but also improves the performance of other NLP tasks, including Named Entity Recognition (NER) [11], sentence parsing [4], text classification [5-7], text summarization [8-11], and machine translation [18].

As shown in our previous systematic literature review [19], the OOV phenomenon is prevalent in low-resource languages (LRLs) such as Indonesian [20], Yoruba [21], and Ainu [22]. The limited size of available corpora prevents models from fully learning word-class patterns in these languages, making them more sensitive to novel word forms. Several studies have also linked OOV challenges to the lack of representative features for word representation [8], while others highlight linguistic factors, particularly grammatical variations. OOV words may arise from language and spelling variations [9]. For instance, in Indonesian, the word *makan* (“eat”) is a verb (VB), but with the affix *-an* it becomes *makanan* (“food”), which is labeled as a noun (NN). Such morphological processes often lead to OOV cases when derived forms are absent from dictionaries. OOV words can also emerge as neologisms driven by language evolution, such as emoticons (*“:)”*), expressive forms (*“Haaaa”*), abbreviations (*“krn”*), and slang (*“Tq”*). With the rise of social media, OOV words increasingly originate from code-switching and dialectal mixing [23], [24], where words from languages or dialects not included in dictionaries are also likely to be treated as OOV.

Several researchers have proposed strategies for handling OOV, including preprocessing strategies [25], [26], [27], hand-crafted features [20], [28], [29], and learned features [29], [30], [31]. Preprocessing is typically carried out using two methods: misspelling correction [26], [27] and vocabulary extension [25], [26]. Misspelling correction is straightforward and effective but only addresses misspelled OOV words, leaving other types unresolved. Vocabulary extension can also mitigate OOV cases; however, it requires access to large corpora to improve model performance. In low-resource languages, expanding corpora is often costly and time-consuming.

Several other researchers use a hand-crafted features approach to handle OOV [20], [28], [29]. This approach is effective in recognizing OOV words arising from morphological processes and works well in morphologically rich languages (MRLs). However, it is limited in handling unaffixed OOV words such as neologisms and is less applicable to highly inflected languages. The state-of-the-art approach relies on learned features. Neural network–based methods have been widely used to capture word patterns at different levels, including word-level [30], subword level [29], and character level [31]. Among these, character embeddings are reported to be the most effective in recognizing OOV word forms based on character patterns. Nevertheless, this method is less effective in handling previously unseen characters or word forms absent from dictionaries, such as proper nouns and foreign words from other languages or dialects.

Previous research has shown that pretrained embeddings can help address OOV by providing accurate vector representations learned from large-scale corpora. For example, ELMo has been used to represent OOV words and was shown to improve model performance [32]. Other studies have primarily relied on static embeddings such as word2vec [30], GloVe [23], or Fasttext [33]. Among these, FastText is the most widely used, as it represents words based on subword information. However, FastText primarily relies on surface forms, without considering the contextual meaning of words within sentences. Contextual information is essential for predicting labels of words with different forms but the same meaning and class. For instance, the Indonesian words *diri* (“self”) and *sendiri* (“self”) share both meaning and word class. A contextual approach therefore has strong potential to overcome these limitations, yet it has received little attention in addressing OOV in POS tagging.

Therefore, this study proposes contextual embeddings to represent OOV words with contextual information. To the best of our knowledge, this approach is the first applied to handle OOV words in Indonesian. We compare static embeddings (Word2Vec, GloVe, FastText) and contextual embeddings (BERT, Flair, ELMo) to examine the impact of contextual representations on POS tagging performance. The comparison also aims to analyze the strengths and limitations of each embedding in handling OOV words and to identify the most effective pretrained model. To maximize performance, we conduct hyperparameter tuning on six aspects: embedding size, BiLSTM hidden size, dropout rate, training batch size, test batch size, and learning rate. This process is carried out to determine the optimal configuration that yields the highest performance. The study uses a corpus from Fu’s research [34], which has been reselected and revalidated.

To address these goals, we formulate the following research questions:

RQ1: Does contextual embedding improve Indonesian POS tagging performance, especially for OOV words?

RQ2: Which embedding type achieves the best performance for OOV handling in Indonesian POS tagging?

RQ3: How can hyperparameter tuning enhance the performance of contextual embedding models in Indonesian POS tagging?

2 Previous works

Several researchers have explored strategies for handling OOV. This study classifies existing approaches into three categories: preprocessing strategies, hand-crafted features, and learned features. We summarize previous research in Table 1. A detailed explanation of each strategy is provided below.

2.1 Preprocessing strategies

A preprocessing strategy is applied to manipulate data before it is processed in the POS tagging model. Some researchers address OOV words by correcting misspellings [26], [27], [35]. Keiper et. al [26] and Jettakul et. al. [27] corrected spelling by replacing OOV words with the most similar words in the vocabulary, where word similarity is calculated using the Levenshtein distance method. Candidate words are selected based on the smallest distance and compared against a predefined threshold. Meanwhile, Millour and Fort [35] automatically generated spelling variant rules using AlphaMALIG to correct OOV words. This approach reduces the percentage of OOV from 24% to 22% [35]. However, spelling correction only succeeds for 29% of words, while 41% fail to be corrected, and the remaining 29% cannot be processed due to diverse noise levels in the data [27]. Preprocessing can be executed efficiently because of its simplicity, but its effectiveness is limited to a specific issue, requiring additional approaches to comprehensively address OOV challenges.

Table 1: Summary of previous works in handling OOV in POS Tagging

Method – Languages	**Categories – Metrics**	**Limitations**
Static Word Embedding [38] – Indonesian	Word Level – Accuracy	The model tends to mislabel minority classes and unknown words (noun or verb).
Multi-level OOV Handling [8] – English	Preprocessing strategies, Multi-level – Accuracy	The model reveals limitations in addressing misspellings and neologisms.
Hybrid HMM-BiLSTM [48] – Persian	Word Level – Accuracy	Hybrid models that rely on large-scale data pose challenges for low-resource languages.
Context Char Transformer Encoder (CCTE) [31] – English	Character Level – F1	The model is limited to OOV types beyond misspellings.
BiLSTM+CRF [23] – Bengali-Eng-lish, Hindi-English, Telugu-English	Multi-level - Precision, Recall, F1, Accuracy	The model struggles to distinguish between nouns, verbs, and proper names.
TSWR + OOVR [30] – Korean	Multi-level – Accuracy	The model requires substantial computational cost and prolonged training time.
Bayesian HMM [49] – Turkish, Hungarian, Finnish, Basque, English	Morphology, Word Level – NMI, VI	The model is less effective for inflectional languages such as English.
Semi-supervised Tree Models [25] – Indo-European (8) and Uralic (1)	Vocab Extension, Morphology – Accuracy	The small corpora limit the learning of morphological rules for minority labels.
Semi-markov CRF [41] – Chinese, Vietnamese, English, Japanese	Character Level – F1, Accuracy	The model is limited to 23-character words and struggles with inflected languages.
ELMo [32] – 57 languages in UD	Character Level – Accuracy	The study does not provide detailed insights into the model’s limitations.
Word and Character Embedding [33] – Italian	Multi-level – Accuracy	The model shows limited effectiveness for spoken or informal language.
Augmentation + Variation Rule [35] – Alsatian	Spelling Correction, Morphology – Accuracy	The model remains ineffective in handling short words (<4 characters).
Morph + CNN + Attention [29] – Greek	Orthography, Subword Level – Macro F1	The model remains less adaptive to neologisms and misspellings.
Multilingual POS Tagging [43] – English, Italian	Multi-level – Accuracy	The model remains limited in capturing spoken or informal language.
BERToov [50] – Indo-European (14), Uralic (1), Afro-Asiatic (1)	Subword level – Accuracy	The model distorts BERT's knowledge and its performance is inconsistent.
Two-level Backoff [42] – Thai, Chinese	Multi-level – Accuracy	The model suffers from overfitting with small mini-batch sizes.
BiLSTM-CRF [27] – Thai	Spelling Correction, Morphology – Macro F1	The model exhibits high error rates, particularly in fully and middle random cases.
Probabilistic POS Tagging [21] – Yoruba	Morphology – Accuracy, Precision, Recall	Limited corpus size and tagset availability impact model performance
Morph + BiLSTM [28] – Russian	Morphological, con-text, word embedding – Accuracy	The model remains limited in neologisms, proper names, misspellings, and informal words.
TPANN [51] – English	Multi-level – Accuracy	The model struggles on ARK-Twitter due to proper names and weak semantics.
HMM + MA [20] – Indonesian	Morphology – Accuracy	The model remains insufficient for OOV words without affixes.
character-based LSTM [40] – Kazakh, English, Finish, and Russian	Character Level – Accuracy	The model is less effective for inflectional languages like English.
Norm + Lex [26] – German	Vocab Extension, Spelling Correction, Morphology – Accuracy	The model fails to normalize short function words or effectively handle nouns and named entities.

NPT [37] – Farsi	Word Level – Accuracy	Morphological features fit inflectional languages suboptimally
WE-CRF [52] – English	Morphology, Word Level – Accuracy	The model risks overfitting when updating representations of low-frequency OOV words.
FSA-based word representations [53] – English	Character Level	The model fails to disambiguate categories such as adjectives and adverbs.

vocabulary to recognize more words [25], [26]. Janicki [25] added unlabeled words from development data, which improved model performance. Keiper et al. [26] utilized an external lexical corpus separate from the training data, raising accuracy to 90%. However, this strategy demands a large and continuously updated vocabulary to accommodate new words. While effective when the vocabulary size aligns with general language use, its effectiveness diminishes in limited data settings.

2.2 Hand-crafted features

Hand-crafted features are designed by experts based on domain knowledge [36]. To address OOV, some studies employ *morphological* and *orthographic* features. Morphological features capture affix information (prefixes and suffixes) using morphological analysis tools. For instance, Muljono et al. [20] applied MorphInd to extract affixes and clitics in Indonesian, Anastasyev et al. [28] used ABBYY Compreno for Russian, and Partalidou et al. [29] employed tools from the Institute of Language and Speech Processing (ILSP) for Greek. Similarly, Passban et al. [37] utilized stemming to extract affixes and variations in Farsi. Orthographic features, on the other hand, capture surface-level written information such as word shape [29], punctuation, and word case [28].

Feature-based methods rely on linguistic information, such as morphological and orthographic cues, to assign labels. Although these cues seem simple, studies by Ayogu et al. [21] and Muljono et al. [20] showed that morphological features, such as affixes, can enhance HMM accuracy. In neural models, Passban et al. [37] demonstrated that combining prefix, suffix, and variation features raised MLP accuracy to 97.38% (+2.73%). Similarly, Anastasyev et al. [28] found that combining morphological and orthographic features improved BiLSTM results. Morphological features generally outperform context-based ones [37], particularly in agglutinative and morphologically rich languages with stable spelling and grammar, such as Indonesian, which also contains clitics [20].

Although useful, linguistic information does not always improve model performance, particularly in highly inflected languages such as Greek [29]. Since morphological features rely on linguistic expertise, adapting them to each language requires additional effort. Anastasyev et al. [28] also demonstrated that relying solely on linguistic information may decrease accuracy because it fails to capture word context. Nevertheless, there is still potential to investigate more fine-grained orthographic features that have not yet been widely applied in POS tagging.

2.3 Learned features

Learned features are obtained from datasets through training processes designed to accomplish a specific task [36]. In the context of NLP, textual data is transformed into numerical vectors known as embeddings. Various approaches exist for constructing word embeddings, such as word-level, subword-level, character-level, and multi-level representations.

Yu et al. [30] employed Word2Vec to represent words, which offers two architectures: Skip-gram and CBOW. The Skip-gram model learns word vectors by predicting surrounding context words [30], while CBOW predicts a target word by averaging the vectors of its context words [37]. Yu et al. [30] further introduced a Skip-gram variation to construct Task-Specific Word Representations (TSWR) tailored for several NLP tasks. To address the OOV problem, they applied a mapping strategy to derive suitable word representations from Skip-gram vectors.

In addition, Yu et al. [30] represent words at the subword level using FastText. FastText extends Word2Vec by constructing word vectors from the average of subword (character n-gram) representations [29].For instance, the vector of the word *apple* is computed from the sum of its n-grams, such as “<ap”, “app”, “appl”, “apple”, “apple>”, “ppl”, “pple”, “pple>”, “ple”, “ple>”, and “le>”, where angle brackets mark word boundaries [31]. Fasttext can represent OOV words better than word level embedding [38]. In text classification, subword level representation effectively handled OOV [39]. Beyond

FastText, another study trained a BiLSTM model to learn representations of OOV words [30].[30][30][37][30][30][29][31][38][39][30]Meanwhile , character-level representation is widely used for handling OOV in deep learning–based models. Won et al. [31] explored the use of CNNs to construct character embeddings, while Makazhanov & Yessen-bayev [40] and Kemos et al. [41] investigated LSTM and BiLSTM architectures. Some studies also introduce variations by adding position markers at the beginning and end of words. In addition, Makazhanov & Yessenbayev [40] converted characters into 8-bit ASCII codes.

Several studies combine multiple levels of representation within a single framework. Marulli et al. [33] and Bhattu et al. [23] integrated word embedding with character embedding, while other works compared the effectiveness of different embedding methods, such as word2vec, fasttext [33], and glove [23]. Beyond word and character features, Boonkwan & Supnithi [42] enriched their model by incorporating morphological context through the BiRNN method. Pota et al. [43] derived word embeddings from character embeddings trained on pretrained embeddings using BiLSTM. Moudjari et al. [24] developed dialect-specific embeddings for Modern Standard Arabic, which encompasses multiple dialects. Yu et al. [30] proposed an Out-of-Vocabulary Representation (OOVR) by combining a modified word embedding (TSWR) with subword representation.

The learned features approach addresses OOV by leveraging embeddings trained on large datasets. Word-level embeddings are widely used due to their simplicity and efficiency, with Passban et al. [37] showing that CBOW (93.11% accuracy) surpassed GloVe (90.09%). While word-level embeddings capture semantic relations effectively, they fail with unseen words, morphological changes, and spelling errors. Subword-level embeddings overcome these issues and need a smaller vocabulary by modeling subword units, often outperforming static word vectors [29] Nonetheless, they struggle to capture broader sentence context and perform poorly with very short words (fewer than three characters).

Character-level embeddings can naturally address the OOV problem by capturing intrinsic features of words at the character level. They are more robust to morphological variations and misspellings, particularly in short words. However, character-level embeddings require more complex architectures and result in larger model sizes, which lead to longer training times. While they can effectively capture fine-grained information within words, they struggle to represent broader sentence-level context. Bhattu et al. [23] demonstrated that combining multiple levels of embeddings can achieve better performance than relying on a single type. This approach leverages embeddings trained on large-scale data to provide vector representations for OOV words, with a primary emphasis on character-level embeddings.

Character-level embedding focuses solely on learning the arrangement of characters within a word, without taking into account the surrounding context in which the word appears. As a result, the model may still fail to handle words that do not follow previously learned form patterns, such as foreign words from different dialects or languages. In contrast, Liu et al. [32] employed ELMo to represent OOV words by incorporating both sentence context and internal word structure. This method has been shown to enhance the performance of POS Tagging in addressing OOV. Nevertheless, only a limited number of studies have investigated the impact of contextual embeddings on improving POS Tagging performance, particularly in handling OOV. In particular, no study has utilized contextual embedding to handle OOV words in Indonesian.

Therefore, this study proposes a contextual embedding approach to address OOV. To the best of our knowledge, this approach is the first applied to handle OOV words in Indonesian. Several types of embeddings are compared to determine the most effective for improving Indonesian POS Tagging performance in handling OOV. In addition, hyperparameter tuning is conducted to identify the optimal configuration for maximizing performance gains.

3 Dataset preparation

This study employs a dataset from Fu et al. [34], consisting of 21,024 sentences and 355,021 words. However, our observations indicate that the dataset contains labeling errors [44], as it was created using a semi-automatic approach without revalidation. To address this, we revalidated a subset of sentences to ensure consistent and correct labels. The validation involved 25 selected linguists with an inter-annotator agreement of 0.868 using the Fleiss Kappa method, which demonstrate excellent consistency among annotators. They worked in groups of 3-4 people each. They used a voting method, with the majority vote (>50%) determining the most appropriate label for sentences containing the same word. When they encountered difficulties or deadlocks, one expert served as a reference for the entire group. As a result, we obtained 30,960 words from 3,175 validated sentences. The label distribution of the dataset is presented in Table 2, showing a

Table 2: Dataset distribution per label

Fold	Label	Words	Percentage
Adjective	JJ	859	2.77%
	JJS	24	0.08%
Adverb	MD	568	1.83%
	RB	2,065	6.67%
Noun	NN	6,912	22.33%
	NNP	5,403	17.45%
	SP	12	0.04%
Pronoun	PRD	999	3.23%
	PRF	61	0.20%
	PRI	6	0.02%
	PRL	1	0.00%
	PRP	901	2.91%
	WH	120	0.39%
Verb	VB	3,863	12.48%
	VO	32	0.10%
Conjunction	CC	24	0.08%
	SC	179	0.58%
	CD	758	2.45%
Determiner	DT	163	0.53%
	OD	39	0.13%
Interjection	UH	179	0.58%
Particle	P	114	0.37%
Preposition	IN	2,141	6.92%
	PO	24	0.08%
Punctuation and symbol	SYM	195	0.63%
	Z	4,983	16.09%
Miscellaneous	FW	113	0.36%
	ID	211	0.68%
	X	11	0.04%

Table 3: OOV words distribution per fold

Fold	Train	Validation	Test	OOV
1	22,105	2,458	6,397	1,037
2	22,309	2,564	6,087	1,056
3	22,237	2,507	6,216	1,106
4	22,288	2,536	6,136	1,010
5	22,253	2,583	6,124	999

predominance of noun groups (NN and NNP), punctuation (Z), and verbs (VB).

For model evaluation, the dataset was divided into training and testing sets with a 4:1 ratio. The training set was further split into training and validation subsets with a 9:1 ratio. To ensure robustness, we employed a five-fold cross-validation approach, allowing the model to be trained and tested on different data partitions. The distribution of training, validation, and testing data is presented in Table 3. Notably, approximately 16% (1,037 words) of the testing data were identified as OOV words.

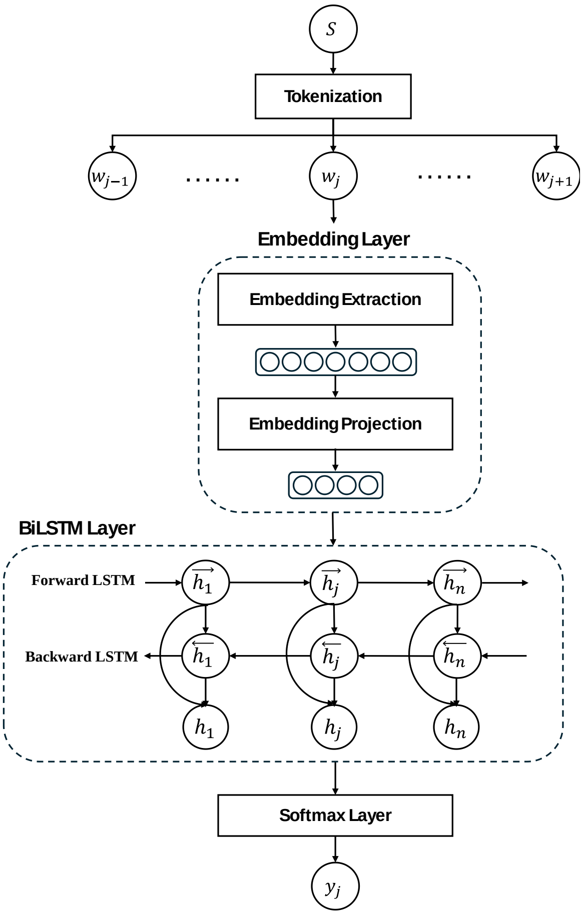


Figure 1: POS Tagging model architecture with Contextual embedding

4 Proposed model

This study introduces a vector representation based on contextual embedding to handle OOV in POS Tagging. The proposed approach is structured into four stages: tokenization, embedding layer, BiLSTM layer, and softmax layer, as illustrated in Figure 1. In the final stage, the model is trained with hyperparameter tuning and evaluated under multiple scenarios. Each stage is described in detail in the following subsections.

4.1 Tokenization

Tokenization is the process of segmenting a sentence (S) into words (w). In this study, tokenization is performed by splitting words based on spaces, while also separating punctuation marks unless they are inherently part of the word. Examples include reduplicated forms such as *barang-barang* (“goods”), legal references such as *39/M-Dag/Per/9/2009*, and dates such as *8/12/2005*. The original word form is preserved to avoid semantic distortion that may occur when a token is divided into excessively small units.

4.2 Embedding layer

The embedding layer converts words into vectors so they can be processed by the BiLSTM method. In the baseline model, the lookup embedding method generates a static vector with random values for each word. In this study, we propose an embedding layer consisting of two processes: embedding extraction and projection. Embedding extraction converts words into numeric vectors. In this study, three static embeddings (Word2Vec, GloVe, and FastText) and three contextual embeddings (BERT, ELMo, and Flair) were employed.

Static embedding assigns a unique vector representation to each word. These vectors are derived from training on a large corpus using specific algorithms. The methods and datasets employed in this study are described in the following subsections.

Word2vec[1] was trained using the Indonesian CoNLL17 corpus, which contains 2,899,107 words. The model is commonly trained using two alternative algorithms: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts a target word based on its surrounding context, whereas Skip-gram predicts the surrounding context given a target word. In this study, the Skip-gram algorithm was employed to generate word vector representations, with each word represented in a 100-dimensional vector space.

GloVe[2] was trained on the Common Crawl corpus consisting of 840 billion tokens. Unlike Word2Vec, which learns context from neighboring words, GloVe captures context based on the probability of word co-occurrence within the corpus. This allows GloVe to represent both global and local statistical information, enabling it to capture broader semantic relationships than Word2Vec. The resulting vector representation of GloVe has 300 dimensions.

Fasttext[3] is an extension of Word2Vec that was trained on the Wikipedia corpus and Common Crawl, with the number of tokens ranging from 10 to 100 billion depending on the language. Similar to Word2Vec, FastText can be trained using either the skip-gram or CBOW approach. However, unlike Word2Vec, FastText represents words as a collection of subword units by using character n-grams (typically with a length of 5) and a context window size of 5. This enables FastText to

[1] https://vectors.nlpl.eu/repository/20/50.zip

[2] https://nlp.stanford.edu/data/glove.840B.300d.zip

[3] https://dl.fbaipublicfiles.com/fasttext/vectors-crawl/cc.id.300.vec.gz

generate embeddings for rare and even out-of-vocabulary words. The resulting vector representation has a dimensionality of 300.

Contextual embeddings, on the other hand, assign a unique representation to each word depending on its context. The vector is dynamically generated based on the entire sentence in which the word appears. These embeddings are trained with more complex models and massive datasets, enabling them to produce context-dependent representations.

BERT[4] was trained on the large-scale Indo4B dataset containing 4 billion words from 12 corpora. Its input consists of subwords generated using the WordPiece method. Each subword is converted into an initial embedding, which is a combination of token embedding, position embedding, and token type embedding. These embeddings are then processed through a transformer architecture with multi-head self-attention and 12 layers of feedforward networks. BERT learns contextual representations through Masked Language Modeling (MLM) to predict missing words and Next Sentence Prediction (NSP) to capture sentence-level relationships. Since the output of BERT is at the subword level, the vectors of all subwords within a word are concatenated and averaged to form a single word representation. The final output is a vector representation with 768 dimensions.

ELMo[5] was trained on the Indonesian CoNLL17 corpus, which contains 2,899,107 words. Its input begins with character-level embeddings derived from character convolutions using the CNN method. The model is then trained with two LSTM layers, each processing the sequence in opposite directions—forward and backward—enabling ELMo to capture context-dependent nuances of word meaning. The final word representation is constructed through a weighted sum of three components: the character embedding, the forward LSTM, and the backward LSTM. The weights are optimized using the softmax activation function to balance the contribution of each component. The resulting ELMo embeddings have a dimensionality of 1024.

Flair[6] was trained on datasets from Wikipedia and OPUS containing 174,467,241 words. Flair employs a bidirectional LSTM architecture that processes sentences both forward and backward at the character level. The representation of each word is derived from character embeddings: in the forward layer, the final character of the word is used, while in the backward layer, the initial character is considered. This design enables Flair to capture contextual information from both preceding and succeeding words in a sentence. Each layer produces a 2048-dimensional vector, resulting in a combined word representation of 4096 dimensions.

The dimensionality of word embeddings varies significantly, ranging from 100 to 4096. Such differences create challenges in ensuring fair comparisons, and embeddings with very high dimensions can increase model complexity and lead to overfitting. To address this, a projection layer is applied to standardize and reduce the embedding dimensions. By applying this approach, the risk of overfitting from disproportionately large vectors is minimized. The final projection produced embeddings with 100 dimensions.

4.3 BiLSTM layer

This study employs the BiLSTM method as a POS Tagging model, following the approach proposed by Kurniawan & Aji [45]. BiLSTM operates with LSTM cells trained in two directions: forward ($\overrightarrow{h_t}$) and backward ($\overleftarrow{h_t}$), as shown in (1) and (2). Figure 1 illustrates the BiLSTM architecture, where the forward and backward layers are concatenated as in (3). a softmax layer is used for classification to interpret the BiLSTM output (h_t) as word classes (T_t). We employ a two-layer BiLSTM architecture to more effectively capture sentence context. Two-layer BiLSTM has been shown to produce better performance than 1 or 3 layers [46]. Each LSTM cell contains 100 neurons, producing a 200-dimensional vector representation.

$$\overrightarrow{h_t} = LSTM_{forward}(x_t, \overrightarrow{h_{t-1}}) \tag{1}$$

$$\overleftarrow{h_t} = LSTM_{backward}(x_t, \overleftarrow{h_{t+1}}) \tag{2}$$

$$h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t} \tag{3}$$

4.4 Softmax layer

We employ a classification layer to decode the BiLSTM outputs into word classes for POS tagging. A softmax layer with greedy decoding is used to select the word class by computing the highest probability $P(y)$ from the word class vector (V^y). The input to this layer is the BiLSTM output, denoted as h_t. The predicted word class y is expressed as shown in (4):

$$P(y = j|x) = \frac{e^{h_j}}{\sum_{k=0}^{K} e^{h_k}} \tag{4}$$

During training, the model maximizes the likelihood of the correct tag sequences. The final output sequence of tags is determined based on the highest score, computed as shown in (5):

$$y^* = \underset{y' \in y}{\text{argmax}}\ S(x, y') \tag{5}$$

4.5 Evaluation

This study evaluates the models using two metrics to assess the reliability of each embedding in handling OOV words.

[4] https://huggingface.co/indobenchmark/indobert-base-p1

[5] https://vectors.nlpl.eu/repository/20/158.zip

[6] https://flairnlp.github.io/docs/tutorial-embeddings/flair-embeddings

The first metric is *accuracy*, defined as the ratio of correctly predicted word classes to the total number of words (also referred to as Detection Accuracy). *Accuracy* measures the overall performance of the model without considering the distribution of data or label imbalances. It is calculated based on the number of correctly predicted positive labels (True Positives, TP) and negative labels (True Negatives, TN), divided by the total number of instances, as shown in (6):

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (6)$$

Table 4: Embedding performance comparison results

Category	Embedding	General		IV		OOV	
		Acc	F1	Acc	F1	Acc	F1
Baseline Model		85.81%	67.11%	90.43%	73.06%	62.97%	26.48%
Static Embedding	Word2vec	70.68%	57.59%	74.73%	62.52%	50.60%	24.43%
	GloVe	76.21%	59.14%	83.11%	64.28%	41.97%	21.67%
	Fasttext	92.03%	67.56%	93.98%	72.03%	82.40%	33.91%
Contextual Embedding	ELMo	91.61%	64.78%	92.54%	68.92%	87.06%	43.42%
	Flair	**95.63%**	**81.50%**	**97.14%**	**87.07%**	**88.12%**	**53.46%**
	BERT	92.73%	69.74%	94.35%	74.77%	84.73%	43.10%

The second metric is the *macro F1-score*, chosen because it evaluates the model's performance across all word classes, regardless of their distribution. This metric is particularly useful for assessing performance on minority labels and mitigating bias toward majority labels. The *macro F1-score* is calculated as the average F1 score across all labels (n), as shown in (7). The F1 score for each label is computed using the number of correctly predicted positive instances (True Positives, TP) divided by TP plus half of all misclassified instances (False Positives and False Negatives), as shown in (8).

$$macro\ F1 - score = \frac{\sum_{i=1}^{n} F1_i}{n} \quad (7)$$

$$F1 = \frac{TP}{TP+\frac{1}{2}(FP+FN)} \quad (8)$$

This study evaluated the models under three scenarios: a general scenario, which tests all words in the dataset, and two specific scenarios, focusing on particular subsets of words. The specific scenarios are divided into In-Vocabulary (IV) and OOV tests. IV evaluates words that appear in the training data (i.e., in the dictionary), while OOV assesses words that are not present in the training data.

4.6 Hyperparameter tuning

This study performed hyperparameter tuning to identify the optimal configuration for the POS Tagging model. The hyperparameters considered included learning rate, dropout, batch size, BiLSTM hidden size, and embedding size. Tuning was conducted using Optuna with 30 trials. The objective was to maximize accuracy and macro F1-score while minimizing the loss.

The learning rate is a hyperparameter that controls the speed at which the model learns. A high learning rate may cause instability and prevent the model from converging, whereas a low learning rate can slow down the learning process. In this study, learning rates ranging from 0.0001 to 0.1 were explored.

Dropout is a regularization hyperparameter that randomly deactivates nodes in a model, preventing over-reliance on dominant nodes. This technique helps make the model more robust to variations in the data. The dropout value indicates the proportion of nodes deactivated in a layer, with higher values resulting in more nodes being deactivated. In this study, dropout values ranging from 0.1 to 0.5 were evaluated.

Batch size is a hyperparameter that determines the number of data samples processed in one iteration. It is set separately for both training and testing data. Smaller batch sizes typically lead to faster updates, while larger batch sizes require more memory for temporary storage. In this study, batch sizes following a power-of-two pattern (8, 16, 32, 64, 128, 256) were evaluated.

BiLSTM hidden size is a hyperparameter that determines the dimensionality of the BiLSTM layer for capturing contextual patterns. Larger hidden sizes enable the BiLSTM to store more contextual information, while smaller sizes limit its capacity. In this study, hidden sizes ranging from 100 to 300 were evaluated.

Embedding size is a hyperparameter that determines the dimensionality of the vectors generated by the embedding layer, similar to the BiLSTM hidden size. It controls the number of values in the projected embedding representation. Larger embedding sizes allow the model to capture more contextual information, while smaller sizes limit this capacity. In this study, embedding sizes ranging from 20 to 150 were evaluated.

Several hyperparameters were tested simultaneously to identify the optimal configuration for the POS Tagging model. The contribution of each hyperparameter to variations in accuracy, macro F1-score, and loss was analyzed using the fANOVA approach. fANOVA computes the Fraction of Variance Explained (FoVE), quantifying the impact of each hyperparameter on changes in the target metric. Finally, the performance of the proposed model was compared with that of models from previous studies.

5 Experiment and results

The experiments were conducted on a shared computer equipped with an Intel i7-12700 (2.10 GHz) processor, 96 GB of RAM, and an NVIDIA GeForce RTX

3080 Ti GPU. This study uses the Adam optimizer to update weights iteratively based on training data. The model is trained for 50 epochs using an early stopping mechanism. Training stops if the dev loss increases for five consecutive epochs. The number of epochs for each fold varies, ranging from 28 to 44. However, the baseline model across all folds has the maximum number of epochs (50). This also affects the training duration of each model. The baseline and static embedding models require approximately three to four minutes of training time per fold. Meanwhile, the contextual embedding model requires between 30 and 70 minutes per fold, depending on the number of epochs.

The results from each fold were averaged to obtain the overall accuracy and macro F1-score, as presented in Table 4. Static embeddings generally underperform compared to the baseline model. Although FastText showed higher overall performance (+6.22%) relative to the baseline, its performance decreased slightly in the IV scenario (-1.03%). In contrast, contextual embeddings tended to outperform the baseline model. Among them, Flair achieved the best performance across both general and specific evaluations, including IV and OOV cases. However, in some scenarios, ELMo performed slightly worse than the baseline model.

5.1 Static embedding

Model performance tended to be lower when using Word2Vec (-15%) and GloVe (-10%) embeddings. Both are static embeddings trained on large amounts of word-level data. The reduced performance is primarily due to Word2Vec's inability to generate vectors for words outside its vocabulary. This is evidenced by the comparison of prediction results shown in Table 5, which illustrates the correlation between prediction accuracy and the presence of words in the embedding vocabulary.

Based on the Table 5, Word2Vec and GloVe were able to recognize only 24% (1,582 words) of the corpus used in this study. Among the unrecognized words, Word2Vec correctly predicted only 16% (256 words), while 83% (1,326 words) were predicted incorrectly. In contrast, GloVe correctly predicted 75% (1,200 words) and mispredicted 25% (382 words). Word2Vec's poor performance is attributed to its inability to provide vector representations for unknown words; the model uses random vectors for these words, resulting in inaccurate label predictions. For example, in sentence S-1, the words "*Cipinang*" and "*Melayu*" lacked vector representations in Word2Vec, causing the model to incorrectly assign them the label JJ (adjective).

Meanwhile, most of the GloVe model's prediction errors occurred for words that were present in the GloVe dictionary. The model was able to correctly predict unrecognized words as NN (nouns). Conversely, some words that existed in GloVe were predicted incorrectly. This is because GloVe was trained on English data, resulting in vector representations that did not align well with the Indonesian vocabulary. The vector representations generated by GloVe are illustrated in Figure 2. The vectors were visualized using t-SNE with a perplexity of 30, reducing the dimensionality to two dimensions for easier interpretation. For example, the words "*wanita*" (woman), "*warga*" (inhabitant), and "*negeri*" (country) were labeled NN (nouns), while the words "foreign," "direct," and "card" were labeled FW (foreign words). Meanwhile, the word "*Pondasi*" (foundation), which was labeled NN, was predicted as FW in sentence S-2. The GloVe representation for "*Pondasi*" was closer to words labeled FW than to words labeled NN, leading the model to assign it the FW label.

Table 5: Comparison of Word2vec and Glove prediction results

Prediction Status	**Word2vec**		**GloVe**	
	Avail.	**Unavbl.**	**Acc**	**Avail.**
True	4,171	256	3,658	1,200
False	643	1,326	1,156	382

Table 6: Calculation of the distance between the words *diri* (self) and sendiri (self)

Word	**Similarity**		
	Word2vec	**Glove**	**Fasttext**
Diri	0.95278	0.90645	0.68872
Sendiri			

Table 7: Statistical analysis of the words *diri* (self) and *sendiri* (self)

Fold	***diri***		***sendiri***	
	True	**False**	**Acc**	**True**
1	6	0	3	4
2	2	0	5	0
3	6	1	5	0
4	2	0	5	3
5	5	1	4	0
Total	21	2	22	7
Accuracy	**91.30%**		**75.86%**	

(S-1) *Kelurahan* (NNP) ***Cipinang*** (NNP) ***Melayu*** (NNP)
Cipinang (NNP) **Melayu** (NNP) Sub-district (NNP)

(S-2) ***Pondasi*** (NN) *itu* (PRD) *diletakkan* (PRD) *di* (IN) *dekat* (JJ) *taman* (NNP)
The foundation (NN) was (PRD) placed (PRD) in (IN) near (JJ) the park (NNP)

Meanwhile, the FastText model generally outperformed the default model, achieving an improvement of approximately 7%. FastText is a static embedding trained on large amounts of data at the subword level. Unlike Word2Vec and GloVe, FastText generates vector representations for combinations of subwords in Indonesian, enabling it to recognize word forms effectively. This allows FastText to provide vector representations even for words that are not explicitly present in its vocabulary. For example, the word "*dikembalikkannya*" (returned) in sentence S-3 is a verb with multiple affixes "di-*",* "*-kan*," and "*-nya*." FastText successfully identified it as a VB (verb) because it can recognize commonly used affix patterns in verbs.

However, strong word form recognition can also lead to prediction errors in PRF labels (reflexive pronouns). For instance, the words "*diri*" (self) and "*sendiri*" (self) under PRF labels have different vector representations, as shown in Table 6. The similarity value between these two words is lower compared to other embeddings, which causes the FastText model to fail in correctly predicting the label for "*sendiri*" (self). These prediction errors did not occur by chance; they were repeated. We analysed them statistically, as shown in Table 7. The prediction error for the word 'sendiri' occurred in folds 1 and 4. This resulted in the model achieving 75.86% accuracy in these challenging cases. However, subword-based word representation struggles to represent words with the same meaning but different forms, such as 'diri' and 'sendiri'.

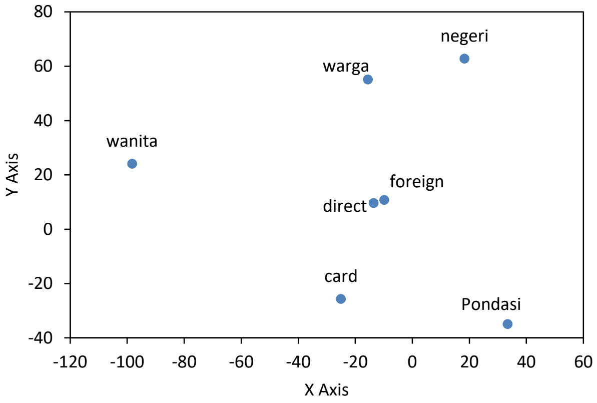


Figure 2: Illustration of the vector representation generated by GloVe

(S-3) *PDIP* (NN) *mendorong* (VB) ***dikembalikannya*** (VB) *kewenangan* (NN)
PDIP (NN) encourages (VB) **to return** (VB) the authority (NN)

Although the model's overall performance improved, FastText performed worse than the baseline model in handling IV cases. This is because FastText provides vector representations without considering the context of words within a sentence. Contextual information is crucial for correctly predicting labels for ambiguous words that have multiple possible labels. For example, in sentence S-4, the word "bawah" (under) functions as a noun because it is part of the phrase "parkir bawah tanah" (underground parking). However, the model predicted it as IN (preposition) because FastText identified "bawah" primarily as a preposition.

(S-4) *Parkir* (NN) ***bawah*** (NN) *tanah* (NN) *akan* (MD) *dibangun* (VB)
under (NN) ground (NN) parking (NN) will be (MD) built (VB)

5.2 Contextual embedding

ELMo performed well in handling OOV cases, achieving the second-best performance after Flair. ELMo captures context at the character level using two LSTM layers, which enables it to provide effective vector representations for previously unseen words, achieving an accuracy of 87.06%. For example, in sentence S-5, the word "Nudirman" was correctly labeled as NNP (proper name) even though the model has never encountered it as a person's name. However, ELMo performed less effectively on IV cases, with a macro F1 score of 68.92%, which is lower than the baseline model. ELMo underperforms on several labels, including NNP, RB and MB. The performance of several other minority labels, including PRF, SC, DT, OD, UH, P and ID, also declines. Consequently, ELMo's macro F1-score is lower than that of the baseline model. ELMo has a tendency to over-contextualise known words (IV). For example, the word 'Guangzhou' in the sentence S-6. In the training data, the word 'Guangzhou' is labelled NNP. However, during testing, ELMo mislabels it as NN because it captures contextual noise from surrounding NN-labelled words. This limitation arises because ELMo's two LSTM layers do not interact with each other, restricting the directional context processing. Additionally, the LSTM method tends to be biased towards frequently occurring labels in sentence.

(S-5) ***Nudirman*** (NNP) *pun* (P) *kembali* (VB) *menanggapi* (VB)
Nudirman (NNP) also (P) responded (VB)

(S-6) Kereta (NN) api (NN) **Guangzhou** (NNP) jalannya (NN) ke (IN) arah (NN) mana (WH)
Guangzhou Railway (NN) train (NN) route (NN) to (IN) which (NN) direction (WH)

The Flair model demonstrated a substantial performance improvement (+25%), particularly in OOV cases. Its performance also improved in ambiguous cases (+2%). Overall, the model achieved a significant gain (+10%) by leveraging Flair's contextual representation. Flair can recognize both OOV and IV words effectively because it captures word forms using a character-level BiLSTM and word context using a sequential BiLSTM. This dual-level modeling enables Flair to accurately predict labels for OOV and ambiguous words. For example, in sentence S-7, the word "terkalahkan" (defeated) was correctly labeled as VB because it contains typical verb affixes such as "*ter-*" and "-kan." Flair was also capable of distinguishing different meanings of the word "*baru*." In sentence S-8, "*baru*" (just) carries the meaning of "not long ago" and was labeled RB (adverb), whereas in sentence S-9, "*baru*" (new) conveys the meaning of "beginning" and was labeled JJ (adjective).

(S-7) Kami (PRP) tak (RB) **terkalahkan** (VB)
We (PRP) are undefeated (VB)

(S-8) *Kasus* (NN) *ini* (PRD) ***baru*** (RB) *pertama* (OD) *kali* (NN)
This (PRD) case (NN) is the first (OD) time (NN)

(S-9) *Memasuki* (VB) *awal* (NN) *tahun* (NN) ***baru*** (JJ)
Entering (VB) the beginning (NN) of the **new** (JJ) year (NN)

Although Flair performs well in most cases, it still struggles to predict certain OOV words with specific characteristics. For instance, the capitalized word "*Kuba*" in sentence S-10 should be labeled as NNP because it is a country name. In Indonesian, capitalized words generally indicate proper nouns. However, Flair failed to recognize this pattern and labeled "*Kuba*" as NN (common noun). Additionally, Flair was unable to correctly label reduplication words, such as "*Jarang-jarang*" (rarely) in sentence S-11, which was labeled NN instead of the correct JJ (adjective). While Flair effectively recognizes general word patterns, it has not yet mastered reduplication patterns. Since such words are relatively rare, specific guidance or features may be required to help Flair correctly identify them.

(S-10) *Hubungan* (NN) *AS* (NNP) – (Z) ***Kuba*** (NNP) *memanas* (VB)
US (NNP) – (Z) **Cuba** (NNP) relations (NN) heat up (VB)

(S-11) ***Jarang-jarang*** (JJ) *Gus* (NNP) *Dur* (NNP) *menolak* (VB) *berkomentar* (VB)
Rarely (JJ) Gus (NNP) Dur (NNP) refuses (VB) to comment (VB)

The BERT model achieved the second-best performance overall, after Flair. However, in OOV cases, BERT is outperformed by ELMo. Unlike other contextual embeddings, BERT uses a transformer-based Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) approach to capture the context of words in a sentence. BERT learns word patterns based on subword arrangements, allowing it to recognize the morphological forms of OOV words. For example, in sentence S-12, BERT correctly identified "*perumahan*" (housing) as NNP because it contains noun-specific affixes such as "*pe-*" and "*-an*". Furthermore, BERT also assigned the NNP label correctly to words like "*Pandau*" and "*Permai*" by leveraging the contextual relationship with surrounding words.

(S-12) ***Perumahan*** (NNP) ***Pandau*** (NNP) ***Permai*** (NNP)
Pandau (NNP) **Permai** (NNP) **Housing** (NNP)

However, BERT's performance did not show a significant improvement over the default model in the IV case. This is because BERT failed to correctly label certain word classes, such as OD, PRF, and CC. For instance, in sentence S-13, the word "*pertama*" (first) was labeled as CD (numerical), whereas the correct label is OD (ordinal). Similarly, in sentences S-14 and S-15, the word "*sendiri*" (alone) was incorrectly labeled as RB (adverb), and "*Selain*" (apart) was labeled as SC (coordinating conjunction), both of which do not match their true labels.

(S-13) *Perhitungan* (NN) *suara* (NN) *pilpres* (NN) *putaran* (NN) ***pertama*** (OD)
First (OD) round (NN) presidential (NN) election (NN) vote (NN) count (NN)

(S-14) *Peristiwa* (NN) *lori* (NN) *jalan* (VB) ***sendiri*** (PRF)
Incident of (NN) a lorry (NN) driving (NN) **on its own** (PRF)

(S-15) **Selain** (CC) buku (NN) , (Z) DPRD (NNP) juga (RB) menganggarkan (VB) pengadaan (NN) laboratorium (NN)
Apart from (CC) books (NN), (Z) DPRD (NNP) also (RB) budgeted (VB) for the procurement of (NN) laboratories (NN)

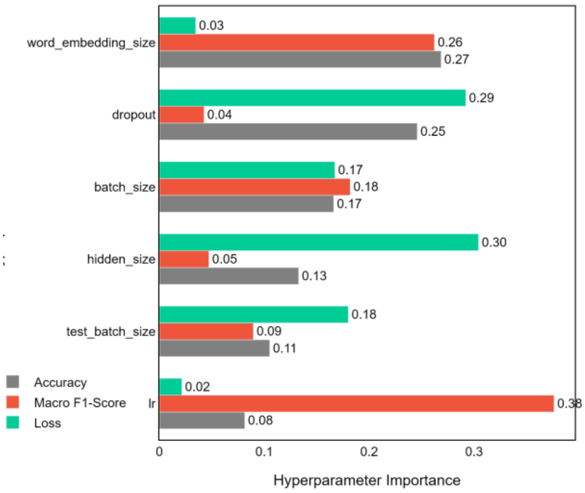


Figure 4: Hyperparameter importance graph against evaluation metrics

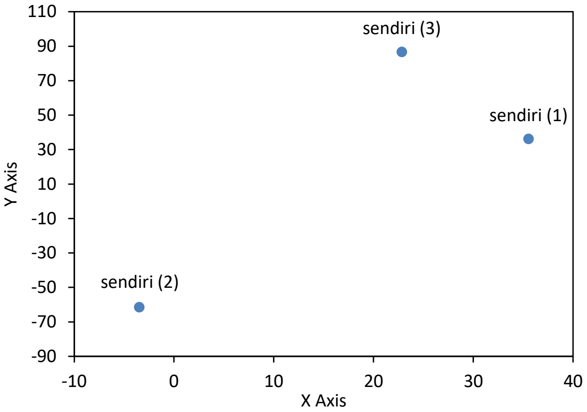


Figure 3: Illustration of a vector representation of the word "*sendiri*" (self) generated by BERT

This prediction error occurs because BERT represents words based on their context and position within the sentence. As a result, the vector representations of the same word vary from sentence to sentence. For example, Figure 3 shows the vector representations of the word "*sendiri*" (self) from three different sentences. Although the correct label is PRF, the vectors differ across sentences. In the first and third sentences, “*sendiri*” was correctly labeled, whereas in the second sentence, it was mislabeled because it appeared far from other occurrences of the word.

The experiments demonstrate that contextual embeddings have promising potential to enhance POS tagging performance in both IV and OOV cases. As their rankings indicate, contextual embeddings (Flair, BERT and ELMo) perform better than static embeddings. In general, both Flair and BERT perform well in terms of accuracy and macro F1-score metrics. To demonstrate the difference between the two models, we performed a t-test. The results of this test show that the t-statistic value for the difference between the BERT and Flair models is -24.59, with a p-value of 0.001. This proves that the BERT model differs from Flair by an average of 24.59, with Flair obtaining the highest value. A p-value of less than 0.05 indicates a statistically significant difference. Nevertheless, Flair still has limitations when it comes to handling certain out-of-vocabulary (OOV) words, particularly when it comes to distinguishing between noun (NN) and noun plural (NNP) labels. It is also not yet capable of correctly processing reduplicated word forms.

5.3 Hyperparameter tuning

To achieve maximum performance, this study optimized six hyperparameters: word embedding size, dropout, training batch size, BiLSTM hidden size, test batch size, and learning rate (lr) to determine the best configuration for the Flair-based POS Tagging model. Hyperparameter tuning was conducted using the Optuna library with 30 trials. The optimal configuration yielded a test macro F1 score of 84.66% and an accuracy of 95.28%, surpassing the model’s performance prior to tuning. The best-performing configuration consisted of an embedding size of 67, BiLSTM hidden size of 167, dropout of 0.26, training batch size of 8, testing batch size of 128, and a learning rate of 0.087235.

In addition, we analyzed the importance of each hyperparameter with respect to the target metrics, as shown in Figure 4. The analysis indicates that embedding size and dropout have the greatest influence on accuracy, with importance scores of 0.27 and 0.25, respectively. Meanwhile, hidden size and dropout have the strongest impact on loss, with importance scores of 0.30 and 0.29. Notably, higher dropout values correspond to lower loss.

Dropout regulates how many neurons are deactivated in each layer to prevent the model from over-relying on a single neuron in the embedding and BiLSTM layers. A low dropout value results in weak regularization, allowing the model to memorize specific patterns, which can lead to overfitting and lower accuracy. Conversely, a high dropout value enforces stronger regularization, allowing the model to learn more generalizable features. This makes the model more robust to new data, as reflected in the increased accuracy with a higher dropout value.

The hyperparameter with the greatest influence on the macro F1 score is the learning rate, which has an importance score of 0.38. Adjusting the learning rate significantly affects the performance of the BiLSTM model. However, no clear pattern was observed between changes in the macro F1 score and the learning rate. We hypothesize that the Flair model generates contextual embedding values with inherent randomness, so variations in the learning rate strongly impact other metrics. The fANOVA analysis confirms this significance, as changes induced by learning rate variations are highly influential.

6 Discussions

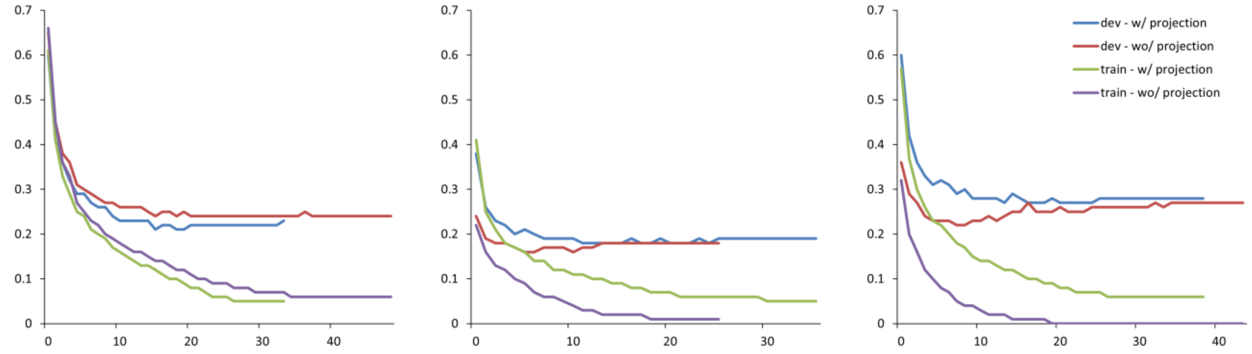
Our experiment used an embedding architecture with embedding projection. This approach prevented the model from becoming too complex due to receiving large vector inputs from pretrained embedding. To prove the effect of embedding projection, we conducted an ablation study of models with and without embedding projection, as shown in Table 8.

Table 8: Embedding performance with (w) and without (wo) projection layer.

Model	w/ Projection		wo/ Projection	
	Acc	F1	Acc	F1
Word2vec	70.68%	**57.59%**	**72.30%**	55.06%
GloVe	76.21%	59.14%	**81.78%**	**65.41%**
Fasttext	**92.03%**	**67.56%**	86.95%	61.16%
BERT [54]	92.73%	69.74%	**93.95%**	**77.93%**
ELMo	91.61%	64.78%	**93.88%**	**74.47%**
Flair	95.63%	81.50%	**95.77%**	**82.79%**

Based on Table 8, models that do not use embedding projection perform better than those that do. This is because the projection process removes some important information. Contextual embeddings, such as those produced by BERT, ELMo and Flair, contain syntactic and semantic information and have high dimensions of 768, 1024 and 4096 respectively. However, FastText produces the opposite result. FastText performs better when using embedding projection than when not using it. This is because FastText builds representations using subword n-grams, meaning its vectors sometimes contain redundant information. Embedding projection helps FastText reduce this redundant information, making the vector representation more compact and stable.

However, models without an embedding projection can make the model more complex, as the vector dimensions of pre-trained embeddings, especially contextual embeddings, are quite high. More complex models are more likely to overfit, which means that they appear to perform well but fail to handle new data. We also evaluated the model using a graph comparing train and dev loss to determine the impact of embeddings on model training stability as shown in Figure 5. FastText, with a dimension of 300, tends to be more stable with and without embedding projection. Meanwhile, ELMo and BERT show significant differences in loss. The training loss for the model without embedding projection reached its lowest point. However, the dev loss did not decrease as much. In fact, after the 10th epoch, the dev loss tended to increase. This indicates overfitting, whereby the model memorises the training data too much and fails to generalise. Consequently, the model is unable to predict the dev data correctly.



(a) Fasttext (b) ELMo (c) BERT

Figure 5: Grafik perbandingan train loss dan dev loss pada model with (w/) dan without (wo/) dengan projection embedding.

Meanwhile, models with embedding projection tend to have a smaller gap between training and development loss. This indicates that these models tend to be stable, as the information obtained from pre-trained embeddings is not processed immediately within the model. However, this information is adjusted and dimension-reduced so that the vector dimension is reduced, thereby reducing the model's complexity. Although models with embedding projection perform worse than models without, the former are better at handling data outside the training data. When handling OOV words, it is important to consider not only performance aspects such as accuracy and F1 matrices, but also the model's ability to generalise when handling new data.

The results of the experiment revealed that the Flair embedding model outperformed others. This is because the Flair model was trained using 174 million words, which is significantly more than the 4 million and 2 million words used to train the BERT and ELMo models, respectively. Additionally, the Flair architecture uses BiLSTM with character input, enabling Flair to recognise word and sentence patterns more effectively. Character embedding allows Flair to learn word structures based on letter sequences. Meanwhile, the BiLSTM method enables Flair to capture the context of words in sentences from two directions (forward and backward).

We compared our proposed model with previous POS Tagging models, including HMM, MEMM, BiLSTM, BiLSTM+CRF, and the most recent BiLSTM+CRF models incorporating morphological and character features. The comparison results are presented in Table 9. Based on the results in Table 9, our proposed BiLSTM model with contextual embedding (BiLSTM + context) outperformed previous POS Tagging models, achieving an accuracy of 95.63%. This performance surpasses that of the model reported by Kurniawan & Aji [45], which had an accuracy of 91.49%. Our proposed model improves the accuracy of previous research by Kurniawan [45] by 4.14% in general evaluations and by 12.38% in out-of-vocabulary (OOV) evaluations. Our proposed model also demonstrates superior performance compared to traditional machine learning models such as HMM (83.51%), MEMM (85.66%), and CRF (81.87%).

Table 9: Embedding performance comparison results

Model	General		IV		OOV	
	Acc	F1	Acc	F1	Acc	F1
HMM	83.51%	61.23%	94.91%	75.84%	27.13%	7.96%
MEMM	85.66%	49.69%	88.08%	53.56%	73.64%	20.04%
CRF	81.87%	55.16%	85.85%	59.49%	62.19%	23.53%
CRF [54]	90.60%	66.96%	92.80%	74.74%	79.70%	26.74%
BiLSTM	85.81%	67.11%	90.43%	73.06%	62.97%	26.48%
BiLSTM + CRF	85.27%	66.61%	90.17%	72.32%	60.95%	26.53%
BiLSTM + CRF [45]	91.49%	75.49%	94.67%	83.41%	75.74%	35.88%
BiLSTM + Context (prop)	**95.63%**	81.50%	**97.14%**	87.07%	**88.12%**	**53.46%**
BiLSTM + Context fine-tuned (prop)	95.41%	**82.42%**	97.05%	**89.43%**	87.30%	52.63%

Additionally, it significantly improves the handling of OOV words, achieving a +5% performance gain.

The fine-tuned model also demonstrated a higher macro F1-score compared to the unfine-tuned model, indicating its improved ability to predict minority labels. For instance, the word "*rasanya*" (feel like), which carries the SP label in sentence S-16, appears only once in the dataset, and the SP label is rarely used. Nevertheless, the model successfully captured the pattern of SP followed by the subword "-*nya*" and labeled it correctly. Similarly, the model accurately labeled other minority labels, such as the word "*disebutnya*" (said) in sentence S-17 with the VO label. This word contains a clitic affix attached to the verb, and the model is able to recognize this pattern and assign the correct label.

(S-16) ***Rasanya*** (SP) *aku* (PRP) *harus* (MD) *berhenti* (VB) *sekolah* (NN)
I feel (SP) that I (PRP) should (MD) stop (VB) school (NN)

(S-17) Pelaku (NN) penyerangan (NN) **disebutnya** (VO) adalah (MD) pemuda (NN)
The perpetrator (NN) of the attack (NN) was said (VO) to be (MD) a young man (NN)

The hyperparameter tuning in this study was conducted on a single fold with one run per configuration. Consequently, the reported results should be interpreted as indicative trends rather than statistically rigorous comparisons. Future research should perform repeated runs or multi-fold validation to enable confidence interval estimation and statistical testing.

However, the fine-tuned model still faced the same limitations as the previous model. It struggled to predict certain OOV words with specific characteristics, such as capitalized words and reduplications. In Indonesian, capital letters are often used at the beginning of sentences, in direct quotations, and for personal names, titles, nicknames, greetings, religions, nationalities, ethnicities, languages, months, days, events and places. Since capital letters are frequently used in object and personal names, we analysed the confusion matrix of the two labels (NN and NNP), as shown in Table 10.

While most words labelled NN (95.17%) can be labelled correctly, there are still 44 words labelled NN that are incorrectly labelled as NNP, as well as 24 words that are labelled with other labels. Similarly, most words labelled NNP (95.94%) can be labelled correctly, but 31 words labelled NN are incorrectly labelled as NNP, and 16 words are labelled with other labels. For instance, the word *'Indah'* in sentence S-18, which is labelled JJ. This word is usually used to describe nouns. However, in that sentence, it refers to a person's name. The model failed to label it correctly because the word *'indah'* has multiple meanings, person’s name (proper noun) and beautiful (adjective). In this case, the Flair model tends to produce vector representations similar to those of adjectives.

In addition to representation issues, there are problems related to annotation errors in the corpus, which result in suboptimal evaluation results. For instance, the word 'Bang' (Bro) in sentence S-19 is labelled as an NNP. This is correct because it is a nickname or greeting. However, the label for 'Bang' in the corpus is 'NN', so more careful label validation is needed to produce a high-quality corpus. Furthermore, discussions with language experts are required to establish guidelines for the application of NN and NNP labels to prevent similar labelling errors from occurring again.

(S-18) ***Indah*** (NNP) *menunjukkan* (VB) *ruangan* (NN)
Indah (NNP) shows (VB) the room (NN)

(S-19) ***Bang*** (NN) *taksinya* (NN) *pakai* (VB) *argo* (NN) *tidak* (RB)
Bro (NN) his taxi (NN) uses (VB) the meter (NN) doesn't (RB)

We also analysed prediction errors in reduplicated words using the confusion matrix shown in Table 11. Reduplicated words have a unique form compared to other words. They are written by repeating the base word and adding a hyphen between the two parts. Based on Table 11, reduplicated words with the labels 'NN' and 'NNP' can be distinguished with 100% accuracy. This is because reduplicated words with these labels have simple patterns. The words are repeated without undergoing any changes. For example, the word '*anak-anak*' (children) in sentence S-20 is formed from the word '*anak*' (child). Although it has undergone a change in form, the words ‘anak’ and ‘anak-anak’ still have the same label (NN).

However, reduplicated words with the labels VB, JJ, RB and IN cannot yet be predicted correctly. This is because the word is rarely used and changes when repeated. For example, the word '*bolak-balik*' (back and forth) in sentence S-21 is formed from the word '*balik*'

Table 10: Confusion matrix of capitalized words

Predict Actual	NN	NNP	Other
NN	**1341**	44	24
NNP	31	**1113**	16

Table 11: Confusion matrix of reduplication words

Pred Act	NN	NNP	VB	JJ	RB	IN
NN	**19**	0	0	0	0	0
NNP	0	**2**	0	0	0	0
VB	1	0	**3**	1	2	0
JJ	2	0	0	**4**	1	0
RB	0	0	0	1	**6**	0
IN	1	0	0	0	0	**0**

(back). The vowels change when forming a reduplicated word. Although both words are verbs, the model predicts them as adjectives. Another example is the word '*bersama-sama*' (together) in sentence S-22, which is formed from the word '*sama*' (same). The prefix '*ber-*' is added to the base word, resulting in a longer reduplicated word. Additionally, the label changes from an adjective (JJ) to a verb (VB). The Flair model is unable to handle this case because reduplicated words are rare. Furthermore, reduplicated words can undergo vowel changes and affixation, which can alter the label of the base word.

(S-20) ***Anak-anak*** (NN) *sudah* (RB) *siap* (VB)
Children (NN) are (RB) ready (VB)

(S-21) *Tamsil* (NNP) *harus* (MD) ***bolak-balik*** (VB) *memenuhi* (VB) *panggilan* (NN)
Tamsil (NNP) must (MD) go back and forth (VB) to answer (VB) the call (NN).

(S-22) *Warga* (NN) *melakukan* (VB) *penyisiran* (NN) *Pantai* (NN) *secara* (IN) ***bersama-sama*** (VB)
Residents (NN) conducted (VB) a cleanup (NN) of the beach (NN) with (IN) **together** (VB).

This highlights the need for additional information, such as orthographic cues or word shape, to help Flair correctly recognize these words. In this study, our focus was on maximizing Flair embedding performance through hyperparameter tuning. Meanwhile, other contextual embeddings, such as ELMo and BERT, have the same potential as Flair for overcoming OOV issues. However, their performance is lower than Flair's in the default hyperparameter setting. This potential remains unexplored in this study and could be a focus of future research. In addition, the scope of this study is limited to testing one language only: Indonesian. It would be interesting to discuss whether future research should involve cross-lingual testing, given that each language has its own grammar.

Future research will focus on leveraging such shape information to further improve the performance of contextual embedding–based models. Previous research has shown that morphological and orthographic information, such as prefixes, suffixes, capital letters, and surface forms, can improve model performance when using HMM [25] and CRF [21] methods. We also plan to use the latest character-level embedding approach to capture morphological information more comprehensively. We will apply the attention mechanism to capture global context information in order to address the issues of polysemy and proper nouns that are still problematic in POS tagging. In addition, an adaptive approach using fuzzy logic [47] is also promising when dealing with uncertain conditions, such as those encountered in OOV. These methods present further potential for development to overcome the limitations of contextual models.

7 Conclusion

OOV remains an unresolved problem in POS tagging, primarily due to limited datasets in low-resource languages (LRLs) and the lack of representative features. This challenge is further complicated by the high complexity of grammatical variations. State-of-the-art approaches often rely on character-level embeddings to recognize OOV word forms. However, such information is still insufficient for handling unpatterned OOV words, such as proper nouns and polysemous terms. To address this limitation, this study employed contextual embeddings for more effective OOV handling.

This study compared two types of embeddings—static (Word2Vec, GloVe, FastText) and contextual (ELMo, BERT, Flair)—to evaluate their effectiveness in handling OOV cases. The best-performing embeddings were further fine-tuned to optimize the model by adjusting several hyperparameters, including embedding size, BiLSTM hidden size, dropout rate, training batch size, test batch size, and learning rate. We benchmarked the proposed model against previous approaches based on machine learning (HMM, MEMM, CRF) and deep learning (BiLSTM, BiLSTM+CRF). Model performance was assessed using accuracy and macro F1-score, under two evaluation scenarios: general evaluation and specific evaluation for in-vocabulary (IV) and out-of-vocabulary (OOV) words.

The evaluation results indicated that models with contextual embeddings outperform other approaches. Among the contextual embeddings, Flair achieved the highest performance with an accuracy of 95.65%. Our proposed model also proved effective in handling OOV cases, reaching 88.12% accuracy. Fine-tuning experiments further revealed that hyperparameters such as embedding size, dropout rate, hidden size, and learning rate significantly affect both accuracy and macro F1-score. Despite Flair's strong performance, several OOV cases remained challenging, particularly word reduplication and capitalized OOV words. Future research should therefore explore the integration of shape-based features to enhance the performance of contextual embedding models.

Acknowledgement

This research was supported in part by Institut Teknologi Sepuluh Nopember [scholarship number 1333/IT2/T/HK.00.01/2022] and the Ministry of Education, Culture, Research, and Technology of Indonesia [grant number 038/E5/PG.02.00.PL/2024]. The authors would like to thank the 25 students from the language and literature department of Airlangga University for assisting in analyzing the data and discussing Indonesian grammar.

References

[1] A. Chiche and B. Yitagesu, "Part of Speech tagging: A systematic review of Deep Learning and Machine Learning approaches," *J Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00561-y.

[2] M. Alfian, U. L. Yuhana, and D. Siahaan, "Indonesian Part-of-Speech tagger: A comparative study," in *2023 10th International Conference on Advanced Informatics: Concept, Theory and Application (ICAICTA)*, IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/ICAICTA59291.2023.10390353.

[3] A. Kalykulova and A. Nugumanova, "T-Extractor: A Hybrid Unsupervised Approach for Term and Named Entity Extraction Using Rules, Statistical, and Semantic Methods," *Informatica (Slovenia)*, vol. 49, no. 2, pp. 299–318, 2025, doi: 10.31449/inf.v49i2.8148.

[4] S. F. Kusuma, D. O. Siahaan, and C. Fatichah, "Automatic question generation with various difficulty levels based on knowledge ontology using a query template," *Knowl Based Syst*, vol. 249, p. 108906, Aug. 2022, doi: 10.1016/j.knosys.2022.108906.

[5] M. Z. Abdullah and C. Fatichah, "Feature-based POS tagging and sentence relevance for news multi-document summarization in Bahasa Indonesia," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 541–549, 2022, doi: 10.11591/eei.v11i1.3275.

[6] L. Hu, Y. Tang, X. Wu, and J. Zeng, "Considering optimization of English grammar error correction based on neural network," *Neural Comput Appl*, vol. 34, no. 5, pp. 3323–3335, Mar. 2022, doi: 10.1007/S00521-020-05591-2/FIGURES/17.

[7] D. Hoesen and A. Purwarianti, "Investigating Bi-LSTM and CRF with POS Tag Embedding for Indonesian Named Entity Tagger," *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, pp. 35–38, 2019, doi: 10.1109/IALP.2018.8629158.

[8] J. V. Lochter, R. M. Silva, and T. A. Almeida, "Multi-level out-of-vocabulary words handling approach," *Knowl Based Syst*, vol. 251, Sep. 2022, doi: 10.1016/j.knosys.2022.108911.

[9] P. Kolachina, M. Riedl, and C. Biemann, "Replacing OOV Words For Dependency Parsing With Distributional Semantics," in *NoDaLiDa 2017 - 21st Nordic Conference of Computational Linguistics, Proceedings of the Conference*, 2017, pp. 11–9.

[10] S. Garcia-Bordils *et al.*, "Out-of-Vocabulary challenge report," in *Computer Vision -- ECCV 2022 Workshops*, 2023, pp. 359–375. doi: 10.1007/978-3-031-25069-9_24.

[11] X. Cai, S. Dong, and J. Hu, "A deep learning model incorporating part of speech and self-matching attention for named entity recognition of Chinese electronic medical records," *BMC Med Inform Decis Mak*, vol. 19, 2019, doi: 10.1186/s12911-019-0762-7.

[12] Imamah, U. L. Yuhana, A. Djunaidy, and M. H. Purnomo, "Development of text classification based on difficulty level in adaptive learning system using Convolutional Neural Network," *International Electronics Symposium 2021: Wireless Technologies and Intelligent Systems for Better Human Lives, IES 2021 - Proceedings*, pp. 238–243, Sep. 2021, doi: 10.1109/IES53407.2021.9594021.

[13] F. Gargiulo, S. Silvestri, M. Ciampi, and G. De Pietro, "Deep Neural Network for hierarchical extreme multi-label text classification," *Applied Soft Computing Journal*, vol. 79, pp. 125–138, 2019, doi: 10.1016/j.asoc.2019.03.041.

[14] S. Chotirat and P. Meesad, "Part-of-Speech tagging enhancement to Natural Language Processing for Thai WH-Question classification with Deep Learning," *Heliyon*, vol. 7, no. 10, 2021, doi: 10.1016/j.heliyon.2021.e08216.

[15] S. K. Nambiar, S. Peter David, and S. Mary Idicula, "Abstractive summarization of text document in Malayalam language: enhancing attention model using POS tagging feature," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 2, 2023, doi: 10.1145/3561819.

[16] W. Liu and L. Wang, "POS-tagging enhanced Korean text summarization," in *Intelligent Computing Methodologies*, Springer International Publishing, 2017, pp. 425–435. doi: 10.1007/978-3-319-63315-2_37.

[17] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic Text Summarization: A comprehensive survey," Mar. 01, 2021. doi: 10.1016/j.eswa.2020.113679.

[18] V. H. Vu, Q. P. Nguyen, K. H. Nguyen, J. C. Shin, and C. Y. Ock, "Korean-Vietnamese neural machine translation with named entity recognition and part-of-speech tags," *IEICE Trans Inf Syst*, vol. E103D, no. 4, 2020, doi: 10.1587/transinf.2019EDP7154.

[19] M. Alfian, U. L. Yuhana, D. Siahaan, H. Munazharoh, and E. Pardede, "Out-of-Vocabulary Handling in Part-of-Speech Tagging: A Semantic Web-Driven Systematic Review," *Int*

J Semant Web Inf Syst, vol. 21, pp. 1–36, Sep. 2025, doi: 10.4018/IJSWIS.388421.

[20] Muljono, U. Afini, and C. Supriyanto, "Morphology analysis for Hidden Markov Model based Indonesian Part-of-Speech tagger," in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, 2017, pp. 237–240. doi: 10.1109/ICICOS.2017.8276368.

[21] I. I. Ayogu, A. O. Adetunmbi, B. A. Ojokoh, and S. A. Oluwadare, "A comparative study of hidden Markov model and conditional random fields on a Yorùba part-of-speech tagging task," in *Proceedings of the IEEE International Conference on Computing, Networking and Informatics, ICCNI 2017*, 2017. doi: 10.1109/ICCNI.2017.8123784.

[22] K. Nowakowski, M. Ptaszynski, F. Masui, and Y. Momouchi, "Improving Basic Natural Language Processing Tools for the Ainu Language," *Information 2019, Vol. 10, Page 329*, vol. 10, no. 11, p. 329, Oct. 2019, doi: 10.3390/INFO10110329.

[23] S. N. Bhattu, S. K. Nunna, D. V. L. N. Somayajulu, and B. Pradhan, "Improving code-mixed POS tagging using code-mixed embeddings," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, no. 4, p. 1, 2020, doi: 10.1145/3380967.

[24] L. Moudjari, F. Benamara, and K. Akli-Astouati, "Multi-level embeddings for processing Arabic social media contents," *Comput Speech Lang*, vol. 70, 2021, doi: 10.1016/j.csl.2021.101240.

[25] M. Janicki, "Semi-supervised induction of POS-tag lexicons with tree models," in *International Conference Recent Advances in Natural Language Processing, RANLP*, 2019, pp. 507–515. doi: 10.26615/978-954-452-056-4_060.

[26] L. Keiper, A. Horbach, and S. Thater, "Improving POS tagging of German learner language in a reading comprehension scenario," in *Proceedings of the 10th International Conference on Language Resources and Evaluation, LREC 2016*, 2016.

[27] A. Jettakul, C. Thamjarat, K. Liaowongphuthorn, C. Udomcharoenchaikit, P. Vateekul, and P. Boonkwan, "A comparative study on various Deep Learning techniques for Thai NLP lexical and syntactic Tasks on noisy data," in *Proceeding of 2018 15th International Joint Conference on Computer Science and Software Engineering, JCSSE 2018*, 2018. doi: 10.1109/JCSSE.2018.8457368.

[28] D. G. Anastasyev, A. I. Andrianov, and E. M. Indenbom, "Part-of-speech tagging with rich language description," in *Komp'juternaja Lingvistika i Intellektual'nye Tehnologii*, 2017.

[29] E. Partalidou, E. Spyromitros-Xioufis, S. Doropoulos, S. Vologiannidis, and K. I. Diamantaras, "Design and implementation of an open source Greek POS Tagger and Entity Recognizer using spaCy," in *Proceedings - 2019 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2019*, 2019, pp. 337–341. doi: 10.1145/3350546.3352543.

[30] H. Yu, J. An, J. Yoon, H. Kim, and Y. Ko, "Simple methods to overcome the limitations of general word representations in natural language processing tasks," *Comput Speech Lang*, vol. 59, pp. 91–113, 2020, doi: 10.1016/j.csl.2019.04.009.

[31] M. S. Won, Y. S. Choi, S. Kim, C. W. Na, and J. H. Lee, "An embedding method for unseen words considering contextual information and morphological information," in *Proceedings of the ACM Symposium on Applied Computing*, 2021, pp. 1055–1062. doi: 10.1145/3412841.3441982.

[32] Y. Liu, W. Che, Y. Wang, B. Zheng, B. Qin, and T. Liu, "Deep contextualized word embeddings for universal dependency parsing," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, no. 1, pp. 1–17, 2019, doi: 10.1145/3326497.

[33] F. Marulli, M. Pota, and M. Esposito, "A comparison of character and word embeddings in bidirectional LSTMs for POS tagging in Italian," in *Smart Innovation, Systems and Technologies*, 2019, pp. 14–23. doi: 10.1007/978-3-319-92231-7_2.

[34] S. Fu, N. Lin, G. Zhu, and S. Jiang, "Towards Indonesian Part-of-Speech tagging: Corpus and models," *2018 International Conference on Asian Language Processing (IALP)*, vol. 1, pp. 303–307, 2018.

[35] A. Millour and K. Fort, "Unsupervised data augmentation for less-resourced languages with no standardized spelling," in *International Conference Recent Advances in Natural Language Processing, RANLP*, 2019, pp. 776–784. doi: 10.26615/978-954-452-056-4_090.

[36] G. Antipov, S. A. Berrani, N. Ruchaud, and J. L. Dugelay, "Learned vs hand-crafted features for pedestrian gender recognition," *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*, pp. 1263–1266, Oct. 2015, doi: 10.1145/2733373.2806332.

[37] P. Passban, Q. Liu, and A. Way, "Boosting neural Pos tagger for farsi using morphological information," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 16, no. 1, pp. 1–15, 2016, doi: 10.1145/2934676.

[38] M. Alfian, U. L. Yuhana, D. Siahaan, H. Munazharoh, and E. Pardede, "Handling Out-of-Vocabulary in Indonesian POS Tagging: A Comparative Study," in *2025 International Conference on Smart Computing, IoT and Machine Learning, SIML 2025*, Surakarta: Institute of Electrical and Electronics Engineers Inc., Jul. 2025, p. 1. doi: 10.1109/SIML65326.2025.11080832.

[39] Y. Kimura, T. Komamizu, and K. Hatano, "An Automatic Labeling Method for Subword-Phrase Recognition in Effective Text Classification," *Informatica (Slovenia)*, vol. 47, no. 3, 2023, doi: 10.31449/inf.v47i3.4742.

[40] A. Makazhanov and Z. Yessenbayev, "Character-based feature extraction with LSTM networks for POS-tagging task," in *Application of Information and Communication Technologies, AICT 2016 - Conference Proceedings*, 2017. doi: 10.1109/ICAICT.2016.7991654.

[41] A. Kemos, H. Adel, and H. Schütze, "Neural semi-Markov conditional random fields for robust character-based part-of-speech tagging," in *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 2019, pp. 2736–2743.

[42] P. Boonkwan and T. Supnithi, "Bidirectional deep learning of context representation for joint word segmentation and POS tagging," in *Advances in Intelligent Systems and Computing*, 2018, pp. 184–196. doi: 10.1007/978-3-319-61911-8_17.

[43] M. Pota, F. Marulli, M. Esposito, G. De Pietro, and H. Fujita, "Multilingual POS tagging by a composite Deep Architecture based on Character-Level features and on-the-fly enriched Word Embeddings," *Knowl Based Syst*, vol. 164, pp. 309–323, 2019, doi: 10.1016/j.knosys.2018.11.003.

[44] M. Alfian, U. L. Yuhana, D. Siahaan, and H. Munazharoh, "Annotation Error Detection and Correction for Indonesian POS Tagging Corpus," *Lontar Komputer : Jurnal Ilmiah Teknologi Informasi*, vol. 16, no. 1, p. 41, Jun. 2025, doi: 10.24843/lkjiti.2025.v16.i01.p04.

[45] K. Kurniawan and A. F. Aji, "Toward a standardized and more accurate Indonesian Part-of-Speech tagging," *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*, pp. 303–307, 2019, doi: 10.1109/IALP.2018.8629236.

[46] N. Reimers and I. Gurevych, "Reporting score distributions makes a difference: Performance Study of LSTM-networks for sequence tagging," in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017. doi: 10.18653/v1/d17-1035.

[47] A. Boulkroune, F. Zouari, and A. Boubellouta, "Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems," *JVC/Journal of Vibration and Control*, 2025, doi: 10.1177/10775463251320258.

[48] S. Besharati, H. Veisi, A. Darzi, and S. H. H. Saravani, "A hybrid statistical and deep learning based technique for Persian part of speech tagging," *Iran Journal of Computer Science*, vol. 4, no. 1, p. 35, 2021, doi: 10.1007/s42044-020-00063-1.

[49] N. Bölücü and B. Can, "Unsupervised joint PoS tagging and stemming for agglutinative languages," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 3, pp. 1–21, 2019, doi: 10.1145/3292398.

[50] B. Wang, A. Wang, F. Chen, Y. Wang, and C. C. J. Kuo, "Evaluating word embedding models: Methods and experimental results," 2019, *Cambridge University Press*. doi: 10.1017/ATSIP.2019.12.

[51] T. Gui, Q. Zhang, H. Huang, M. Peng, and X. Huang, "Part-of-speech tagging for twitter with adversarial neural networks," in *EMNLP 2017 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2017, pp. 2411–2420. doi: 10.18653/v1/d17-1256.

[52] L. Qu, G. Ferraro, L. Zhou, W. Hou, N. Schneider, and T. Baldwin, "Big data small data, in domain out-of domain, known word unknown word: The impact of word representations on sequence labelling tasks," in *CoNLL 2015 - 19th Conference on Computational Natural Language Learning, Proceedings*, 2015, pp. 83–93. doi: 10.18653/v1/k15-1009.

[53] J. Wulff and A. Søgaard, "Learning finite state word representations for unsupervised Twitter adaptation of POS taggers," in *ACL-IJCNLP 2015 - Workshop on Noisy User-Generated Text, WNUT 2015 - Proceedings of the Workshop*, 2015, pp. 162–166.

[54] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. Pearson Education, 2024.

Evaluating Centrality-Based Seed Node Strategies for Influence Diffusion in OSNs: A Study across SCC, WCC and Full Networks using SIR, LT and IC Diffusion Models

Halima Baabcha[1, 3*], Meriem Laifa[1, 3] and Samir Akhrouf [2,3]
[1] Department of Computer Science, University Mohammed El Bachir El Ibrahimi, Bordj Bou Arreridj , Algeria
[2] LIAM Laboratory, University Mohamed Boudiaf , M’sila, , Algeria
[3] Department of Computer Science, University Mohamed Boudiaf, M'sila, Algeria
Email: halima.baabcha@univ-bba.dz, meriem.laifa@univ-bba.dz, samir.akhrouf@univ-msila.dz
*Corresponding author

Keywords: centrality measures, influential nodes, influence diffusion, SIR, LTM, ICM, SCC, WCC and viral marketing

Received: May 11, 2025

This study investigates influence diffusion in online social networks (OSNs) through a comprehensive analysis of centrality measures and diffusion models using the Higgs Twitter dataset. We model OSNs as directed graphs, focusing on strongly connected components (SCCs) and weakly connected components (WCCs). Seven centrality measures (out-degree, in-degree, betweenness, closeness, eigenvector, PageRank, and Katz centrality) are calculated to identify key influential nodes. The top-ranked nodes are then subjected to influence diffusion simulations using three models: Linear Threshold (LT), Independent Cascade (IC), and Susceptible-Infected-Recovered (SIR) across three types of activity networks with different structural characteristics. Our findings reveal significant variations in centrality performance depending on network topology and diffusion dynamics. This methodology integrates structural network analysis with dynamic diffusion modeling to evaluate the effectiveness of influence spread. The experimental results show that out-degree and betweenness centralities are most effective for influence propagation, with the SIR model supporting sustained diffusion. The experimental results reveal that out-degree and betweenness centralities are the most effective measures for influence propagation, with out-degree being particularly impactful for initiating diffusion. The SIR model demonstrated superior efficacy for sustained influence spread, aligning more closely with real-world influence dynamics. Additionally, analyzing influence propagation within WCCs enables more computationally efficient identification of key influencers, without significant loss in accuracy. This work offers actionable insights for influence modeling and provides a practical methodology for selecting centrality measures tailored to specific diffusion scenarios. It explores the influence diffusion across different platforms, enabling researchers to assess and compare user impact by offering a detailed examination of network structures, key node significance and influence diffusion.

Povzetek: Študija na usmerjenih grafih združi več centralnostnih mer z difuzijskimi modeli kot metodo za izbiro ključnih vplivnežev in prilagojeno oceno širjenja vpliva glede na topologijo in dinamiko.

1 Introduction

In the modern digital era, OSNs have evolved into dynamic ecosystems that facilitate seamless communication, large-scale information exchange, and interactive social engagement. With billions of users actively participating daily, OSNs generate massive volumes of data, providing invaluable opportunities for researchers to investigate user behavior [1], social influence analysis [2], and the complex mechanisms of information diffusion [3]. These analyses are crucial for a variety of applications, such as enhancing public health interventions [4], mitigating the spread of misinformation [5], detecting communities [6], and optimizing viral marketing strategies [7].

Essentially, Viral Marketing (VM) offers an effective advertising strategy for commercial companies via OSNs, where companies try to promote their services and products through word-of-mouth propagation among friends or followers.

A major challenge in VM is identifying key users who wield significant influence within networks. These key nodes have a significant impact in determining the efficiency of information propagation, making them valuable for both accelerating and inhibiting the spread of messages. Various centrality metrics including degree centrality, PageRank, closeness, eigenvector, and Katz centrality are widely used to identify critical nodes that maximize influence propagation [8][9]. Each centrality measure presents unique methodologies for ranking users based on their potential influence. The efficiency of these measures depends significantly based on the nature of the interaction and the underlying network structure.

However, despite their widespread adoption, the comparative effectiveness of centrality measures remains an open research question, with diverse interpretations and applications across different network structures. This complexity arises because centrality metrics can yield differing node importance rankings, depending on network topology.

This study advances the discourse on influence maximization by providing a extensive evaluation of centrality measures in the context of OSNs. By elucidating the relationship between network structure, node importance, and information diffusion, our research lays a foundation for future investigations into evolving user behaviors, temporal dynamics, and adaptive strategies for influence propagation in ever-changing digital landscapes. Furthermore, this study serves as a valuable starting point for new researchers in this area, offering a structured framework that can be extended through the develop of hybrid centrality measures or novel diffusion models to better modeling the complexity of real-world social networks.

To provide clear objectives, this study is guided by the following research questions :

RQ1: Which centrality measures (in-degree, out-degree, betweenness, closeness, eigenvector, PageRank, and Katz) perform best for maximizing influence under different diffusion models (SIR, IC, and LT) ?

RQ2: How do structural properties of networks particularly SCCs,WCCs, and full networks affect the effectiveness of diffusion?

RQ3: How consistent are the rankings of top-k influential nodes across SCC, WCC, and full networks when evaluated with different centrality measures?

RQ4: How do the diffusion models (LT, IC, and SIR) differ in terms of coverage, speed, and sustainability of influence spread when applied to centrality-based seed selection?

The rest of this paper is organized as follows: Section 2 presents a literature review, while Section 3 introduce key definitions and models used in this study. The proposed methodology is detailed in Section 4. Results and discussion are presented in section 5. Section 6 discusses the implications of our study. Finally, this paper is concluding in Section 7.

2 Literature review

Influence Maximization (IM) is a significant problem in complex networks and a key step in solving this problem is identifying influential nodes that can maximize the information or influence propagation. Centrality measures are widely used to identify the influential nodes in complex networks. Here, we provide a brief literature review on the influential node's detection using centrality measures, and a comparison of information diffusion using different models and centrality measures as seed nodes.

For a more detailed review, we refer the reader to our previous work [10]. In particular, our earlier work presented a survey of social influence analysis in OSNs, with a particular emphasis on viral marketing applications. That study reviewed the most recent and important methods for influence modeling, influence maximization, and influential node detection. It aimed to guide novice researchers by synthesizing leading works and identifying open challenges in this area. However, was primarily conceptual and did not provide empirical validation of centrality measures or diffusion models. The present paper addresses this gap by moving beyond a survey to a systematic experimental evaluation of seven centrality measures under three diffusion models (LT, IC, and SIR), tested across SCC, WCC, and full network structures. This transition from theoretical synthesis to quantitative comparison represents a key extension of our prior work.

Authors in [11] present a study of social networks, models of OSNs and different models used for identiying influential users in OSN. They summarized the common concepts and terms used in describing OSN, such as connectivity, diameter, strongly connected components (SCC) and weakly connected components (WCC). They reviewed principal centrality measures such as in-degree, out-degree, eigenvector, closeness and betweenness centrality then highlighted the role of centrality for the detection of influential nodes. Furthermore, they discussed and analyzed different information diffusion models, such as Linear Threshold (LT), Independent cascade (IC) and Susceptible-Infected-Recovered (SIR) models with detailing their mechanisms of node activation and implications for the identification of influential nodes. Then, they reviewed some significant influential node identification techniques.

Additionally, Singh et al. [12] presented a comprehensive survey about information diffusion in social networks. They discussed various metrics and methods to analyze influence and its propagation models. They started with various concepts and centrality measures such as degree, closeness and betweenness centrality, density and degree distribution etc, which were important for understanding the structural roles of actors within social networks and their potential impact on information propagation. Further, authors discussed quantitative and qualitative analysis of social influence and influence maximization. Finally, they discussed various influence propagation models, particularly the IC, LT models, epidemic models and so on with their special variants.

Arrami et al. [13] presented a study whose primary goal was to review different works that aimed to select opinion leaders in OSNs, contributing significantly to understanding their impact on consumer behavior and marketing strategies. They explored various opinion leader detection methods, then classified them into two categories: centrality techniques and maximization techniques. Furthermore, they provided the advantages of the centrality measures resulting from the analysis of OSNs and their limits.

Another study focused specifically on identifying and analyzing the measures, approaches, and models used to measure user influence on OSNs through a comprehensive literature review [14]. Authors conducted a literature review of 25 studies published between 2014 and 2020, identifying 21 influence metrics, 4 algorithms, and 8 diffusion models. They categorized approaches into centrality-based, diffusion-based, walk-based, link-based, and popularity-based measures, and also examined Twitter-specific metrics such as Retweet Impact and FollowerRank. The study highlighted that LT/IC models and greedy algorithms are inefficient for large-scale networks, while heuristic methods improve scalability. Although this work offered a broad theoretical overview and emphasized challenges, it did not provide empirical comparisons of centrality measures. Our work complements this by experimentally evaluating seven centrality measures under three diffusion models (LT, IC, and SIR) across SCC, WCC, and full networks.

On the other hand, Singh [15] conducted a review of centrality measures applied to social network data analysis and key nodes identification. The study began with examining the fundamental notion of centrality measures with a particular attention to their roles in social network analysis. The author then provided a concise overview of computational algorithms for these measures, followed by an exploration of various research directions related to centrality metrics. Additionally, this review summarized several applications of differents centrality measures for analyzing real-world OSNs. While the study highlights the importance of centrality in ranking nodes based on their significance in various applications, it lacks a practical component in the study of influence diffusion models and the effectiveness of centrality measures in large-scale or sparse networks.

Regarding information diffusion models, Yujie [16], presented a comprehensive survey about several classic information diffusion models in OSN, like an explanatory model: the Suspected-Infected (SI) model, the Suspected-Infected-Suspected (SIS) model, and the Suspected-Infected-Recovered-Suspected (SIRS) model; predictive model: IC, LT models, emphasizing the critical role these networks play as data dissemination platforms. Then, they discussed some applications ofthose information propagation models in different social networks. Furthermore, the study highlights the practical applications of these models across various social networks, illustrating their importance in managing information propagation and mitigating misinformation.

The absence of a practical component in the study of influence diffusion models and the effectiveness of centrality measures in different network structures presents significant weaknesses. Additionally, the relationship between centrality measures and information diffusion models is not adequately specified, potentially leading to confusion about how these approaches can complement each other in identifying effective influencers. Furthermore, these works do not clarify the implications of different network structures on the applicability of these models, which could impact the accuracy of influencer identification and engagement strategies.

Existing studies often focus on either (i) ranking influential nodes using centrality measures without validating their actual spreading potential [11][14] or (ii) analyzing information diffusion without systematically comparing different seed selection strategies [17]. Additionally, many evaluations have been conducted on one type of network structure without considering variations in SCC and WCC. To address these gaps, this study provides a comprehensive evaluation of centrality measures as a seed selection method to maximize influence diffusion. We analyze how different centrality-based strategies perform in replies, mentions, and retweet networks and compare their effectiveness in maximizing influence spread under IC, LT, and SIR models. By combining structural network analysis with diffusion simulations, this study offers valuable insights into the relative strengths and weaknesses of different centrality measures in OSN.

3 Background

3.1 Definitions

An OSN can be modeled as a weighted graph $G = (V, E, w)$, where V is a set of nodes $(v_1, v_2, ..., v_i)$ representing users, E is a set of edges $(e_1, e_2, ..., e_j)$ representing social interactions, and w denotes the edge weights. In this network representation, V corresponds to actors (individuals), while E captures the relationships between them, such as replies, mentions, and retweets. For instance, if an edge e_j exists between nodes v_m and v_n, it signifies a connection or association between the corresponding actors.

The primary goal of this study is to analyze information diffusion in OSNs. To achieve this, we examined various types of graphs, including SCC, WCC, and complete complex networks. Addressing this issue requires reviewing several definitions of influence metrics within OSNs. This study specifically focuses on the following definitions:

3.1.1 Influential users

Top-k users are the k users that are capable of generating maximum influence and widest information propagation to their connected users in an OSN, and can be characterized as a person who has the power to affect people, actions, or events. The top k nodes in an OSN are crucial for understanding the structure and evolution of the network and can be used to detect influential users, optimize information diffusion, and improve the overall performance of the network [18].

3.1.2 Strongly connected components (SCC)

This is a maximal subset of nodes in a directed graph such that every node is reachable from every other node within that subset. This means that for any two nodes v_i and v_j in the SCC, there exists a directed path from v_j to v_i and a directed path from v_i to v_j. SCCs are crucial for understanding the structure and connectivity of directed graphs, with applications in several fields such as OSN analysis [19].

3.1.3 Weakly connected components (WCC)

Is a maximal set of nodes $C\subseteq VC$ such that for every pair of nodes there is a path between v_i and v_j if the edge direction is ignored. Equivalently, if a directed graph is transformed into an undirected graph by treating every directed edge as an undirected edge, the connected components of this undirected version are WCCs [20].

3.2 Centrality measures

In this section, the most important metrics are introduced. In graph theory, centrality is defined as a measure of the importance of a given node in a graph. The problem of finding the most influential users in OSNs is, in the end, a measure of importance. Centrality has attracted the most attention in the early years of influencer identification [21]. For clarity, all the abbreviations used in this paper are listed in Table 1.

3.2.1 Degree centrality

This ranks users with more connections higher in terms of centrality; in other words, it measures the total number of connections a user has with other users. However, it does not indicate the frequency of communication and is often a local maximum for network measures [22]. In an undirected graph, the degree centrality C_d for user v_i in an undirected graph is:

$$C_d\,(v_i) = d_i \qquad (1)$$

Where d_i is the degree (number of adjacent edges) of node v_i. In a directed graph, there are two types of degree centrality.

In-degree centrality: It refers to the number of edges pointing inwards at a node, usually refers to the popularity of a user [22]. We formulate:

$$C_d\,(v_i) = d_i^{in} \qquad (2)$$

Out-degree centrality: The out-degree of a node refers to the number of edges that lead it out. We formulate:

$$C_d\,(v_i) = d_i^{out} \qquad (3)$$

And consider users with more connections to be more important users [23]. Users with a large number of connections are linked to other users in the graph. Such a node is considered important if it has many neighbors because it has a higher likelihood of intercepting or capturing whatever flows are resources or information through the network [22].

3.2.2 Betweenness centrality

Betweenness centrality [22] indicates the capability of a user for faster transfer of information through the graph. As is the case for edges with high betweenness, users with high betweenness occupy critical positions in the graph structure and are therefore able to play critical roles [24]. This is often enabled by the large amount of flow carried by users that occupy a position at the interface of tightly knit groups. Betweenness centrality measures the amount of network flow that a given node controls, while this measure gives the volume by traffic that passes through a node in the network [23] [25]. However, this metric can measure the possibility of infection by the traffic that passes through a node and controls the diffusion of infection to other nodes in the network. The betweenness centrality of node v is given by :

$$B(v)=\sum_{u\neq v\neq w}\frac{\sigma_{u,w}\,(v)}{\sigma_{u,w}} \qquad (4)$$

Where $\sigma u,\ w\ (v)$ refers to the total number of shortest paths connecting u and w that pass-through v and $\sigma u,\ w$ is the total number of shortest paths from u to w.

3.2.3 Closeness centrality

This is a measure of the importance of a node in the graph. It captures the average distance between one node and every other node in a graph. The fact behind Closeness centrality indicates that the more central the user, the faster it can reach other nodes in the graph [26]. A user with a higher closeness centrality is more prominent within the network, as they can reach all other nodes more quickly. This metric is computed using the following formula:

$$C_c\,(u) = (N - 1) \,/\, \Sigma\ d\,(u, v) \qquad (5)$$

Where $d(u,\ v)$ is the shortest path distance between the nodes u and v. In closeness centrality, the intuition is that the more central the nodes, the more quickly they can reach other nodes. Formally, these nodes should have a smaller average shortest path length than other nodes.

3.2.4 Eigenvector centrality

Is a measure of a node’s influence in a network, based on the idea that a node’s importance depends on its neighbors’ importance. It is defined for both directed and undirected graphs [27]. This measure is calculated using the following equation:

$$C_e\,(v_i) = \frac{1}{\lambda}\sum_{j=1}^{n} A_{j,i}C_e(v_j) \qquad (6)$$

We can use the adjacency matrix A of the graph. Let $Ce\ (v_i)$ denote the eigenvector centrality of node v_i. We want the centrality of v_i to be a function of its neighbors’ centralities. It is posited that this is proportional to the summation of the centralities. In the adjacency matrix A of the network, Ai,j indicates the connection (and possibly the strength of that connection) between nodes i and j, and λ is a constant (the leading eigenvalue of A). Nodes connected to other highly central nodes received higher scores. Eigenvector centrality captures the number of connections a node has and crucially, the influence of these connections. This principle underlies other centrality measures such as PageRank, which adopts the same concept for ranking web pages [25]. Previous research has used eigenvector centrality to detect influential nodes in a graph [28][29].

3.2.5 Katz centrality

Calculates the influence of users (nodes) by considering all network paths. This is a measure of node influence in a network that extends the concept of eigenvector centrality by incorporating both the number of immediate neighbors and the importance of more distant neighbors. Unlike pure eigenvector centrality, it assigns a small amount of centrality to each node as a baseline, and then propagates influence through the network; attenuating contributions by a factor of α (often called the attenuation or damping factor). The Katz centrality considers all network links [30]. The main difference between Katz and closeness centralities is that the former assigns a certain minimum score to every user in the network [31]. While, Katz has high computational complexity of centrality, which limits its application to large networks [30].

Formally, for a node *i*, its Katz centrality *xi* is given by:

$$C_{katz}(v_i) = \alpha \sum_{j=1}^{n} A_{j,i} C_{katz}(v_j) + \beta \qquad (7)$$

Where:

- *Ai,j* is the *(i,j)* entry of the adjacency matrix *A*.
- α is a constant selected such that $\alpha < \frac{1}{\lambda_{\max}}$, where λ *max* is the largest eigenvalue of *A*. This condition ensures convergence.
- β is a constant "bias" or the baseline score received by each node.

In essence, Katz centrality not only counts direct connections but also indirect connections of every length, weighting longer paths by decreasing the power of α. A node is considered more central if it has many neighbors and/or it is connected to other central nodes [23].

3.2.6 Page rank centrality

Is an algorithm proposed by Page and Brin, the co-founders of Google, to rank web pages [32]. It is a widely known measure for ranking web pages based on their importance in the web, which is considered as a graph. It counts the quality of the links to a node to determine the prominence of the node. This measure can be calculated using the following equation.

$$PR(i) = \frac{(1-d)}{n} + d * \sum_{j\in\Gamma(i)} \frac{PR(j)}{C(j)} \qquad (8)$$

Where *PR(i)* and *PR(j)* represent the PageRank values of nodes *i* and *j*, C(j) represents the out degree of the node, *d* is the damping factor (usually 0.85), *n* is the total number of web pages, and *Γ(i)* is the set of neighboring nodes of node *i* [33].

3.3 Information diffusion models

Information diffusion models are mathematical and computational frameworks used to describe and analyze how information, ideas, behaviors, or innovations spread through a network of individuals or entities. These models are widely used in OSNs analysis, epidemiology, marketing, and communication studies to understand the mechanisms governing influence propagation. Typically, these models consider factors such as influence probability, network structure, and individual adoption thresholds. The most common categories include the IC, LT, and SIR models. These models can help researchers and practitioners predict trends, optimize viral marketing campaigns, and mitigate misinformation spread in networks [34][35]. Information diffusion models typically involve three primary components: the sender, who initiates the process; the receiver, who receives the information; and the medium, which facilitates the transmission of information. The diffusion process can be influenced by differents parameters such as user characteristics, network structure, and external factors, which can affect the spread of information [36]. For more details on this, check [17]. An overview of these models is presented in the reminder of this section.

3.3.1 Susceptible-infected-recovered (SIR) model

Is a well-known approache for evaluating centrality measures in OSNs analysis. The epidemiological model simulates the diffusion of a virus within a network, classifying nodes into three distinct categories [10] (as illustrated in Figure 1):

- *Susceptible (S) nodes: I*s unaware of the information diffusion within the graph. These nodes are uninfected but not immune, and can be contaminated by neighboring infected nodes. Initially, all nodes were considered susceptible, except for the source node.
- *Infected (I) nodes*: An infected node has acquired and is aware of the information spreading through the network, and it actively shares this information with its neighboring nodes. After a specific time period, the infected node moves to the recovered state, with this transition governed by the infection probability β at each time step and the recovery probability λdt over a time interval dt. The average duration a node remains infected is denoted by *D*.
- *Recovered (R) nodes:* The recovered nodes lose interest in the information and no longer spread it. They also become immune to further infection. At the end of the process, only susceptible and recovered nodes remain in the network.

The dynamics of the *SIR* model are governed by a set of ordinary differential equations that describe the transitions between these states. The total population in the network is denoted by *N*, and at any time *t*, the sum of susceptible, infected, and recovered nodes equals N. The SIR system can be described using the following ordinary differential equations:

$$\frac{dS}{dt} = \frac{\beta IS}{N}$$

$$\frac{dI}{dt} = \frac{\beta IS}{N - \lambda I} \qquad \mathbf{1.1}$$

$$\frac{dR}{dt} = \lambda I$$

$$S(t) + I(t) + R(t) = N \qquad (9)$$

The SIR model extends beyond traditional epidemiological contexts to the analysis of the information spread in social networks, providing insights into the influence exerted by various nodes based on their centrality measures [23].

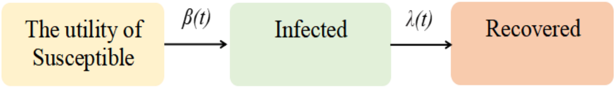


Figure 1: Illustration of SIR model.

3.3.2 Linear threshold model (LTM)

In this model, every edge e(u, v) carries a weight W(u, v), with the total weight of all incoming edges to node v being less than or equal to 1, and each node *v* is also associated with a threshold θ_v. The LT model starts with some active nodes, with all other inactive nodes, and a random choice of thresholds *θ.* The LTM samples the value of *v* for each user *v* uniformly at a random probability from [0,1]. At step 0, nodes in the seed set S are marked as active, while all other nodes are initialized as inactive. The model then proceeds iteratively, updating each node's status. In step *t*, all nodes that were active in step *t-1* stay active, and any inactive node *v* at step *t -1* may become active. The influence spread of the seed set *S* under the LT model, denoted as *σ(S)*, represents the expected number of nodes that become active when *S* is activated initially. For more details on this model, check [10] [37].

3.3.3 Independent cascade model (ICM)

In ICM, a probability *p (u,v)* is associated with each edge *e(u,v)* where *u* and *v* are two nodes in the graph. *p (u,v)* is the probability that *u* succeeds in activating *v*. In this model, node *v* is independently activated by each of its incoming neighbors by introducing an influence probability *p (u,v)* to each edge *e(u,v)*. Based on the influence probabilities and given a seed set S at time step 0, a diffusion instance of the IC model unfolds in discrete steps. Each active node *u* at step *t* activates each of its outgoing neighbors *v* which is inactive in step *t-1* with probability *p(u,v)*. The activation process can be considered as flipping a coin with head probability *p (u,v)*: If the result is heads, then *v* is activated; otherwise, *v* remains inactive. The diffusion instance terminates when no additional nodes are activated. The influence spread of seed set *S* under IC is the expected number of activated nodes when *S* is the initial active node set and the above stochastic activation process is applied. For more details on this model, check [10] [38].

4 Methodology

Our methodology enables an in-depth analysis of how centrality-based node selection affects influence spread across networks. By systematically partitioning the network into its WCC and SCC, we can assess how structural properties impact influence propagation. As illustrated in Figure 2, the proposed methodology consists of three main steps.

Table 1: Abbreviations

Node Id	Value
E	Set of edges between users
V	Set of users in the graph
w	Weights associated with edge in E
Indg	In-degree centrality
Outdg	Out-degree centrality
Cl	Closeness centrality
Bt	Betweenness centrality
Ev	Eigenvector centrality
Pr	Pagerank
Kz	Katz centrality
WCC	Weakly connected components
SCC	Strongly connected components
SIR	Suspected - Infected - Recovered
LTM	Linear threshold model
ICM	Independent cascade model
S	Seed nodes
G (V, E)	Graph
N	Total number of nodes in the graph

4.1 Step 1: Initialization and SCC, WCC identification

In this initial step, we decomposed the graph based on the WCC and SCC for each network. For SCC, we used Tarjan's [39][40] algorithm which is a Depth-First Search (DFS)-based approach used to find SCCs in a directed graph. It was introduced by Robert Tarjan in 1972 [41] and is known for its efficiency with a time complexity of *O (V+E)*, where *V* is the number of vertices, and *E* is the number of edges. The algorithm is based on DFS traversal and utilizes a low link value to track the smallest reachable node from a given node. It employs a stack-based approach to store nodes in the current SCC and detects cycles and back edges, which helps to identify SCCs efficiently [42] [43]. Detecting SCCs in a directed graph is essential for understanding information diffusion, as SCCs represent regions where information can circulate freely among nodes.

One of the most significant aspects of SCC detection is its ability to identify self-contained clusters, in which nodes mutually reinforce the information propagation. For example, in OSNs, a strongly connected community ensures that a message (e.g., news, a viral post, or misinformation) continues circulating within the group, making it highly resilient to external removal [44]. Conversely, nodes outside an SCC might receive information, but do not contribute to its continued spread. This is particularly relevant for influence maximization [45][46][47], where marketers, advertisers, or political campaigns seek to target key SCCs to ensure that information cascades efficiently within a community before attempting to expand outward [34].

For efficient WCC detection, we are used the Breadth-First Search (BFS) method [48], which allows the analysis of how information can spread across different parts of a network even when direct connections are absent. It is an effective method for detecting WCCs in directed graphs. To find WCCs using BFS, the directed graph is first treated as undirected by considering all edges as bidirectional. The algorithm then initializes a set of unvisited nodes and iterates through each node in the network. Each unvisited node, BFS is performed to explore all reachable nodes by systematically traversing neighbors, marking them as visited, and grouping them into a component. This process is repeated until all nodes are visited, resulting in the identification of all WCCs in the graph. The BFS approach operates with a time complexity of $O(V + E)$, making it suitable for large-scale networks. For instance, on OSNs, WCCs can reveal clusters of users who, although not directly connected, can still influence each other through intermediaries.

This understanding is crucial for modeling diffusion processes, designing effective communication strategies, and controlling the spread of information or misinformation. Studies have shown that the structure of WCCs significantly affects the efficiency and reach of network information dissemination. [49] Analyzed influnece diffusion and consensus dynamics in weakly connected component in directed graphs and provided insights into how information spreads in such structures. Therefore, analyzing WCCs provides valuable insights into the structural mechanisms that underpin information diffusion in directed graphs.

4.2 Step 2: Influence calculation and top-k nodes selection

Once the network components (SCCs, WCCs, and the full network) are identified, we determine the top ten most influential seed nodes using multiple centrality measures. These measures have been extensively cited in the literature as strong indicators of influence maximization. We used seven centrality measures: in-degree, out-degree, betweenness, closeness, pagerank, eigenvector, and Katz centrality, each highlighting the different structural aspects of a node's influence.

For each network structure type (SCC, WCC, and full network), we identified nodes with the highest centrality values to serve as seed nodes in the influence diffusion models (SIR, LT, and IC). Consider (S_{Indg}), (S_{Outdg}), (S_{Cl}), (S_{Bt}), (S_{Ev}), (S_{Pr}), and (S_{Kz}) represent nodes with the top centrality values. For the selection of seed nodes, we studied the three networks separately for each type of structure (SCC, WCC, and the full network). By selecting the top central nodes in the SCCs, WCCs, and the full network, We selected the top-10 nodes (k = 10) as seeds, following common practice in influence maximization research where small, fixed seed sets are widely adopted to balance diffusion effectiveness with computational feasibility. In addition, in viral marketing contexts, companies typically target only a limited number of highly influential users due to budgetary and operational costs, since engaging fewer but strategically chosen seeds is more cost-effective and still sufficient to trigger large-scale diffusion. Therefore, our choice of k = 10 aligns with both prior studies and practical considerations [50][51][51][53]and centrality value were used in their raw form for ranking nodes. Since each centrality measure was applied independently, only the relative order of nodes was necessary for seed selection, and no normalization was performed. we performed a comparative analysis of the influence spread in different network structures.

4.3 Step 3: Influence diffusion simulation

To evaluate the effectiveness of centrality-based seed selection in influence propagation, we utilized the Higgs Twitter dataset[1], which captures user interactions surrounding the discovery of a Higgs boson-like particle on July 4, 2012. The dataset includes retweets, replies, mentions between July 1 and July 7, 2012, and consists of three directional networks representing different types of user activities, with anonymized user IDs maintained across all layers. This structure enables large-scale network studies, where one layer represents the social structure while the other three encode distinct user dynamics. We focused on the Higgs Twitter dataset, but it's important to mention that the dataset contains three separate graph representations (retweet networks, reply networks, and mention networks). Each of the three graphs has its own structural characteristics, which include size, density and degree-distribution. Thus, it also provides the different testbeds in a single dataset and allows us to evaluate the performance of the centrality measures and diffusion models on different networks. Table 2 outlines the dataset's characteristics. The study was implemented using Google Collaboratory, leveraging libraries such as NetworkX, iGraph, NumPy, Pandas, Matplotlib, and NDlib for the propagation process.

To simulate information diffusion using the selected seed nodes across the three types of networks, we employed the three most widely used diffusion models [54]: LT, IC, and SIR. Each model represents a distinct mechanism by which individuals adopt and spread information, ensuring more comprehensive understanding diffusion of influence. These models are particularly relevant for viral marketing analysis, as they capture different dynamics of information spread. The selected seed nodes were designated as source nodes in these models.

The diffusion process was analyzed separately for each types of network, allowing us to compare the influence maximization strategies across different network topologies. Key performance metrics include the final reachability (number of influenced nodes), speed of information propagation, and influence retention within the network substructures. Influence spread was measured over the course of the diffusion process and summarized at its termination. For the LT and IC models, we recorded both the number of activated nodes and the number of inactivated nodes across time steps, reporting the final state once propagation completed. For the SIR model, we tracked the temporal evolution of susceptible, infected, and recovered nodes, with the final counts at the last time step serving as the evaluation metrics. For each centrality measure and diffusion model, we performed 50 independent runs, and the reported results correspond to the averages across these runs. Tables (6-8) present the average influence spread measured under each diffusion model for each network structure, with values representing the final node states after diffusion terminates.

To simulate the spread of influence among selected key users identified using centrality measures, the parameter values for each model were carefully chosen to ensure a realistic representation of information propagation. For the SIR model, the infection probability (β) was set to 0.5, and the recovery rate (γ) to 0.05, simulating an epidemic-like diffusion process where nodes stop transmitting influence once they recover [55][56][57]. This configuration reflects short-term influence dynamics, where information spread is temporary.

The LT model was configured with a threshold of 0.1, meaning that at least 10% of a node's neighbors must be active before the node itself becomes active. This choice represents a gradual accumulation of influence, where activation requires reinforcement from multiple sources, making it suitable for modeling social behaviors like opinion formation and product adoption [58]. For the IC model, an infection probability of 0.9 was used to simulate a highly influential propagation process, where selected influencers have a strong likelihood of activating their neighbors. Since IC model lacks a recovery mechanism, this setup is ideal for modeling permanent adoption scenarios, such as viral marketing. To ensure consistency across the models, we conducted 50 iterations for each simulation, allowing for a comprehensive assessment of influence diffusion patterns under different network structures. These configurations enabled a comparative analysis of how centrality-based influencers drive information spread across diverse diffusion models. This comprehensive evaluation helps to determine which model best utilizes the network structure for optimal influence dissemination. By integrating SCC and WCC identification, top-k seed nodes selection, and multi model diffusion simulation, this approach provides a robust framework for analyzing and optimizing the information propagation.

1.2 5 Results and discussion

5.1 Results

Our experiments' results are categorized into two major sections: (i) the ranking of the most influential nodes using centrality measures in various network structures (Tables 3-5), and (ii) the diffusion performance of these measures in three different diffusion models (Tables 6-8).

The top-10 nodes for each centrality measure for the reply, mention, and retweet networks are presented in Tables (3-5), respectively, and detecting equivalence across the Full, SCC, and WCC structures demonstrates how node rankings can vary with network representation.

The influence spread results using the SIR, LT, and IC models for the same networks are summarized in Tables (6-8), respectively. These tables show the average number of activated nodes produced by different centrality measures, across the Full, SCC, and WCC structures, which allows for a direct comparison of their effectiveness at driving diffusion. To ensure the robustness of our findings, all results were averaged over 50 independent runs for each diffusion model and centrality measure [59][60].

Table 3, presents the top ten key users selected by each centrality in the different graphs structures on Twitter reply dataset. The equivalence of the top ten nodes between the Full, SCC, and WCC networks shows that certain measures remain highly stable, while others vary significantly. Eigenvector centrality was the most consistent, with 100% equivalence in WCC and full network and 90% equivalence in SCC and WCC/full network, meaning that all the top ten nodes remain the same in both networks. Similarly, in-degree, eigenvector, betweenness, pagerank, and Katz exhibit 100% equivalence in the WCC and full network, indicating that the core influential nodes remained unchanged regardless of the inclusion of WCCs. Eigenvector centrality remains stable because it identifies nodes that are connected to other influential nodes, making their rankings less sensitive to changes in network structure. Similarly, in-degree, betweenness, and pagerank rely on direct connections or shortest paths, which remain largely unchanged between WCC and the full network. This confirms that WCC retains the essential structure of the network, meaning additional weakly connected nodes do not significantly impact rankings.

The equivalence between SCC and WCC/Full is low in most centrality measures, with in-degree, closeness, pagerank, and Katz at 10%, out-degree and betweenness at 40%. This suggests that SCC functions as a structurally independent core, and expanding to the full network significantly changes the role of key nodes. The low equivalence percentages indicate that central nodes in SCC are not necessarily influential in the broader network. The higher out-degree equivalence (40%) suggests that some information-spreading nodes retain their importance, but in general, SCC nodes play distinct roles compared to their WCC/full network.

Similarly, Table 4 shows that the most consistent measures eigenvector, betweenness, pagerank, and Katz show 100% equivalence in the full network and WCC, meaning all top ten nodes remained the same. This occurs because these measures account for both direct and indirect influence, and the removal of WCC does not significantly affect major hubs.

Similarly, out-degree centrality exhibits 100% equivalence between the WCC and full networks, confirming that the core influential nodes remain unchanged regardless of the inclusion of WCC. Since WCC retains all major hubs, the rankings of the most central nodes remain consistent.

In contrast, the equivalence between SCC and WCC/full varies: it is high for betweenness (90%), eigenvector and closeness (80%), out-degree (90%) and lower for Katz and pagerank (40%-60%). This suggests that the SCC functions as an independent structural core, and expanding to the full network significantly alters the role of key nodes. The lower equivalence for Katz and pagerank indicates that the introduction of weakly connected nodes redistributes influence across the network. While the SCC captures a dense, highly connected subset, the WCC and full network introduce additional connections that shift centrality rankings.

Finally, Table 5 shows that eigenvector centrality showed 100% equivalence across networks. This stability occurs because eigenvector centrality is based on recursive influence meaning that if a node is central in WCC, it remains central when weakly connected nodes are added to form the full network. Additionally, other measures such as out-degree, betweenness, in-degree, and Katz also exhibit stability between WCC and the full network. This suggests that the core influential nodes remain largely unchanged regardless of whether WCC are included.

The equivalence between SCC and WCC/full networks is low (40%) for in-degree and Katz Centrality, indicating that SCC functions as a structurally independent core. This happens because SCC consists of a highly interconnected subset of nodes that operate differently from the full and WCC networks. The low equivalence (40%) means that when the network expands beyond SCC, the importance of key nodes shifts significantly.

This suggests that SCC acts as an independent core structure where central nodes play different roles compared to the broader network.

The SIR model reveals that out-degree, betweenness and pagerank outperform other measures in facilitating the spread. Despite its theoretical efficiency [61], closeness centrality results in minimal diffusion. This occurs because out-degree centrality identifies nodes with the most direct connections, allowing them to infect many others immediately, making it highly effective for rapid diffusion. Katz centrality also performs well as it considers both direct and indirect influence, making it a strong measure for long-range impact. However, closeness centrality, which prioritizes nodes with the shortest paths to all others, is ineffective in sparse networks, where long distances between nodes hinder rapid information propagation; it is worth noting that the performance of closeness centrality can vary depending on the specific network and the parameters used in the SIR model [62].

Similarly, in the LTM, out-degree centrality leads to the highest activation in full and WCC network, followed by Indg centrality and Bt centrality in SCC network, while the other centrality performs poorly. The LTM activates nodes based on the influence of their neighbors, which explains why out-degree centrality is the most effective nodes with many outgoing links can spread influence more easily. In-degree centrality also contributes because nodes that receive influence from many sources are more likely to activate.

Finally, the IC model produces even more restrictive results, yet out-degree centrality remains the most effective. The IC mode introduces probabilistic activation, meaning nodes influence their neighbors with a certain probability. Since nodes with high out-degree have more neighbors, they increase the chances of spreading activation, even in a probabilistic setting. (See Table 6, Table 7, and Table 8 for detailed results).

5.2 Discussion

From the experimental results, it can be seen that out-degree and betweenness centralities are the most effective measures for influence propagation. Out-degree centrality, in particular, is strong in initiating the diffusion process due to its capacity to activate many neighbors [62]. This is because out-degree centrality measures the number of edges originating from a node, indicating its potential to spread influence to other nodes in the network [63]. Betweenness centrality, on the other hand, plays a critical role in sustaining influence over time by bridging different parts of the network. This is because betweenness centrality measures the fraction of shortest paths between all node pairs that pass-through a given node, indicating its ability to connect different parts of the network and facilitate the spread of influence [64][65].

The SIR model demonstrated superior efficacy for sustained diffusion, reflecting a more realistic simulation of influence propagation compared to LT and IC models, which are more appropriate for large networks into account the influence propagation over time and the role of network structure in facilitating or hindering this process [14][66][67].

Overall, our reserch draw attentions to the necessity of considering network structure and the dynamics of

Influence propagation when designing marketing campaigns or evaluating the influence value of a node in a network. The most well-known and frequently utilized metrics in social network analysis are centrality measures, nevertheless, their application in the identification of the influencers depends on the network's characteristics. A key contribution of this study lies in demonstrating that SCC/WCC decomposition yields new insights that are not captured when only the full network is analyzed. WCC analysis provides computational efficiency without significantly altering results, making it a practical alternative for large-scale networks. SCC analysis, on the other hand, reveals structurally distinct influencer roles, which are obscured at the full-network level. This dual perspective extends state-of-the-art centrality-based influence maximization approaches by linking performance not only to the choice of centrality measure and diffusion model but also to the underlying structural representation of the network.

Unlike prior influence maximization studies that apply centrality measures directly on full networks [68][69], or analyze these measures only at a theoretical level without specification of network structure [11][12][14][15], our work introduces a decomposition-based evaluation across SCCs and WCCs. This design provides two key insights: (i) performance differences of centrality measures are topology-dependent, with measures such as out-degree dominating in SCCs while betweenness gains importance in WCCs, and (ii) WCC-based analysis offers significant computational efficiency without substantially sacrificing diffusion performance. These insights, which are not captured when analyzing only full networks, highlight the value of incorporating SCC/WCC decomposition in centrality-based influence maximization.

Our approach combining multi-model diffusion, centrality comparison, and structural analysis (SCC/WCC/Full) offers a more complete understanding of influence propagation. These findings provide a foundation for designing hybrid influence maximization frameworks and strategies that are both effective and computationally efficient.

Table 2: Summary statistics of used datasets.

	Full graph		**SCC graph**		**WCC graph**	
	Nodes	**Egdes**	**Nodes**	**Egdes**	**Nodes**	**Edges**
Reply	3891	32523	322	708	12839	14944
Mention	116408	150818	1801	7069	91606	132068
Retweet	256491	328132	984	3850	223833	308596

6 Implications

The findings of this study provide valuable theoretical and practical implications for OSN analysis, influence maximization, and information diffusion strategies. By systematically comparing different centrality measures across various network structures and diffusion models, this study offers a comprehensive perspective on how influence propagates in OSNs. The integration of structural network considerations with diffusion modeling contributes to a more robust methodology for influence analysis.

From a practical perspective, social media analysts and digital marketers can apply these insights to refine influence-spreading strategies. Identifying key seed users based on diffusion models that align with specific campaign objectives can enhance brand visibility and engagement in viral marketing. A notable contribution of this study is its focus on network component structures, particularly the role of SCCs and WCCs in shaping diffusion outcomes. Unlike conventional approaches that treat networks as uniform entities, our findings demonstrate that strategies targeting SCCs can improve local influence retention, whereas those focusing on WCCs enable broader dissemination across loosely connected communities. These insights can inform the development of more effective influence maximization algorithms that integrate both local and global centrality measures.

Another important finding from our comparative analysis is the necessity of selecting diffusion models that best suit the network's characteristics and the intended spread of information. This study is among the first to provide a detailed, quantitative evaluation of how local centrality measures, such as out-degree centrality, play a dominant role in the LT model, whereas global measures like pagerank, and betweenness centrality become more relevant in the SIR model. Additionally, out-degree centrality emerges as the most effective measure for initiating the diffusion process due to its direct quantification of an individual’s capacity to spread information. Conversely, betweenness centrality plays a crucial role in sustaining influence by acting as a bridge between different network segments.

Among the diffusion models assessed, the SIR model is particularly effective due to its balanced treatment of infection and recovery, supporting sustained diffusion compared to LT or IC models. This makes it well-suited for simulating real-world information dissemination where messages need to spread continuously until fading.

Finally, this study highlights the practical utility of WCC-based analysis as a computationally efficient alternative to full-network evaluation. Since centrality rankings for measures such as in-degree, PageRank, Katz, and eigenvector remain stable in WCC, analyzing the full network is often unnecessary. WCC retains the essential structural features while excluding weakly or entirely disconnected nodes, leading to reduced computational complexity without compromising accuracy. This refinement is particularly valuable for large-scale OSN applications where rapid decision-making is critical for example, identifying and activating key Twitter accounts during the first minutes of a breaking news event, selecting top YouTube influencers for short-lived marketing campaigns, or monitoring Facebook groups during emergencies to support timely information dissemination.

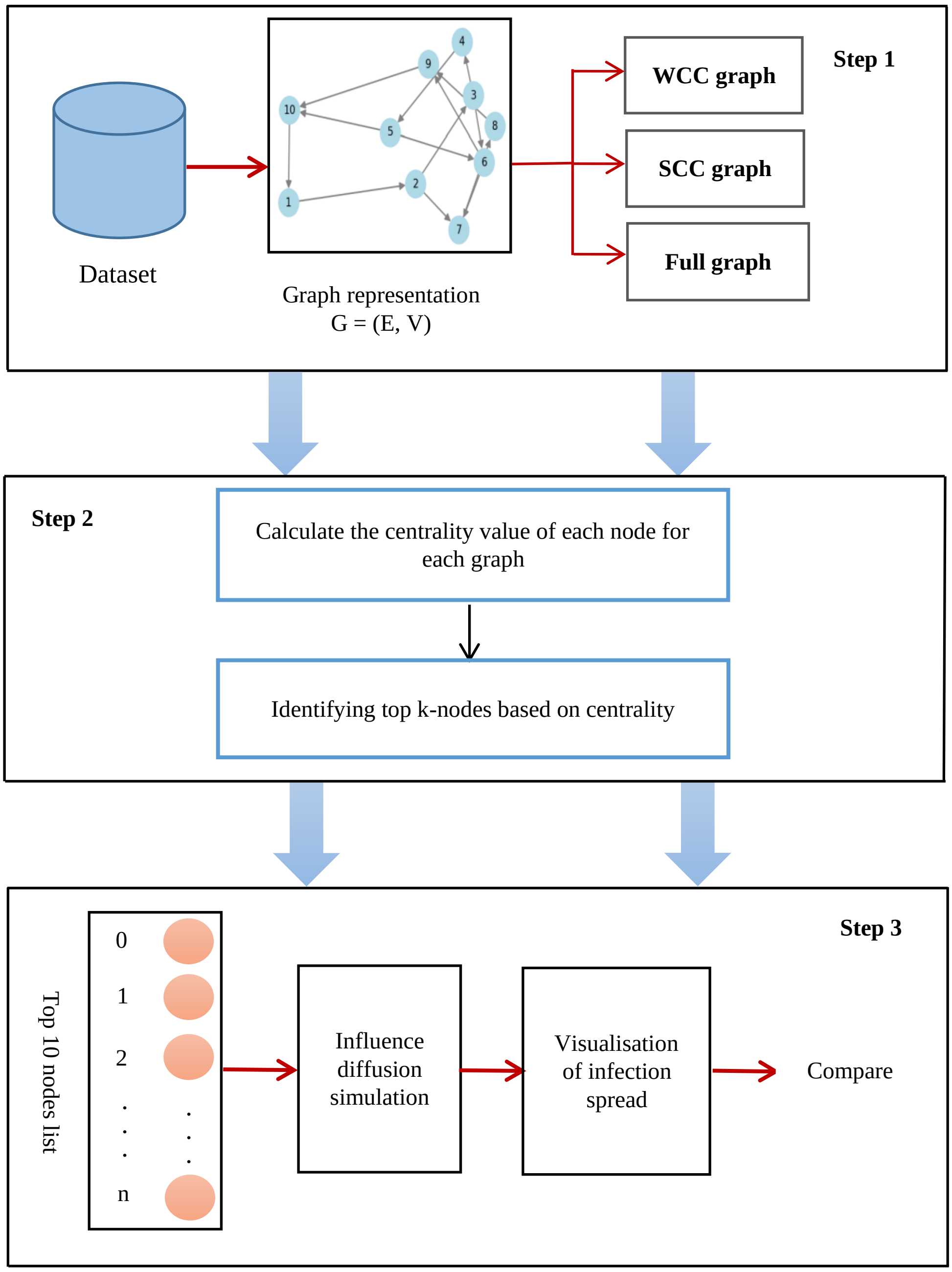

Step 1
Dataset
10
9
4
3
8
5
6
1
2
7
Graph representation
G = (E, V)
WCC graph
SCC graph
Full graph
Step 2
Calculate the centrality value of each node for each graph
Identifying top k-nodes based on centrality
Step 3
Top 10 nodes list
0
1
2
n
Influence diffusion simulation
Visualisation of infection spread
Compare

Figure 2 : Workflow of the methodology

Table 3: Top 10 key users identified by various centrality measures in reply dataset.

	Ranking	Indg		Outdg		Ev		Bt		Cl		Pr		Kz	
		Node Id	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**
SCC network	1	677	23	9021	24	9021	1.0000	67382	44556.25	67382	0.3116	677	0.0438	9021	0.0068
	2	9021	21	67382	15	50218	0.5113	35376	32084.41	98204	0.2833	9021	0.0215	677	0.0065
	3	67382	14	52908	14	50244	0.4649	12751	29038.85	13808	0.2803	67382	0.0186	52908	0.0052
	4	6241	13	6241	13	8855	0.4465	9021	27318.42	35376	0.2755	36436	0.0182	6241	0.0052
	5	13808	13	98204	10	80429	0.3863	13808	27098.21	12751	0.2750	13808	0.0180	13808	0.0050
	6	52908	12	9964	9	80426	0.3552	6940	23586.11	677	0.2736	52908	0.0154	67382	0.0050
	7	98204	11	33833	9	9036	0.3157	69883	21405.54	6241	0.2677	6241	0.0136	12751	0.0047
	8	12751	11	12751	8	8989	0.2996	69891	21252.83	214092	0.2618	5137	0.0128	3604	0.0046
	9	3604	9	42172	8	71888	0.2989	6241	20867.89	89805	0.2603	3604	0.0122	9964	0.0045
	10	9964	8	3604	1	50277	0.2983	98204	19752.06	42172	0.2597	12751	0.0109	5137	0.0044
WCC network	1	677	1206	9021	35	9021	1.0000	13808	1.4e+06	88	0.2462	677	0.0648	677	0.0078
	2	88	1071	16695	33	50218	0.5113	677	1.2e+06	677	0.2338	88	0.0254	88	0.0067
	3	220	470	433454	32	50244	0.4649	36436	1.2e+06	27311	0.2249	10836	0.0122	220	0.0032
	4	3549	218	6241	26	8855	0.4465	67382	1.2e+06	76803	0.2221	220	0.0110	3549	0.0014
	5	317	168	113517	25	80429	0.3863	52908	1.2e+06	96775	0.2204	10844	0.0104	317	0.0011
	6	349	142	67382	24	80426	0.3552	42177	1.1e+06	3998	0.2172	10867	0.0079	349	0.0008
	7	1988	105	52908	20	9036	0.3157	52882	1.1e+06	98204	0.2171	201222	0.0079	3369	0.0007
	8	7690	96	72466	18	8989	0.2996	201222	1.1e+06	103986	0.2170	207364	0.0079	7690	0.0007
	9	3369	92	269152	17	103181	0.2989	12751	6.9e+05	163525	0.2167	152385	0.0079	1988	0.0006
	10	16460	79	20971	17	71888	0.2989	6940	6.4e+05	184519	0.2161	237807	0.0079	16460	0.0006
Full network	1	677	1206	9021	35	9021	1.0000	13808	1.4e+06	124554	1.000	677	0.0243	677	0.0026
	2	88	1071	16695	33	50218	0.5113	677	1.2e+06	286277	1.000	88	0.0095	88	0.0022
	3	220	470	433454	32	50444	0.4649	36436	1.2e+06	274148	1.000	10836	0.0046	220	0.0010
	4	3549	218	359985	31	8855	0.4465	67382	1.2e+06	274149	1.000	220	0.0041	3549	0.0004
	5	317	168	6241	26	80429	0.3863	52908	1.2e+06	279630	1.000	10844	0.0039	317	0.0004
	6	349	142	113517	25	80426	0.3552	42177	1.1e+06	179783	1.000	10867	0.0029	349	0.0002
	7	1988	105	67382	24	9036	0.3157	52882	1.1e+06	188380	1.000	201222	0.0029	3369	0.0002
	8	7690	96	52908	20	8989	0.2996	201222	1.1e+06	52504	1.000	207364	0.0029	7690	0.0002
	9	3369	92	72466	18	103181	0.2989	12751	6.9e+05	201087	1.000	152385	0.0029	1988	0.0002
	10	16460	79	269152	17	71888	0.2989	6940	6.4e+05	191708	1.000	237807	0.0029	16460	0.0002

Table 4: Top 10 key users identified by various centrality measures in mention dataset.

	Ranking	Indg		Outdg		Ev		Bt		Cl		Pr		Kz	
		Node Id	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**	**Node Id**	**Value**
SCC network	1	88	462	89805	129	88	1.0000	13808	728765.58	88	0.4807	88	0.0975	88	0.0083
	2	3998	137	1276	42	3998	0.5429	12751	613890.54	3998	0.4059	3998	0.0457	677	0.0027
	3	677	130	9021	42	13808	0.5284	89805	559561.26	89805	0.3992	52087	0.0406	3998	0.0021
	4	13808	119	26158	41	12751	0.2706	64911	551493.61	13808	0.3874	13808	0.0288	13808	0.0019
	5	1988	81	6241	35	67382	0.2605	88	520146.06	2417	0.3784	64911	0.0244	2417	0.0016
	6	2417	68	67382	32	5226	0.2366	6940	492184.47	5226	0.3742	677	0.0149	9021	0.0015
	7	5226	63	492	32	677	0.2214	67382	465101.31	1276	0.3742	3604	0.0092	1988	0.0013
	8	9021	55	12751	31	64911	0.2676	35376	424761.87	12751	0.3735	2417	0.0086	12751	0.0011
	9	12751	54	20385	30	11991	0.2115	110903	284255.80	26158	0.3684	12751	0.0049	3604	0.0011
	10	35376	51	4665	28	52087	0.1855	9021	263752.60	6241	0.3671	9021	0.0048	6241	0.0011
WCC network	1	88	11953	89805	169	88	1.0000	88	5.1e+07	88	0.3553	13813	0.0769	88	0.0089
	2	677	3906	26158	57	3998	0.5429	64911	5.0e+07	2417	0.3120	88	0.0466	677	0.0029
	3	2417	2533	1276	49	13808	0.5284	13808	4.2e+07	3998	0.3114	4741	0.0170	2417	0.0020
	4	59195	1601	9021	44	13813	0.5217	89805	2.6e+07	13813	0.3075	3998	0.0130	59195	0.0014
	5	3998	1587	6241	43	12751	0.2706	12751	2.6e+07	89805	0.3030	677	0.0125	7533	0.0012
	6	7533	1528	492	40	67382	0.2605	6940	2.3e+07	13808	0.2994	3369	0.0102	383	0.0010
	7	383	1357	149922	39	5226	0.2366	110903	2.2e+07	1276	0.2987	2417	0.0097	3998	0.0008
	8	1988	1189	67382	38	677	0.2214	67382	2.2e+07	677	0.2984	52087	0.0087	3369	0.0007
	9	13813	1066	4665	38	4259	0.2178	35376	1.9e+07	5226	0.2953	59195	0.0082	1988	0.0007
	10	519	805	12751	37	64911	0.2176	3998	1.9e+07	26158	0.2910	13808	0.0082	13813	0.0006

Full network	1	88	11953	89805	169	88	1.0000	88	5.1e+07	107757	1.000	13813	0.0642	88	0.0070
	2	677	3906	26158	57	3998	0.5429	64911	5.0e+07	124554	1.000	88	0.0389	677	0.0023
	3	2417	2533	1276	49	13808	0.5284	13808	4.2e+07	286277	1.000	4741	0.0142	2417	0.0016
	4	59195	1601	9021	44	13813	0.5217	89805	2.6e+07	284372	1.000	3998	0.0108	59195	0.0011
	5	3998	1587	8241	43	12751	0.2706	12751	2.6e+07	274148	1.000	677	0.0104	7533	0.0010
	6	7533	1528	492	40	67382	0.2605	6940	2.3e+07	274149	1.000	3369	0.0085	383	0.0008
	7	383	1357	149922	39	5226	0.2366	110903	2.2e+07	111667	1.000	2417	0.0081	3998	0.0006
	8	1988	1189	67382	38	677	0.2214	67382	2.2e+07	158380	1.000	52087	0.0073	3398	0.0005
	9	13813	1066	4665	38	4259	0.2178	35376	1.9e+07	52504	1.000	59195	0.0069	1988	0.0005
	10	519	805	12751	37	64911	0.2176	3998	1.9e+07	185346	1.000	13808	0.0068	13813	0.0005

Table 5: Top 10 key users identified by various centrality measures in retweet dataset.

	Ranking	Indg		Outdg		Ev		Bt		Cl		Pr		Kz	
		Node Id	Value	Node Id	Value	Node Id	Value	Node Id	Value	Node Id	Value	Node Id	Value	Node Id	Value
SCC network	1	88	209	53508	35	88	1.0000	64911	205916.69	88	0.4701	88	0.0583	88	0.0076
	2	3998	105	64911	34	3998	0.7022	6940	174494.42	677	0.4142	2342	0.0377	1988	0.0038
	3	677	95	27705	31	11991	0.6915	35376	162656.19	1988	0.4041	64911	0.0269	677	0.0033
	4	13808	71	492	26	42172	0.6521	28951	154796.13	27705	0.3949	3998	0.0212	5226	0.0026
	5	1988	69	75798	25	64911	0.5449	88	137928.53	6940	0.3938	39420	0.0167	349	0.0025
	6	2417	66	182906	25	13808	0.4732	103447	114613.31	3998	0.3905	169287	0.0161	9964	0.0021
	7	5226	47	39885	24	39885	0.4648	3547	114184.74	5226	0.3867	134095	0.0161	6940	0.0021
	8	9021	46	103447	23	56968	0.4268	511	99540.44	53508	0.3857	28951	0.0157	13808	0.0020
	9	12751	42	4509	22	3547	0.4199	9021	85390.09	50305	0.3842	13808	0.0156	3998	0.0019
	10	35376	38	35376	22	110903	0.4018	1988	81233.94	64911	0.3829	42172	0.0147	519	0.0019
WCC network	1	88	14060	38535	134	88	1.0000	64911	5.3e+07	88	0.3234	88	0.0275	88	0.0038
	2	14454	9160	181190	84	3998	0.7022	88	4.4e+07	677	0.2971	2342	0.0158	14454	0.0019
	3	677	5613	81405	66	11991	0.6915	35376	2.9e+07	38535	0.2957	64911	0.0098	677	0.0015
	4	1988	4335	64911	49	42172	0.6521	28951	2.7e+07	1988	0.2907	39420	0.0079	1988	0.0010
	5	349	2802	54301	49	64911	0.5449	6940	2.5e+07	14454	0.2849	14454	0.0077	283	0.0007
	6	283	2039	27705	48	13808	0.4732	103447	2.5e+07	1276	0.2832	677	0.0075	349	0.0006
	7	3571	1980	53508	42	39885	0.4648	3547	2.0e+07	6940	0.2831	2567	0.0069	68278	0.0006
	8	6948	1959	232850	41	56968	0.4268	677	1.8e+07	3998	0.2810	134095	0.0067	6948	0.0005
	9	14572	1692	492	38	3547	0.4199	39885	1.7e+07	349	0.2793	169287	0.0067	3571	0.0005
	10	68278	1689	52204	38	110903	0.4018	1988	1.4e+07	5226	0.2771	1988	0.0062	3549	0.0005
Full network	1	88	14060	38535	134	88	1.0000	64911	5.3e+07	326123	1.000	88	0.0253	88	0.0034
	2	14454	6190	181190	84	3998	0.7022	88	4.4e+07	326124	1.000	2342	0.0146	14454	0.0017
	3	677	5613	81405	66	11991	0.6915	35376	2.9e+07	81236	1.000	64911	0.0091	677	0.0013
	4	1988	4335	64911	49	42172	0.6521	28951	2.7e+07	57763	1.000	39420	0.0091	1988	0.0009
	5	349	2802	54301	49	64911	0.5449	6940	2.5e+07	172959	1.000	14454	0.0071	283	0.0006
	6	283	2039	27705	48	13808	0.4732	103447	2.5e+07	283287	1.000	677	0.0069	349	0.0006
	7	3571	1980	53508	42	39885	0.4648	3547	2.0e+07	263179	1.000	2567	0.0064	68278	0.0005
	8	6948	1959	232850	41	56968	0.4268	677	1.8e+07	293265	1.000	134095	0.0062	6948	0.0005
	9	14572	1692	492	38	3547	0.4199	39885	1.7e+07	321295	1.000	169287	0.0062	3571	0.0004
	10	68278	1689	52204	38	110903	0.4018	1988	1.4e+07	430115	1.000	1988	0.0057	3549	0.0004

Table 6: Results of the influence propagation of various centrality measures in a reply network under IC, LT and SIR model.

		LTM			ICM			SIR		
		SCC	WCC	Full	SCC	WCC	Full	SCC	WCC	Full
Reply network	Indg	100	46	46	10	10	10	208	408	410
	Outdg	102	219	233	10	27	25	242	528	647
	Cl	62	59	10	10	10	10	211	466	10
	Ev	52	48	48	10	10	10	190	460	416
	Bt	73	64	64	10	11	10	219	497	494
	Pr	93	10	10	10	10	10	200	374	343
	Kz	99	46	46	10	10	10	224	365	505

Table 7: Results of the influence propagation of various centrality measures in a mention network under IC, LT and SIR model.

		LTM			ICM			SIR		
		SCC	**WCC**	**Full**	**SCC**	**WCC**	**Full**	**SCC**	**WCC**	**Full**
Mention network	**Indg**	317	10	10	14	10	10	1094	1872	1848
	Outdg	333	209	209	35	54	45	1096	1927	1911
	Cl	290	146	10	25	28	10	1117	1854	11
	Ev	287	42	42	10	10	14	1140	1904	1814
	Bt	350	112	112	34	39	30	1101	1756	1922
	Pr	334	14	14	13	10	10	1121	1886	1909
	Kz	332	10	10	18	10	10	1155	1858	1874

Table 8: Results of the influence propagation of various centrality measures in a retweet network under IC, LT and SIR model.

		LTM			ICM			SIR		
		SCC	**WCC**	**Full**	**SCC**	**WCC**	**Full**	**SCC**	**WCC**	**Full**
Retweet network	**Indg**	227	10	10	10	10	10	608	1234	1678
	Outdg	193	184	184	21	62	50	638	1635	1635
	Cl	217	18	10	16	28	10	619	10	10
	Ev	138	27	27	10	10	11	657	1671	1671
	Bt	219	45	45	13	17	14	642	1664	1664
	Pr	143	13	13	10	10	10	569	1727	1727
	Kz	227	10	10	10	10	10	625	1296	1296

7 Conclusion

This study presented a comprehensive analysis of influential node identification in OSNs using seven centrality measures, including in-degree, out-degree, betweenness, closeness, eigenvector, pagerank and Katz centrality. It aimed to evaluate how influence propagates across different network structures, such as SCC and WCC and discern the effectiveness of various diffusion models, specifically the LT, IC, and SIR models. The primary objective of this study was to deepen the understanding of influence maximization in OSNs by systematically evaluating centrality measures in relation to network structures and influence diffusion dynamics.

Our analysis revealed that out-degree and betweenness centralities were the most effective measures for influence propagation, with out-degree being particularly strong in initiating the diffusion process due to its capacity to activate many neighbors, while betweenness played a critical role in sustaining influence over time by bridging different parts of the network. Furthermore, the SIR model demonstrated superior efficacy for sustained diffusion, reflecting a more realistic simulation of influence dissemination compared to LT and IC models, which are more appropriate for activation-based scenarios. We also found that utilizing WCC can optimized influence campaigns, enabling marketers to enhance efficiency while retaining key influencers.

However, the study acknowledged limitations, including reliance on established centrality measures and diffusion models that may not fully accounted for the complexity of user interactions in rapidly evolving digital landscapes. In future work, we aim to propose a hybrid model that combines the strengths of local and global centrality measures with the dynamics of different diffusion models based on our findings. This hybrid approach will incorporate temporal changes and real-time feedback mechanisms, ensuring adaptability to the evolving nature of OSNs.

References

[1] Safari, R. M., Rahmani, A. M., & Alizadeh, S. H. (2019). User behavior mining on social media: a systematic literature review. *Multimedia Tools and Applications,78*(23),33747-33804. https://doi.org/10.1007/s11042-019-08046-6

[2] Peng, S., Zhou, Y., Cao, L., Yu, S., Niu, J., & Jia, W. (2018). Influence analysis in social networks: A survey. *Journal of Network and Computer Applications,106*,17-32. https://doi.org/10.1016/j.jnca.2018.01.005

[3] Yan, Z., Zhou, X., Ren, J., Zhang, Q., & Du, R. (2023). Identifying underlying influential factors in information diffusion process on social media platform: A hybrid approach of data mining and time series regression. *Information Processing & Management,60*(5),103438. https://doi.org/10.1016/j.ipm.2023.103438

[4] Shen, B., Guan, T., Ma, J., Yang, L., & Liu, Y. (2021). Social network research hotspots and trends in public health: A bibliometric and visual analysis. *Public Healt in Practice*, *2,* 100155. https://doi.org/10.1016/j.puhip.2021.100155

[5] Ni, P., Zhu, J., Gao, Y., & Wang, G. (2024). Minimizing the misinformation concern over social networks. *Information Processing & Management*, *61*(1),103562. https://doi.org/10.1016/j.ipm.2023.103562

[6] Zheng, H., Zhao, H., & Ahmadi, G. (2024). Towards improving community detection in complex networks using influential nodes. *Journal of Complex Networks*, *12*(1),cnae001. https://doi.org/10.1093/comnet/cnae001

[7] Wang, W., & Street, W. N. (2018). Modeling and maximizing influence diffusion in social networks for viral marketing. *Applied network science*, *3*, 1-26. https://doi.org/10.1007/s41109-018-0062-7

[8] Medjahed, F., Molina, E., & Tejada, J. (2025). Effectiveness of Centrality Measures for Competitive Influence Diffusion in Social Networks. *Mathematics (2227-7390),13*(2). https://doi.org/10.3390/math13020292

[9] Meshcheryakova, N., & Shvydun, S. (2024). A comparative analysis of centrality measures in complex networks. *Automation and Remote Control*, *85*(8), 685-695. https://doi.org/10.1134/S0005117924700127

[10] Baabcha, H., Laifa, M., & Akhrouf, S. (2022). Social Influence Analysis in Online Social Networks for Viral Marketing: A Survey. In *International Conference on Managing Business Through Web Analytics* (pp. 143-166). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-06971-0_11

[11] Sarkar, D., Kole, D. K., & Jana, P. (2016). Survey of influential nodes identification in online social networks. *International Journal of Virtual Communities and Social Networking (IJVCSN), 8*(4), 57-69. https://doi.org/ 10.4018/IJVCSN.2016100104

[12] Singh, S. S., Singh, K., Kumar, A., Shakya, H. K., & Biswas, B. (2019). A survey on information diffusion models in social networks. In *Advanced Informatics for Computing Research: Second International Conference, ICAICR 2018, Shimla, India, July 14–15, 2018, Revised Selected Papers, Part II 2* (pp. 426-439). Springer Singapore. https://doi.org/10.1007/978-981-13-3140-4

[13] Arrami, S., Oueslati, W., & Akaichi, J. (2018). Detection of opinion leaders in social networks: A survey. In *Intelligent Interactive Multimedia Systems and Services 2017 10* (pp. 362-370). Springer International Publishing. https://doi.org/10.1007/978-3-319-59480-4_36

[14] Ribeiro, A. C., Azevedo, B., Oliveira e Sá, J., & Baptista, A. A. (2020). How to measure influence in social networks? In *International Conference on Research Challenges in Information Science* (pp. 38-57). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-50316-1_3

[15] Singh, R. R. (2022). Centrality measures: a tool to identify key actors in social networks. *Principles of Social Networking: The New Horizon and Emerging Challenges*, 1-27. https://doi.org/10.1007/978-981-16-3398-0_1

[16] Yujie, Y. (2020). A survey on information diffusion in online social networks. In *Proceedings of the 2020 European Symposium on Software Engineering* (pp. 181-186). https://doi.org/10.1145/3393822.343232

[17] Li, M., Wang, X., Gao, K., & Zhang, S. (2017). A Survey on Information Diffusion in Online Social Networks: Models and Methods. *Information, 8*(4), 118. https://doi.org/10.3390/info8040118

[18] Khatri, I., Choudhry, A., Rao, A., Tyagi, A., Vishwakarma, D. K., & Prasad, M. (2023). Influence Maximization in social networks using discretized Harris' Hawks Optimization algorithm. *Applied Soft Computing*, *149,* 111037. https://doi.org/10.1016/j.asoc.2023.111037

[19] Dhingra, S., Dodwad, P. S., & Madan, M. (2016). Finding strongly connected components in a social network graph. *International Journal of Computer Applications*, *136*(7), 1-5. https://doi.org/10.5120/ijca2016908481

[20] Bonifazi, G., Buratti, C., Corradini, E., Marchetti, M., Parlapiano, F., Ursino, D., & Virgili, L. (2025). Defining, Detecting, and Characterizing Power Users in Threads. *Big Data and Cognitive Computing*, *9*(3), 69. https://doi.org/10.3390/bdcc9030069

[21] Puigbò, J. Y., Sánchez-Hernández, G., Casabayó, M., & Agell, N. (2014). Influencer detection approaches in social networks: A current state-of-the-art. In *Artificial intelligence research and development* (pp. 261-264). https://doi.org/10.3233/978-1-61499-452-7-261

[22] Guzman, J. D., Deckro, R. F., Robbins, M. J., Morris, J. F., & Ballester, N. A. (2014). An analytical comparison of social network measures. *IEEE Transactions on Computational Social Systems*, *1*(1), 35-45. https://doi.org/10.1109/TCSS.2014.2307451

[23] Zafarani, R. (2014). *Social Media Mining: An Introduction*. Cambridge University Press. https://doi.org/10.1017/CBO9781139088510

[24] Wang, Y., Zhang, L., Yang, J., Yan, M., & Li, H. (2024). Multi-factor information matrix: A directed weighted method to identify influential nodes in social networks. *Chaos, Solitons & Fractals*, *180*, 114485.

https://doi.org/10.1016/j.chaos.2024.114485

[25] Freeman, L. C. (1977). A set of measures of centrality based on betweenness. *Sociometry* https://doi.org/10.2307/3033543

[26] Pereira, F. S., Gama, J., de Amo, S., & Oliveira, G. M. (2018). On analyzing user preference dynamics with temporal social networks. *Machine Learning*, *107*, 1745-1773.

https://doi.org/10.1007/s10994-018-5740-2

[27] Rashid, Y., & Bhat, J. I. (2024). Topological to deep learning era for identifying influencers in online social networks: a systematic review. *Multimedia Tools and Applications*, *83*(5), 14671-14714. https://doi.org/10.1007/s11042-023-16002-8

[28] Borgatti, S. P., & Everett, M. G. (2006). A graph-theoretic perspective on centrality. *Social networks*, *28*(4), 466-484. https://doi.org/10.1016/j.socnet.2005.11.005

[29] Duda, R. O., & Hart, P. E. (2006). *Pattern classification*. John Wiley & Sons

[30] Lü, L., Chen, D., Ren, X. L., Zhang, Q. M., Zhang, Y. C., & Zhou, T. (2016). Vital nodes identification in complex networks. *Physics reports*, *650*, 1-63. https://doi.org/10.1016/j.physrep.2016.06.007

[31] Liao, H., Mariani, M. S., Medo, M., Zhang, Y. C., & Zhou, M. Y. (2017). Ranking in evolving complex networks. *Physics Reports*, *689*, 1-54. https://doi.org/10.1016/j.physrep.2017.05.001

[32] Wenji, C., Bo, Y., Ruigang, Z., Qiong, W., Binsha, Z., Zengbo, L., & Kai, N. (2022). Research on Key Node Identification Method of Transmission Network based on Improved PageRank Algorithm. In *2022 41st Chinese Control Conference (CCC)* (pp. 5056-5061). IEEE. https://doi.org/10.23919/CCC55666.2022.9902364

[33] Zhao, L., Sun, P., Zhang, J., Peng, M., Zhong, Y., & Liang, W. (2023). A complex network important node identification based on the KPDN method. *Applied Sciences*, *13*(14), 8303. https://doi.org/10.3390/app13148303

[34] Kempe, D., Kleinberg, J., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 137-146). https://doi.org/10.1145/956750.956769

[35] Pastor-Satorras, R., Castellano, C., Van Mieghem, P., & Vespignani, A. (2015). Epidemic processes in complex networks. *Reviews of Modern Physics*, *87*(3), 925. https://doi.org/10.1103/RevModPhys.87.925

[36] Badr, N., Abdel-Kader, H., & Ali, A. H. (2021). Diffusion Models for Social Analysis, Influence and Learning. *IJCI. International Journal of Computers and Information*, *8*(2), 162-169.

https://doi.org/ 10.21608/ijci.2021.207863

[37] Tian, S., Mo, S., Wang, L., & Peng, Z. (2020). Deep reinforcement learning-based approach to tackle topic- aware influence maximization. *Data Science and Engineering*, *5*, 1-11.

https://doi.org/10.1007/s41019-020-00117-1

[38] Khalife, S., Read, J., & Vazirgiannis, M. (2021). Structure and influence in a global capital–ownership network. *Applied Network Science*, *6*, 1-21.

https://doi.org/10.1007/s41109-021-00359-6

[39] Ji, F., & Jin, J. (2025). A map-reduce algorithm to find strongly connected components of directed graphs. *Soft Computing*, 1-20. https://doi.org/10.1007/s00500-025-10451-z

[40] Zhong, Z., Lin, L., Jiang, Z., Yuan, X., Ngai, E., Lam, J., & Kwok, K. W. (2025). Connectivity Determination Algorithm for Complex Directed Networks. *IEEE Transactions on Network Science and Engineering*. https://doi.org/10.1109/TNSE.2025.3549777

[41] Tarjan, R. (1972). Depth-first search and linear graph algorithms. *SIAM journal on computing*, *1*(2), 146-160. https://doi.org/10.1137/0201010

[42] Tarjan, R. E., & Zwick, U. (2024). Finding strong components using depth-first search. *European Journal of Combinatorics*, *119*, 103815. https://doi.org/10.1016/j.ejc.2023.103815

[43] Sobhi, N. A. M. *Detection of terminal strongly connected components using a novel DFS-like algorithm* (Doctoral dissertation, Universiteit van Amsterdam).

[44] Bakshy, E., Rosenn, I., Marlow, C., & Adamic, L. (2012). The role of social networks in information diffusion. In *Proceedings of the 21st international conference on World Wide Web* (pp. 519-528). https://doi.org/10.1145/2187836.2187907

[45] Yanchenko, E., Murata, T., & Holme, P. (2024). Influence maximization on temporal networks: a review. *Applied Network Science*, *9*(1), 16. https://doi.org/10.1007/s41109-024-00625-3

[46] Yang, W., Shi, Q., Yan, J., Wang, C., Song, M., & Wu, M. (2024). Complementary influence maximization under comparative linear threshold model. *Expert Systems with Applications*, *238*, 121826. https://doi.org/10.1016/j.eswa.2023.121826

[47] Jaouadi, M., & Romdhane, L. B. (2024). A survey on influence maximization models. *Expert Systems with Applications*, *248*, 123429. https://doi.org/10.1016/j.eswa.2024.123429

[48] Monserrat, T. J. K. P., Pabico, J. P., & Albacea, E. A. (2015). A Hybrid Graph-drawing Algorithm for Large, Naturally-clustered, Disconnected Graphs. *arXiv preprint arXiv:1507.02766*. https://doi.org/10.48550/arXiv.1507.02766

[49] Veerman, J. J. P., & Kummel, E. (2019). Diffusion and consensus on weakly connected directed graphs. *Linear Algebra and its Applications*, *578*, 184-206. https://doi.org/10.1016/j.laa.2019.05.014

[50] Wang, B., Zhang, J., Dai, J., & Sheng, J. (2022). Influential nodes identification using network local structural properties. *Scientific Reports*, *12*(1), 1833. https://doi.org/10.1038/s41598-022-05564-6

[51] Ullah, A., Wang, B., Sheng, J., Long, J., Khan, N., & Sun, Z. (2021). Identification of nodes influence based on global structure model in complex networks. *Scientific Reports*, *11*(1), 6173. https://doi.org/10.1038/s41598-021-84684-x

[52] Wang, G., Alias, S. B., Sun, Z., Wang, F., Fan, A., & Hu, H. (2023). Influential nodes identification method based on adaptive adjustment of voting ability. *Heliyon*, *9*(5). https://doi.org/10.1016/j.heliyon.2023.e16112.

[53] Wang, Y., Zhang, L., Yang, J., Yan, M., & Li, H. (2024). Multi-factor information matrix: A directed weighted method to identify influential nodes in social networks. *Chaos, Solitons & Fractals*, *180*, 114485. https://doi.org/10.1016/j.chaos.2024.114485

[54] Csókás, E., & Vinkó, T. (2025). A Heuristic for Influence Maximization Under Deterministic Linear Threshold Model. *Informatica*, *48*(4). https://doi.org/10.31449/inf.v48i4.4805

[55] Tang, L., Zhou, Y., Wang, L., Purkayastha, S., Zhang, L., He, J., ... & Song, P. X. K. (2020). A review of multi-compartment infectious disease models. *International Statistical Review*, *88*(2), 462-513. https://doi.org/10.1111/insr.12402

[56] Bhat, N., Aggarwal, N., & Kumar, S. (2020). Identification of influential spreaders in social networks using improved hybrid rank method. *Procedia Computer Science*, *171*, 662-671. https://doi.org/10.1016/j.procs.2020.04.072

[57] Zhao, X., Liu, F. A., Wang, J., & Li, T. (2017). Evaluating influential nodes in social networks by local centrality with a coefficient. *ISPRS International Journal of Geo-Information*, *6*(2), 35. https://doi.org/10.3390/ijgi6020035

[58] Talukder, A., Alam, M. G. R., Tran, N. H., Niyato, D., Park, G. H., & Hong, C. S. (2019). Threshold estimation models for linear threshold-based influential user mining in social networks. *IEEE Access*,*7*,105441-105461. https://doi.org/10.1109/ACCESS.2019.2931925

[59] Wang, F., Zhu, Z., Liu, P., & Wang, P. (2019). Influence maximization in social network considering memory effect and social reinforcement effect. *Future internet*, *11*(4), 95. https://doi.org/10.3390/fi11040095

[60] Liang, J. C., Gong, Y. J., Wu, X. K., & Li, Y. (2024). Customized influence maximization in attributed social networks: heuristic and meta-heuristic algorithms. *Complex & Intelligent Systems*, *10*(1), 1409-1424. https://doi.org/10.1007/s40747-023-01220-2

[61] Ishfaq, U., Khan, H. U., & Iqbal, S. (2022). Identifying the influential nodes in complex social networks using centrality-based approach. *Journal of King Saud University-Computer and Information Sciences*, *34*(10), 9376-9392. https://doi.org/10.1016/j.jksuci.2022.09.016

[62] Şimşek, A. (2022). Lexical sorting centrality to distinguish spreading abilities of nodes in complex networks under the Susceptible-Infectious-Recovered (SIR) model. *Journal of King Saud University-Computer and Information Sciences*, *34*(8), 4810-4820.

[63] Waniek, M., Michalak, T. P., Wooldridge, M. J., & Rahwan, T. (2018). Hiding individuals and communities in a social network. *Nature Human Behaviour*, *2*(2), 139-147.

https://doi.org/10.1038/s41562-017-0290-3

[64] Ghalmane, Z., El Hassouni, M., & Cherifi, H. (2018, October). Betweenness centrality for networks with non-overlapping community structure. In *2018 IEEE workshop on complexity in engineering (COMPENG)* (pp.1-5).IEEE. https://doi.org/10.1109/CompEng.2018.8536229

[65] Doostmohammadian, M., Rabiee, H. R., & Khan, U. A. (2020). Centrality-based epidemic control in complex social networks. *Social Network Analysis and Mining*, *10*, 1-11.

https://doi.org/10.1007/s13278-020-00638-7

[66] More, J. S., & Lingam, C. (2019). A gradient-based methodology for optimizing time for influence diffusion in social networks. *Social Network Analysis and Mining*, *9*(1), 5.

https://doi.org/10.1007/s13278-018-0548-4

[67] Shelke, S., & Attar, V. (2019). Source detection of rumor in social network–a review. *Online Social Networks and Media*, *9*, 30-42.

https://doi.org/10.1016/j.osnem.2018.12.001

[68] Dey, P., Chaterjee, A., & Roy, S. (2019). Influence maximization in online social network using different centrality measures as seed node of information propagation. *Sādhanā*, *44*(9), 205. https://doi.org/10.1007/s12046-019-1189-7

[69] Sengupta, A., Middya, A. I., & Roy, S. (2024). Centrality measures on segmented entropy networks to identify influencers and influencees for financial market scenario. *International Journal of Data Science and Analytics*, 1-22. https://doi.org/10.1007/s41060-024-00608-8

Multimodal Deep Learning Approach for College Students' Mental Health Monitoring Using Online and Offline Data Integration

Jia Xu[1, *], Chunyan Huang[2]
[1]Information Technology Center, Zhejiang Business College, Hangzhou 310059, China
[2]School of Electronic Commerce, Zhejiang Business College, Hangzhou 310059, China
Email: Xujia1892@163.com; smallyellow98@sina.com
[*]Corresponding author

Keywords: Speech recognition, text extraction, facial expression recognition, psychological health monitoring, attention mechanism, emotion recognition

Received: June 12, 2025

In response to the difficulty of real-time monitoring and continuous tracking of college students' mental health in the era of new media, we collect the data from online student platforms and offline psychological interviews, and develop a college students' mental health monitoring system based on speech recognition, text extraction, and facial expression recognition, with the goal of achieving intelligent mental health management. The method is to first collect data from students' online network platforms and offline psychological interviews, mainly including multimodal information such as network text data, speech, video images, etc., to study automated speech recognition and text information extraction process methods. At the same time, for the micro expression recognition needs of video images, we propose a VGG19+SE+TA+LSTM network model, which extracts spatial features from four facial regions respectively. VGG19 is used as the convolutional neural network part on the traditional CNN+LSTM network structure, and channel and time attention mechanisms are introduced to enhance the network. The multi region features are fused as the features of a single frame image, and the multi frame image features are input in time series. The long short-term memory network (LSTM) based on time attention mechanism (TA) is used to extract temporal features. Experimental results have shown that integrating multiple modal data from online and offline sources can achieve the automation and intelligence of an intelligent monitoring system for college students' mental health. The fused feature algorithm improves the recognition rates of positive, negative, and neutral emotions by at least 8% and 4.8% respectively compared to the independent Fbank and MFCC feature algorithms, while the VGG19+SE+TA+LSTM network model improves the UF1 evaluation index by nearly 17.9% and 4.5% compared to the CNN+LSTM and VGG19+LSTM models, with providing emotional cognitive references for college students and effectively assisting college counselors in identifying the psychological emotions of college students.

Povzetek: Študija razvije inteligentni sistem za spremljanje duševnega zdravja študentov, ki združuje večmodalne podatke (besedilo, govor, video) z mikroizrazi prek modela VGG19+SE+TA+LSTM za avtomatsko prepoznavo čustev.

1 Introduction

According to the "2022 Survey Report on the Mental Health Status of Chinese College Students", psychological problems such as anxiety, depression, interpersonal communication disorders, and difficulties in self-awareness affect the learning outcomes and quality of life of college students. With the vigorous development of mobile Internet, new media has become an important platform for contemporary college students to obtain information, express themselves, and interact socially [1]. Digital mental health [2-3] has entered the fast lane of development, which provides new ideas for the prevention and research of college mental health. The combination of online and offline mental health intervention model emerged at the historic moment. Some colleges and universities use Internet technology, give full play to the advantages of new media technology, and expand the channels for college students to seek psychological help and counseling by integrating resources inside and outside the campus to build a new media mental health education platform or develop psychological counseling service APP. The campus mental health platform or APP provides convenient online psychological counseling. Through a comprehensive user feedback mechanism, universities can continuously track the usage and needs of college students. At the same time, the platform or APP can use new media virtual communities to build virtual psychological counseling communities and offline psychological counseling centers, achieving full coverage of online and offline psychological counseling [4]. The virtual community invites professional psychological consultants inside and outside the school to settle down in the mode of "Internet +psychological consultation", provides advice on college students' psychological problems through video, voice and other means, and provides timely online professional psychological counseling services. In addition, the virtual community is also connected with offline psychological counseling centers, establishing an online appointment mechanism for medical treatment, so that college students with face-to-face in-depth counseling needs can quickly obtain professional services from offline psychological counseling centers.

Traditional monitoring and evaluation of mental health in universities mainly rely on subjective and static data collection methods such as psychological scales and daily observations, which cannot dynamically understand students' psychological states, and cannot efficiently and objectively evaluate and warn students of their psychological problems. There are certain limitations in dealing with mental health problems among college students, and it is urgent to introduce new concepts and methods to improve the effectiveness of mental health education. Therefore, many scholars have conducted research on monitoring and evaluating the mental health of college students based on artificial intelligence technology, such as using artificial intelligence to collect and comprehensively analyze students' psychological status through multiple channels [5], using multidimensional indicators [6-7] for mental health evaluation, realizing the transformation of mental health monitoring from static monitoring to dynamic management, and the transformation of mental health evaluation from subjective evaluation to big data algorithms [8-9]. At the same time, artificial intelligence technology can analyze and compare historical data of groups and individuals [10-11] on the basis of full data sampling big data analysis [12-13], and intelligently predict the psychological state of college students based on deep learning algorithm models [14-15]. The above studies have demonstrated that artificial intelligence, relying on real-time dynamic full sample data sampling and deep learning algorithms, can effectively integrate existing evaluation results to dynamically obtain a complete picture of the data, which helps to solve the inherent weakness of traditional evaluation methods that are biased and inefficient, and compensates for the subjective and inefficient limitations of traditional mental health monitoring and evaluation methods. The research content of this paper (including methods, datasets, models used, performance indicators, etc.) is presented in Table 1 below:

Table 1: Comparison table of research on college student mental health monitoring systems

Dimension	Specific content	Comparison
method	1.Multimodal data fusion (online platform data + offline psychological interviews) 2.Speech recognition and text extraction technology 3.Improved VGG19+SE+TA+LSTM network model (for video images)	single-mode speech data single-mode text data single-mode facial expression image
dataset	1. Online: web text data 2. Offline: speech data, video and image data	single-mode dataset
algorithm	1. Multimodal feature fusion (Fbank/MFCC + visual features) 2. Spatial-Temporal Attention Mechanism (SE+TA)	Independent Fbank features Independent MFCC features Independent video frame image features
performance indicators	1. Emotion recognition rate 2. Model UF1 and UAR indicators	CNN+LSTM model VGG19+LSTM model

2 A psychological health service system that integrates online and offline services

Digital empowerment provides new possibilities for mental health education. Through the Internet, big data, artificial intelligence and other technical means, we can achieve personalized, accurate and efficient mental health education, and greatly improve the coverage and service quality of mental health education. This article aims to organically combine traditional offline psychological counseling services with online psychological service platforms to achieve an integrated online and offline psychological health intervention model, forming a new type of psychological health service system with complementary advantages. Specifically, it includes the following aspects:

(1) Offline services: Set up a dedicated psychological counseling room, equipped with a professional team of psychological counselors, regularly carry out individual counseling, group counseling and other activities, and enhance students' mental health awareness and self-adjustment ability through holding psychological health lectures, themed activities and other forms.

(2) Online services: Utilizing the school's official website, APP, and other platforms to build an online psychological service platform, providing students with 24-hour uninterrupted psychological assistance hotline, online consultation, psychological testing, psychological classes, appointments for offline psychological clinics, psychological quality development training, and online "cloud psychological counseling" and other functions. In addition, the popularization of psychological knowledge and mutual support can be achieved through the establishment of mental health education courses, online psychological mutual aid groups, and other means.

(3) Integration of both: organically integrating online and offline services, such as referring offline cases to online for follow-up; During the online consultation process, students who require in-depth intervention should be promptly guided to receive professional treatment offline; Regularly publish offline activity notifications through online platforms to expand coverage and increase participation.

This article gathers online student behavior data and offline psychological interview data, collects digital multimodal data (including online text, voice chat, and video images) from online psychological counseling platforms, mental health apps, and offline psychological interviews, constructs a psychological emotion recognition model based on deep learning technology, optimizes speech recognition, text extraction, and expression recognition algorithms, improves data analysis effectiveness, and aims to achieve real-time monitoring and intelligent evaluation of college students' mental health. This project was approved by Ethics Committee of Zhejiang Business College. Before collecting data, the research team first obtained informed consent from all students and informed them of the research purpose. At the same time, in order to protect the privacy of participants, the team anonymized the collected data, set limited access to the data, and only allowed team members participating in the research to process and analyze it. The system framework diagram is shown in Figure 1 below.

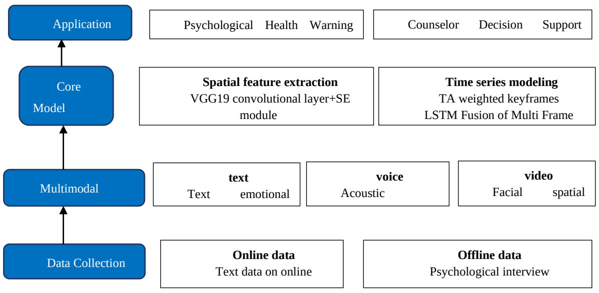


Figure 1: Framework diagram of college student mental health monitoring system

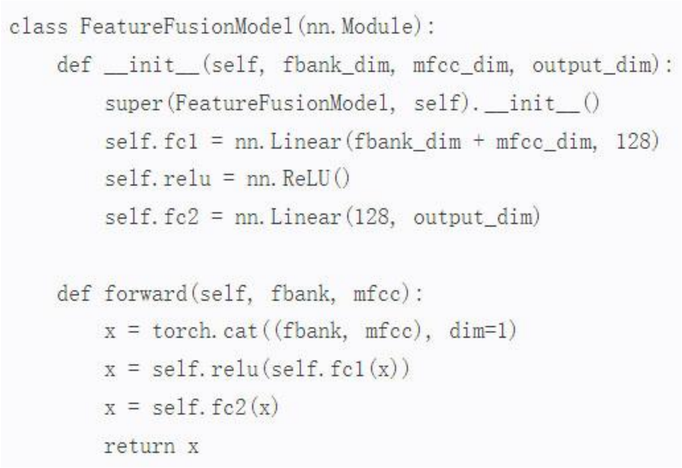
3 Speech recognition for monitoring the mental health of college students

Language data can reflect the health status of the brain, and sound features such as pitch, speed, intonation, volume, and prosody can reveal a person's psychological state [16]. We need to combine linguistic and psychological knowledge to analyze and model the biological and social cognitive information contained in language data and language related data, in order to achieve accurate prediction and evaluation of psychological disorders. By utilizing mobile devices and social platforms to collect voice and language data, digitizing it as a potential resource for early artificial intelligence screening of individuals with psychological disorders. This article studies an intelligent audio detection method for psychological disorders based on the mixed features of Fbank and MFCC [17], which mainly includes the following modules:

(1) Data collection module, used to collect audio datasets of online voice messages or offline psychological interviews with students; A clinically validated speech dataset consisting of 210 participants was constructed, with each participant receiving 15 seconds of audio as the dataset. The dataset categorizes emotions into three types: positive, negative, and neutral, with a ratio of 1:1:1.

(2) The audio preprocessing module is used for preprocessing audio, including audio clipping stage, feature extraction stage, and feature stitching stage. Extract Fbank and MFCC features [18] from the 15S audio, then concatenate them to obtain a 649 * 39 audio feature matrix.

(3) The audio feature vector determination module is used to determine the audio feature vector based on the audio dataset and the audio channel model; Cut the audio into a certain length, extract the Fbank and MFCC features of the audio separately, and then concatenate the Fbank and MFCC features. Compared to simple concatenation and weighted concatenation algorithms for different features, deep learning models have better feature fusion performance. We use convolutional neural networks (CNN) and deep learning models to fuse these two features, constructing an end-to-end neural network model that receives both Fbank and MFCC as inputs and outputs the final fused features or directly uses them for classification/regression tasks. The core pseudocode is as follows:



```
class FeatureFusionModel(nn.Module):
    def __init__(self, fbank_dim, mfcc_dim, output_dim):
        super(FeatureFusionModel, self).__init__()
        self.fc1 = nn.Linear(fbank_dim + mfcc_dim, 128)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(128, output_dim)

    def forward(self, fbank, mfcc):
        x = torch.cat((fbank, mfcc), dim=1)
        x = self.relu(self.fc1(x))
        x = self.fc2(x)
        return x
```

Figure 2: Fbank and MFCC feature fusion

In the aforementioned deep learning model, the EigenFusionModel class takes Fbank and MFCC as inputs, concatenates them, and processes them through a fully connected layer to ultimately output fused features.

(4) Two-dimensional convolutional neural network model construction module, used to construct a two-dimensional convolutional neural network model; The network fully connected stage includes a first deep convolution stage, a second deep convolution stage, and a network fully connected stage. The first deep convolution stage includes a first audio feature convolutional layer, a second audio feature convolutional layer, and a first audio feature pooling layer. The second deep convolution stage includes a third audio feature convolutional layer, a fourth audio feature convolutional layer, and a second audio feature pooling layer. The network fully connected stage includes an audio feature input layer, an audio feature hidden layer, and an audio feature input layer, as shown in Figure 3 below.

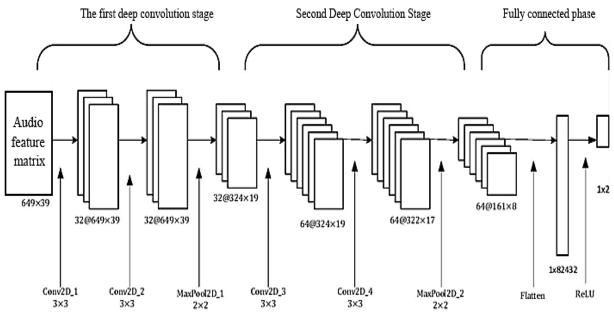


Figure 3: Two-dimensional convolutional neural network model

The two-dimensional convolutional neural network in the above figure consists of two deep convolution stages and one fully connected stage. First, the audio feature matrix Vec_{audio1} with a size of 649 × 39 is normalized by subtracting the average value and then dividing it by the maximum value. Then, in the first deep convolution stage: the first convolution layer (Conv2D_1)+the second convolution layer (Conv2D_2)+the first pooling layer (MaxPool2D_1), the convolution kernel size of the first and second convolution layers is set to 3 × 3, the number of convolution kernels is set to 32, the stride is set to 1, and the zero padding at the boundary is set to 1. The pooling layer adopts the maximum pooling method, and the pooling region kernel size is 2 × 2, with a stride of 2. The output feature vector Vec_{audio2} with 32 channels and a size of 324 × 19 is obtained. Next, through the second deep convolution stage: the third convolutional layer (Conv2D_2)+the fourth convolutional layer (Conv2D_2)+the second pooling layer (MaxPool2D_2), the convolution kernel size of the third and fourth convolutional layers is set to 3 × 3, the number is set to 32, the step size is set to 1, and the boundary zero padding is set to 1. The second pooling layer adopts the maximum pooling method, and the pooling region kernel size is 2 × 2, the step size is 2, the output channel number is 64, and the feature vector Vec_{audio3} with a size of 161 × 8 is output. The output of each convolutional layer is normalized by subtracting the average value and dividing it by the maximum value. Restore the distribution to its original input state. Flatten Vec_{audio3} into a feature vector Vec_{audio4} with a size of 1 × 82432, which serves as the input vector for the fully connected stage. The structure of the fully connected stage includes an input layer, a hidden layer, and an output layer. ReLU is used as the activation function, and the Dropout method is used to randomly inactivate a certain number of neurons to reduce overfitting. The inactivation probability is p=0.3, and the final output is a label feature vector Vec_{output} with a size of 1 × 2. Then use the sigmoid function to process Vec_{output} to obtain Vec_{target}, and determine whether it is a patient based on the two values of Vec_{target}.

(5) A label vector determination module is used to obtain label vectors based on the audio feature vector, video feature vector, and two-dimensional convolutional neural network model; The specific steps are as follows:

a. Set the number of convolution kernels in the first audio convolutional layer, the second audio convolutional layer, the third audio convolutional layer, and the fourth audio convolutional layer to 32, 32, 64, and 64, respectively. The size of the convolution kernels is set to 3 × 3, the step size is set to 1, and the boundary zero padding is set to 1. Normalize the outputs of each convolutional layer.

b. Both the first audio pooling layer and the second audio pooling layer adopt the maximum pooling method, with the pooling region kernel size set to 2 × 2 and the stride set to 2.

c. Flatten the output feature matrices of the first audio deep convolution stage and the second audio deep convolution stage into 1D feature vectors.

d. Take the 2D feature vector output by the audio preprocessing model as the input vector of the 2D convolutional neural network to obtain the 1D label vector.

(6) A module for determining patients with psychological abnormalities (negative emotions), used to identify patients with negative emotions based on the label vector.

Finally, experimental comparisons were conducted for different feature representations, and the above algorithm was applied to speech recognition operations. The speech data of 210 subjects in the data acquisition module were selected as the recognition content. The emotion classification results are "positive", "negative", and "neutral", with 70 samples for each classification. The sampling frequency is set to 16kHz and quantized to 16 bits. The training sample is set to 160, and the test sample is set to 50. First, train the model with training samples, conduct 5 experiments per group, calculate the mean as the result, and finally use the mean of the recognition and classification accuracy as the final recognition rate for each speech. The results are shown in Table 2, which compares the recognition rates of Fbank features, MFCC features, and their fusion features.

Table 2: Comparison of recognition rates of three feature extraction algorithms

	Positive	Negative	Neutral
Fbank features	78.6	77.8	82.3
MFCC features	82.4	81.6	85.5
Fusion features	88.1	86.7	90.3

From the results in the table above, it can be seen that using MFCC features can achieve better recognition performance compared to Fbank features. After feature fusion, the fused features can better present speech characteristics and achieve higher recognition rates.

4 Psychological prediction analysis based on text analysis

(1) Extraction of Online Consultation Text Data

Select the online consultation module built on the school's official website and online psychological service APP platform, and use the web crawler software "Octopus Collector" to crawl user consultation texts and related information, including user ID, question title, question content, question time, question status, questioner's gender, age, psychologist's answer, doctor's title, number of likes, and answer time. Then, the extracted text data is preprocessed, including: ① manual data cleaning to remove invalid text: removing questions that users only submit question titles but do not provide detailed descriptions of the question content, or have incomplete content expression and no doctor answers. ② Text segmentation: Use Jieba segmentation tool in Python to segment user questions. Due to the particularity of Chinese text, in order to achieve better segmentation results, it is necessary to use a complete segmentation vocabulary list for semantic segmentation. This article adds professional vocabulary such as psychiatric and psychological related terms and drug names to a custom vocabulary list to obtain more accurate segmentation results. ③ Removing stop words: There are still a large number of words in the question content that have no practical significance for topic analysis, such as "I", "Hello", "You" and other words that frequently appear in user questions but cannot provide reference for topic recognition. Therefore, they were removed in the study.

(2) Convert voice data into text

For speech data, in addition to distinguishing psychological abnormalities based on speech features in the third section, in order to improve the effectiveness of psychological monitoring, it is necessary to further convert it into text for analysis. The collection of voice data is mainly carried out by various personnel such as teachers from the psychological counseling center, selecting voice messages or interview dialogue data generated by college students for online "cloud psychological counseling" or offline due to their own psychological problems, mainly covering students' basic information, voice recordings, etc. In order to prevent the leakage of students' personal privacy, anonymization was carried out in the data preprocessing stage, blocking the 4-7 digits of the phone number and identifying and deleting the real name in the voice data, ensuring the privacy and security of the data. The data style is shown in Table 3 below:

Table 3: Speech data styles

Number	file name	duration
01	2024-03-05-198****0245.wav	00:35:36
02	2024-03-05-158****3598.wav	00:45:03
03	2024-03-05-153****9845.wav	00:15:42
………	………	………

Due to the fact that the speech information in the dataset we collected is mostly in Chinese, we need to utilize existing mainstream Chinese language speech recognition open-source tools both domestically and internationally for recognition. This article selects the ASRT model and three open-source tools, namely iFLYTEK and Baidu AI's speech recognition, to test the recognition performance of Chinese. The experiment found that iFLYTEK's tool had the best performance, and the comparison results are shown in Table 4:

Table 4: Comparison results of word error rates in speech recognition

Tool/Time Dimension	<10min	10-30min	10-30min
ASRT	10.3%	15.3%	19.8%
Baidu AI	5.2%	7.8%	12.1%
iFLYTEK	4.8%	7.6%	11.5%

In order to reduce the error rate after converting speech into text, we propose a speech recognition model based on multi tool fusion. This model integrates open-source tools to complement each other's strengths and weaknesses, fully leveraging their respective advantages to further improve the accuracy of speech recognition. The method is to use ASRT, iFLYTEK, and Baidu AI for speech recognition of the same speech data, conduct detailed comparative analysis through experiments, and compare the final text obtained through string comparison. For the differences in recognition, based on the principle of minority obeying majority, the content recognized by most speech recognition tools is selected as the final result. If all three are different, the result recognized by iFLYTEK with the lowest overall word error rate in the experiment is selected. After model processing, we can obtain the speech to text conversion result with the lowest error rate.

(3) Text sentiment analysis

The text data obtained after the above (1) and (2) processing is stored in Chinese, and then the ROST CM software [19] is used to study the emotional tendencies of patients with psychological disorders on online health platforms through text mining and sentiment analysis methods. The software will output three emotional tendencies: neutral emotion, positive emotion, and negative emotion, and score each emotion. For example, 0 represents neutral emotion, -1~-100 represents negative emotion,+1~+100 represents positive emotion, and the numerical value represents the degree level.

5 Real time psychological crisis warning based on facial recognition emotion analysis

At present, research has been conducted based on video using computer image processing technology to extract feature information from raw input facial emotion images, and classify facial emotion features according to human emotional expression, in order to achieve psychological state recognition, such as elderly depression recognition [20]. Due to the temporal characteristics of offline psychological interviews with college students, the addition of LSTM network can achieve better recognition performance compared to traditional CNN network image feature extraction. For example, in reference [21], VGG was used as the convolutional neural network part in the traditional CNN+LSTM network, which proved that this method can effectively model spatiotemporal interactions and identify salient features. Micro expressions are unconscious facial information that is difficult to artificially pretend or control, so they can better reflect people's true psychological state than macro expressions. This article proposes a VGG19+SE+TA+LSTM network model based on the requirements of micro expression recognition. VGG19 is added to the traditional CNN+LSTM network structure, and channel and temporal attention mechanisms are introduced to enhance the network. The VGG19+SE+TA+LSTM network model uses a visual geometry group network (VGG19) based on channel attention mechanism (SENet) to extract spatial features of the main facial regions, and the multi region features are fused as the features of a single frame image. The multi frame image features are input in time sequence to a long short-term memory network (LSTM) based on temporal attention mechanism (TA) to extract temporal features. The VGG19+SE+TA+LSTM network designed in this article is shown in Figure 4. The micro expression recognition algorithm mainly consists of four steps: 1) Preprocess the micro expression video frames, and capture the main areas of micro expression changes in each frame of the image: left eyebrow eye, right eyebrow eye, nose bottom, and lips; 2) Extract spatial feature information of each key region through VGG19 network, and fuse the features of these four regions as the spatial features of each frame image; 3) Embedding time and channel attention modules into the network to assign corresponding weights to different video frames and feature channels; 4) Utilize trained networks to achieve micro expression recognition.

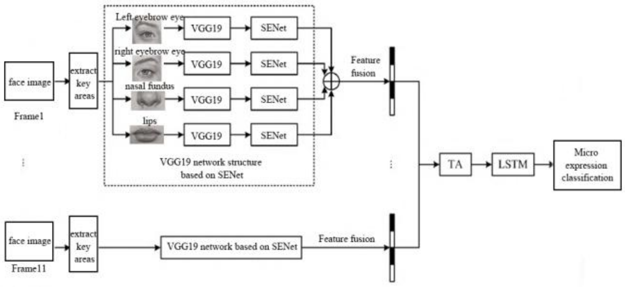


Figure 4: VGG19-SE-TA-LSTM Network

In the above figure, we use VGG19 to extract features of four important facial regions that reflect micro expression changes, and fuse the four features as spatial features of a single frame image. After obtaining spatial features through the VGG19 network, each frame of the image is input into the LSTM network in chronological order to obtain temporal features, and finally subjected to micro expression classification. This article adopts a unidirectional LSTM network, in which the hidden layer contains 256 nodes, followed by a fully connected layer for feature transformation, and then uses BN and Dropout layers to accelerate training convergence and prevent overfitting. Finally, the fully connected layer converts the feature vectors into the dimension of label vectors.

Channel Attention Module (SENet): A single facial region outputs 512 feature channels through the VGG19 convolutional network. After each VGG19 network, a Channel Attention (SENet) module [22] is introduced to improve the network's attention to important feature channels, suppress the influence of useless feature channels, and enhance the network's adaptability and performance by assigning weights to feature channels.

Time Attention Module (TA): Due to the short occurrence time of micro expressions, in order to highlight the role of keyframes in micro expression recognition, this paper introduces a time attention mechanism [23], which assigns corresponding weights to different frames and focuses attention on the keyframes in the sequence.

The output of the VGG19 network module is $F(x) = (f(x_1), f(x_2), \cdots, f(x_T))$ that the length of the sample sequence is T. $F(x)$ will be used as the input of the time attention module to obtain the hidden state $H = (h_1, h_2, \cdots, h_t)$, where h_t represents the hidden vector of the t frame of the sample sequence. Use a fully connected layer to calculate the frame-to-frame correlation in a micro expression sequence, as shown in formula (1) below.

$$s(h_t, h_i) = h_t^T W_a h_i \tag{1}$$

In the formula W_a represents the network weight matrix of the fully connected layer, h_t and h_i represents the hidden vector of the sequence.

At time step t, a_t represent the degree of influence of the entire time series on the time step vector h_t, where each element $a_{t,i}$ represents the magnitude of the effect of the i-th time step in the sequence on predicting the current time step t. $a_{t,i}$ use the normalized exponential function (softmax) to calculate, as shown in formula (2) below:

$$a_{t,i} = soft\max(s(h_t, h_i)) = \frac{\exp(s(h_t, h_i))}{\sum_{i=1}^{T} \exp(s(h_t, h_i))} \tag{2}$$

Finally, the weighted sum can obtain the attention weight a_t of each frame, and assign corresponding attention weights to different frames in the micro expression sequence, as shown in formula (3):

$$a_t = \sum_{i=1}^{T} a_{t,i} h_i \tag{3}$$

In order to verify the experimental comparison results of the algorithm in this paper, the experimental dataset uses the Institute of Psychology of the Chinese Academy of Sciences to build a CAS(ME)^3 face image database. This database provides approximately 80 hours of video, comprising over 8000000 frames, including 1030 manually annotated micro expressions and 3364 macro expressions. Such a large sample size can effectively validate the intelligent analysis method of micro expressions, while avoiding database bias. To ensure the accuracy of the samples in each experiment, considering that the research object of this article is college students, subjects with age too old (>35 years old) or too young (<18 years old) in the fused dataset, as well as samples with blurred images or mixed expressions, were excluded.

The basic emotion types in the database are divided into 8 categories: happiness, sadness, disgust, surprise, contempt, fear, repression, and tension. We classify emotions into positive, negative, surprise, or neutral, and the specific classification operation is: re label happiness as positive; Re label disgust, repression, anger, contempt, sadness, and fear as negative; The category of surprise remains unchanged; Other types of microexpressions are considered as neutral types. The final emotion classification dataset consists of 864 micro expressions, including 153 samples of positive emotions, 303 samples of negative emotions, 179 samples of surprise emotions, and 229 samples of neutral emotions. At the same time, in order to avoid the problem of small samples that restrict the application of deep learning in micro expression analysis and prevent overfitting of deep learning models, we enlarged the data to 10 times through image flipping, translation, scaling, mirror transformation, etc. for model training. The deep learning development platform used in the experiment is TensorFlow 2.0 framework. The experimental parameter settings are as follows: the optimizer introduces an adaptive Adam method; The initial learning rate is set to 0.0001, the attenuation factor is set to 0.8, and the minimum value is set to 0.000001; Set the batch processing quantity to 2 and the epoch to 200. This article sets up three sets of experiments to compare the training and recognition performance of traditional CNN+LSTM model, VGG19+LSTM model, and VGG19+SE+TA+LSTM model with attention mechanism under the same experimental data and environment, and observe the performance changes of the three models. Due to uneven distribution of labels, using traditional evaluation metrics such as Accuracy, Precision, Recall, and F1 will lead to excessive optimism towards those with large sample sizes. Use unweighted F1 score (UF1) and unweighted average recall (UAR) as performance metrics to avoid overfitting of the proposed method to a certain category. The calculation formulas for UF1 and UAR are as follows:

$$UF1=\frac{1}{C}\sum_{i}^{C}\frac{2TP_c}{2TP_c+FP_c+FN_c} \tag{4}$$

Among them, C represents the number of categories, which are divided into four categories: positive, negative, surprise, and neutral. Therefore, C=4; TP_c, FP_c, FN_c refers to true positives, false positives, and false negatives in the classification results.

$$UAR=\frac{\sum_{i=1}^{N_c}\mathrm{Re}\,call_i}{N_c} \tag{5}$$

In the above formula, N_c is the number of samples c, Recall refers to the recall rate, and the calculation formula is as follows:

$$\mathrm{Re}\,call_c = TP_c/(TP_c+FN_c) \tag{6}$$

The experimental results are shown in Table 6 and Figure 5. The VGG19+SE+TA+LSTM model with attention mechanism has better training and recognition performance than the other two models. By summarizing literature and consulting with psychological experts, it was found that students with abnormal mental health usually exhibit negative micro expressions such as tension, anger, disgust, fear, and suppression, and occasionally show micro expressions of "surprise". Based on the micro expression classification results of the model, the mental health risk level is classified. Defining "positive" and "neutral" emotions as low-risk states; Define occasional 'surprise' as a medium risk state; And negative emotions are defined as high-risk states.

Table 6: Comparison of experimental results of different algorithms

Evaluation indicator model	UF1 (fusion)	UAR (fusion)	UF1 (CASME)	UAR (CASME)	UF1 (CASMEII)	UAR (CASMEII)	UF1 (CAS(ME)^2)	UAR (CAS(ME)^2)
CNN+LSTM	0.5231	0.5482	0.4812	0.5014	0.5048	0.5036	0.5574	0.5652
VGG19+LSTM	0.6572	0.6783	0.5342	0.5624	0.5863	0.5846	0.6254	0.6315
VGG19+SE+TA+LSTM	0.7018	0.7036	0.6589	0.6654	0.6945	0.6956	0.7089	0.7046

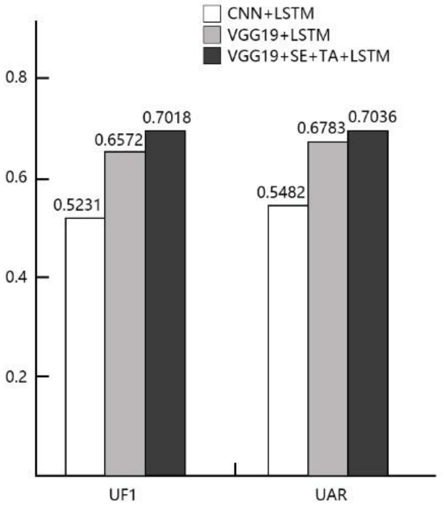


Figure 5: Comparison of recognition effects of different models

6 Discussion and summary

In today's society, mental health issues have become a global concern, especially in universities where students face multiple challenges such as academic pressure, interpersonal relationships, and future planning. The importance of mental health is self-evident. Therefore, how to effectively monitor and evaluate mental health has become an urgent issue for educators and professionals. The innovative improvements in network architecture proposed in this article contain: (1) dual enhancement of attention mechanism: the SE (Channel Attention) module dynamically calibrates the channel feature response to solve the problem of low channel information utilization in facial micro expression recognition of VGG19. TA (time attention) optimized the capture ability of LSTM for long sequence keyframes, and the synergy of the two significantly improved the UF1 index (17.9% higher than CNN+LSTM). (2) Multi region feature fusion strategy: Compared with traditional whole face input, four facial features were extracted in different regions and fused in time sequence, effectively alleviating the problem of local facial features being diluted by global information. At the same time, a trade-off between performance and cost was made through experimental comparison of different network models. In terms of computational efficiency, although VGG19+SE+TA+LSTM increased the number of parameters by about 15% compared to the basic VGG19+LSTM, it reduced redundant calculations by 30% through SE channel compression. At the same time, this paper controlled the overfitting of the model. In the experiment, it was found that the TA mechanism had a regularization effect on small-scale psychological datasets with sample sizes<1000, and the accuracy fluctuation of the test set was reduced by 2.3%. Of course, there are still limitations to this study, such as (1) the impact of sample diversity on data bias risk. Currently, UF1 improvement (4.5%) is based on campus scene data, and the generalization of micro expression recognition for cross-cultural/age groups needs to be verified. (2) the problem of label sparsity: the proportion of "neutral" emotion samples in psychological interview data reaches 62%, which may lead to insufficient sensitivity of UAR indicators to minority (negative emotions). Future research and improvement directions: (1) Optimization of computational costs: lightweight attention modules (such as ECA Net) can be explored to replace SE+TA combinations; (2) Multi modal collaboration bottleneck: Currently, text/speech modalities are only used for auxiliary decision-making, and future research on cross modal attention fusion mechanisms is needed.

This article explores a new practical model - multimodal emotional computing mental health services that integrate online and offline, with universities as the background, revealing its potential and value in improving students' mental health levels.

(1) Online mental health screening: overcoming spatial constraints and achieving comprehensive coverage

With the help of computer and Internet technology, colleges and universities have designed a psychological health assessment APP that is easy to operate and protects privacy, students can conduct psychological tests and self-assessment at any time and any place. This method is not only convenient and fast, but also can avoid the psychological pressure that face-to-face communication may bring, encouraging more students to actively participate. Meanwhile, big data analysis can help us more accurately identify students who may be experiencing psychological distress, providing a basis for subsequent interventions.

(2) Offline intervention: professional guidance, personalized care

Offline intervention is a deep response to the results of online screening. Each university has established specialized psychological counseling centers, equipped with professional psychological counselors, to provide one-on-one counseling services for students in need. Offline intervention emphasizes personalization and depth, which can provide more targeted assistance and meet the individual needs of students.

(3) Integration strategy: Complementary Advantages, Improving Service Quality

The key to the integration of online and offline lies in how to effectively combine the advantages of both. Online screening can serve as a preliminary "warning system" to identify students who may have problems, and then be deeply intervened by offline services. Meanwhile, offline services can also guide students to use online resources for self-learning and adjustment, forming a virtuous cycle. In addition, online platforms can also serve as feedback mechanisms to collect students' evaluations and suggestions on offline services, continuously optimizing service content and methods.

This article takes the author's university as an example. After implementing an integrated online and offline mental health service for one year, the number of students participating in screening significantly increased, and through offline intervention, many students were successfully helped to improve their psychological state. They regularly promote mental health knowledge through online platforms, and hold mental health weeks offline, providing free consultation and workshops. This model not only improves the accessibility and effectiveness of services, but also enhances students' awareness and importance of mental health. Looking ahead to the future, we still need to continue to innovate, and there is still huge room for development in the integration of online and offline mental health services. Future research directions include: (1) Text sentiment analysis using ROST CM is outdated and lacks rigor. Consider adopting transformer-based models for Chinese sentiment analysis, e.g., BERT variants trained on Chinese datasets. Also, quantify the sentiment prediction performance on a labeled subset. (2) Compare the emotional results (positive/negative/neutral) recognized by the system with the evaluation of professional psychologists, establish evaluation criteria such as accuracy, sensitivity, and specificity, and design a double-blind experiment to verify the consistency between the system evaluation and the professional evaluation. (3) Regarding the risk of false negatives, the system failed to identify the real psychological issues. In the future, research plans to adopt multimodal data cross validation and set sensitivity thresholds to prevent them.

Acknowledgment

This work is supported by the research result of scientific research project of Zhejiang Research Institute of Education Science “Research on the Construction of Multi modal Curriculum Resources and Personalized Teaching in Vocational Colleges Empowered by Embodied Intelligence” in 2025 (Grant No.:2025SCG340).

References

[1] Sun L, Yang Z. The problems and causes of college students' mental health education based on new media environment[J]. Applied & Educational Psychology,2024,5(3). DOI:10.23977/APPEP.2024.050322

[2] Kolenik T, Gams M. Intelligent Cognitive Assistants for Attitude and Behavior Change Support in Mental Health: State-of-the-Art Technical Review. Electronics. 2021; 10(11):1250. https://doi.org/10.3390/electronics10111250

[3] Kolenik, Tine & Gams, Matjaz. (2021). Persuasive Technology for Mental Health: One Step Closer to (Mental Health Care) Equality? IEEE Technology and Society Magazine. 40. 80-86. DOI:10.1109/MTS.2021.3056288.

[4] Kolenik, Tine. (2022). Methods in Digital Mental Health: Smartphone-Based Assessment and Intervention for Stress, Anxiety, and Depression. DOI: 10.1007/978-3-030-91181-2_7.

[5] Rui L. Early Warning Model of College Students' Psychological Crises Based on Big Data Mining and SEM. International Journal of Information Technologies and Systems Approach (IJITSA), 2023, 16(2):1-17. DOI:10.4018/IJITSA.316164

[6] Jingjing L, Guangyuan S, Jing Z, et al. Prediction of College Students' Psychological Crisis Based on Data Mining. Mobile Information Systems, 2021. DOI:10.1155/2021/9979770

[7] Li X. Research on the Application of Data Mining Technology in College Students’ Mental Health Education in the Network Age. Security and Communication Networks, 2022. DOI:10.1155/2022/4449066

[8] Panpan L, Feng L. An Assessment and Analysis Model of Psychological Health of College Students Based on Convolutional Neural Networks. Computational Intelligence and Neuroscience, 2022, 20227586918-7586918. DOI:10.1155/2022/7586918

[9] Cai B, Wang D. Prediction of psychological intervention for college students in digital entertainment media environment based on artificial intelligence and parallel computing algorithms. Entertainment Computing, 2025, 52100858-100858. DOI: 10.1016/J.ENTCOM.2024.100858

[10] Tine K, Gü S, Nter, et al. Computational Psychotherapy System for Mental Health Prediction and Behavior Change with a Conversational Agent [J]. Neuropsychiatric disease and treatment,2024,202465-2498. DOI: https://doi.org/10.2147/NDT.S417695

[11] Tine Kolenik. Intelligent cognitive system for computational psychotherapy with a conversational agent for attitude and behavior change in stress, anxiety, and depression. Informatica, 2025,49(2):451-454. DOI: https://doi.org/10.31449/inf.v49i2.8738

[12] International T O. Artificial Intelligence-Based Prediction of Individual Differences in Psychological Occupational Therapy Intervention Guided by the Realization of Occupational Values. Occupational therapy international, 2024, 9853562-9853562. DOI:10.1155/2024/9853562

[13] Li Y, Shuo S, Yu D. Graph Neural Network on Psychological Prediction of College Students Special Education. Journal of autism and developmental disorders, 2023, 54(4):1622-1622. DOI:10.1007/S10803-023-06068-6

[14] Vikas K, Praveen K, Masoud M. An intelligent disease prediction system for psychological diseases by implementing hybrid hopfield recurrent neural network approach. Intelligent Systems with Applications ,2023,18. DOI:10.1016/J.ISWA.2023.200208

[15] Computational N A I. Construction of a Prediction Model for College Students' Psychological Disorders Based on Decision Systems and Improved Neural Networks. Computational intelligence and neuroscience, 2023, 9813150-9813150. DOI:10.1155/2023/9813150

[16] Fadilah A N, Habibie H, Kristina A S, et al. Analysis of the mental health of pharmacy students at A number of public and private universities in Indonesia. Exploratory Research in Clinical and Social Pharmacy, 2024, 16100500-100500. DOI:10.1016/J.RCSOP.2024.100500

[17] Khan A W, Qudous U H, Farhan A A. Speech emotion recognition using feature fusion: a hybrid approach to deep learning. Multimedia Tools and Applications, 2024,83(31):75557-75584. DOI:10.1007/S11042-024-18316-7

[18] Singh K M. Identification of Speaker from Disguised Voice Using MFCC Feature Extraction, Chi-Square and Classification Technique. Wireless Personal Communications,2024,(prepublish):1-15. DOI:10.1007/S11277-024-11542-0

[19] Jia H, Wang X. The Performance Characteristics and Generation Logic of Citizen National Identity in News Communication of Major Scientific and Technological Achievements: Analysis of Short Video Comment Text Based on ROST CM 6.0 Software. Journal of Yangtze River Normal University, 2023, 39 (05): 65-74. DOI:10.19933/j.cnki.ISSN1674-3652.2023.05.008

[20] Tsai H H, Li R J, Shieh Y W. The Application of Emotion Valence Ratios in Facial Emotion

Recognition for Detecting Depression Among Older Adults in Institutional Settings. Studies in health technology and informatics, 2024, 318194-195. DOI:10.3233/SHTI240924

[21] Chouhayebi H, Mahraz A M, Riffi J, et al. Human Emotion Recognition Based on Spatio-Temporal Facial Features Using HOG-HOF and VGG-LSTM. Computers,2024,13(4):101. DOI:10.3390/COMPUTERS13040101

[22] Fu R, Tian M. Classroom Facial Expression Recognition Method Based on Conv3D-ConvLSTM-SEnet in Online Education Environment. Journal of Circuits, Systems and Computers,2023,33(07). DOI:10.1142/S0218126624501317

[23] Saheed K Y, Omole I A, Sabit O M. GA-mADAM-IIoT: A new lightweight threats detection in the industrial IoT via genetic algorithm with attention mechanism and LSTM on multivariate time series sensor data. Sensors International,2025,6100297-100297. DOI:10.1016/J.SINTL.2024.100297

Improving Stock Price Prediction through a Multilayer Perceptron Driven by a Grasshopper Optimization Algorithm: An Analysis of the Hang Seng Index

Lu Xia
Anhui Sanlian University, Hefei 230031, China
E-mail: xialuxialuxialu@126.com

Keywords: Hang Seng index, stock price, economic forecasting, multilayer perceptron, grasshopper optimization algorithm

Received: June 30, 2025

Given that time-series data is nonlinear, noisy, and dynamic, it may be difficult to predict stock prices in turbulent financial markets. To tune essential MLP hyperparameters such as the number of hidden units, learning rate, batch size, and epochs, this study presents a hybrid prediction model that combines a Multilayer Perceptron (MLP) with the Grasshopper Optimization Algorithm (GOA). On an 80/20 train–test split, the model is trained and tested on daily OHLC price and volume data from January 2015 to June 2023 for the Hang Seng Index (HSI). Benchmark models such as Transformer, Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), Outlier-Robust Extreme Learning Machine (OR-ELM), Histogram-Based Gradient Boosting Regression (HGBR), and other hybrid optimizers (BBO-MLP, GA-MLP) are used for comparing performance. With an R^2 value of 0.9912, MAPE value of 0.83%, MAE value of 170.06, and MSE value of 48,618 on the test set, experimental results demonstrate that the suggested GOA–MLP achieved the highest accuracy according to all measures. MAPE and MAE were decreased by GOA–MLP by about 48.1% and 47.9%, respectively, in comparison with the best deep learning baseline (Bi-LSTM). With a total return of 96.52%, a maximum drawdown of 1.7%, and a Sharpe ratio of 3.14 far above that for a Buy & Hold strategy back testing the model in an artificial trading environment confirmed its effectiveness. The results show that the introduced GOA–MLP offers significant risk-adjusted performance enhancement under actual investment scenarios and improves predictive capability.

Povzetek: Študija predstavi hibrid GOA-MLP, kjer optimizator samodejno uglašuje hiperparametre MLP za napovedovanje cen iz šumnih, nelinearnih časovnih vrst ter s tem izboljša napovedno in trgovalno učinkovitost.

1 Introduction

Unpredictable changes in stock prices are because many factors that are interrelated to this behavior. Some of the possible causes influencing this phenomenon can be global economic indicators, changes in unemployment, monetary policy formulated by influential countries, immigration policy, natural calamities, public health situations, and so on. All participants in the stock market now aim to earn maximum income and reduce risk through thorough analysis. The collection of varied data, its integration into a logical framework, and the creation of models that can be trusted for precise forecasting, however, continue to be formidable obstacles. For businesses, investors, and equity traders looking to profit from upcoming market movements, stock price forecasting is a complex and difficult task. Timely and accurate forecasting is particularly challenging in the stock market due to its inherent characteristics, which include high noise levels, non-parametric patterns, nonlinear dependencies, and elements of deterministic chaos [1]. Even if the future were considered entirely uncertain and randomly unfolding day by day, it would yet fall into the realm of prediction, allowing one to foresee events and practically secure profit from them. The AI and ML methods to forecast stock market movements constitute one such approach. Even though the stock market is highly unpredictable, it is still good practice to feed in AI-generated forecasts as input before committing investment funds [2]. An Artificial Neural Network (ANN) is an algorithm specifically developed to comprehend complex problems that cannot be solved by simple ML algorithms or conventional neural networks. ANNs possess a higher level of complexity and sophistication in their relationships than the human brain. The method employs algebraic equations to guide data toward a model or time-series line. Undoubtedly, ANN is a highly popular ML technology with extensive application across disciplines. The technique has demonstrated extraordinary performance in interpreting the correlation between input and output data of complex physical processes. The growing applicability of hybrid predictive models, which fuse the merits of several approaches to improve forecasting accuracy, has been accentuated by recent developments in ML, particularly for complex, nonlinear domains like financial time series. A Grey Wolf Optimizer (GWO) method was proposed by Sneha S. et al. [3] to improve stock price prediction by automatically identifying the best parameters for GARCH and ARIMA models. Their method, compared to hand-tuned conventional models, improved efficiency in forecasting by 5% to 8%. Essam H. Houssein et al. [4] outperform many recent metaheuristics with their hybrid forecasting model, employing support vector regression alongside the equilibrium optimizer (EO-SVR) for predicting closing prices on the Egyptian Exchange. Their analysis demonstrates that little importance is placed on statistical values and technical indicators in prediction performance. For DJIA index stock price forecasting, Burak Gülmez suggests an optimized deep LSTM model using the Artificial Rabbits Optimization (ARO) algorithm (LSTM-ARO). From the outcome of other evaluation metrics, the LSTM-ARO model provides better performance than other neural networks and optimization models [5]. Supported by cutting-edge training methods and Principal Component Analysis (PCA)-based feature selection, Heng Lyu [6] suggests a hybrid ARIMA–LSTM stock market forecasting model that integrates linear and nonlinear pattern recognition. Ernest Kwame Ampomah et al. [7] use a variety of feature extraction and scaling techniques to test the performance of the Gaussian Naïve Bayes (GNB) algorithm in predicting stock price movement. GNB outperforms other GNB-based models on several important metrics in their findings and offers a considerable enhancement in predictive accuracy when combined with Linear Discriminant Analysis (LDA) and Min-Max scaling. Ramzi Saifan discusses, with the Quantopian simulator, the performance of three prediction and daily stock market trading ensemble machine learning techniques: Extremely Randomized Trees, Random Forest, and Gradient Boosting. The paper highlights the great potential of ensemble methods in automated trading systems by showing that all models with technical indicators trained produce substantial returns and high alpha values [8]. Ernest Kwame Ampomah uses NYSE, NASDAQ, and NSE data to study how well tree-based AdaBoost ensemble machine learning algorithms predict the stock market. AdaBoost-ExtraTrees (Ada-ET) outperformed all other algorithms on a range of evaluation measures in the experiment, demonstrating the power of ensemble techniques in predictive performance [9].

Using an ensemble of LSTM-based recurrent neural networks (RNNs) in exchange market forecasting, Algirdas Maknickas and Nijolė Maknickienė suggest a decision support system to investors [10]. The system uses distribution-based prediction methods like high-low, daily-weekly, and UK-NY timing to improve the signal recognition and set the limits. These methods were best for successful, short-term trading in volatile currency markets. The concept of biological neural networks in the brain is primarily inspired by the work of Rosenblatt in 1958. The Multilayer Perceptron (MLP) is a highly utilized form of ANN for constructing data-driven models [11]. Essentially, MLP is comprised of several artificial neurons or computational nodes [12]. MLP consists of three types of layers: the input layer, the hidden layer, and the output layer. Each layer of the neural network consists of linked neurons, which are coupled using weights and biases. Although MLPs have shown promise in a variety of fields, their effectiveness is largely dependent on the choice and initialization of hyperparameters, which can result in slow convergence or entrapment in local minima if done incorrectly.

To resolve design issues that are encountered in the actual world, metaheuristic algorithms that incorporate stochastic operators are increasingly being utilized in engineering [13], [14]. However, deterministic algorithms are not very successful at discovering global optimal solutions because they tend to become stuck in local optimal solutions [15]. Deterministic algorithms are considered to be reliable. Stochastic optimization algorithms, like evolutionary algorithms, can avoid local solutions and find global optimal solutions inside search spaces by using randomization as a core strategy [16]. When it comes to avoiding local optima, these approaches perform better than deterministic algorithms, even though each cycle of these approaches has the potential to give a different result. For instance, the implementation of genetic algorithms (GA) [17], ant lion optimization (ALO) [18], slime mold algorithms (SMA)[19], biogeography-based optimization (BBO)[20], and moth-flame optimization (MFO) [21] at this point. As stated before, the use of metaheuristic optimization algorithms for parameter optimization is explored in this paper to evade the setbacks of conventional MLP training approaches, which are frequently plagued by slow convergence and entrapment in local minima. To improve the learning ability of MLP, this research specifically examines three evolutionary optimizers: BBO, GA, and Grasshopper Optimization Algorithm (GOA). In unstable financial markets like the Hang Seng Index, these algorithms are used to tune the network weights and biases with the aim of accelerating convergence, avoiding poor local solutions, and enhancing prediction accuracy. The grasshopper is a perilous bug in the natural world that devastates plants and damages agricultural yields. According to Simpson et al. [22], grasshoppers are known to form one of the largest swarms in existence, even though they are typically observed as individual insects. The grasshopper undergoes a life cycle consisting of three distinct stages: egg, nymph, and adult. The phenomenon of food source swarming is an intriguing trait seen in grasshoppers, which served as the inspiration for the development of an optimization algorithm [23], [24]. In nature-inspired algorithms, the process of finding food sources often occurs in two stages: exploration and exploitation. In the case of the Grasshopper Optimization Algorithm (GOA), the exploitation process moves slowly, whereas the exploration process moves quickly [25]. The grasshopper naturally engages in both exploration and exploitation. [24]. Subsequently, this research modeled them to create the GOA model. The research aims to (i) improve prediction accuracy over benchmark models; (ii) improve robustness and convergence stability in volatile market conditions; and (iii) validate the model in a realistic investment setting through back testing, which includes cumulative return, drawdown, and Sharpe ratio analysis. Comparative tests against a variety of benchmarks demonstrate that the GOA–MLP meets and exceeds these objectives, offering significant improvements in both predictive accuracy and actual trading performance. Following this introduction, the document is formatted as follows: Section 2 of the investigation involves investigating the data source in detail and its associated aspects, which is just one of the analytical approaches used in the study. Section 3 elaborates on the obtained results and the pertinent discourse. The key conclusions are then succinctly outlined.

2 Methods and materials

2.1 The proposed model's outline

To predict the Hang Seng Index (HSI), this study suggests a hybrid approach that includes a MLP neural network and the GOA. The structure of the GOA-MLP model comprises two well-harmonized parts, as shown in Figure 1: the GOA tunes the network parameters for improved learning and convergence, and the MLP learns intricate patterns from the stocks. OHLC (Open, High, Low, Close) price and volume of the HSI from January 2015 to June 2023 were used to train and test the model; 80% of the data were used for training and 20% for testing. The predictive accuracy was greatly improved by using GOA as a metaheuristic optimizer. This allowed the model to avoid local minima and better adapt to the chaotic and nonlinear characteristics of financial time series. According to the results, the GOA-MLP hybrid model offers a reliable and effective framework for predicting the stock market in volatile conditions.

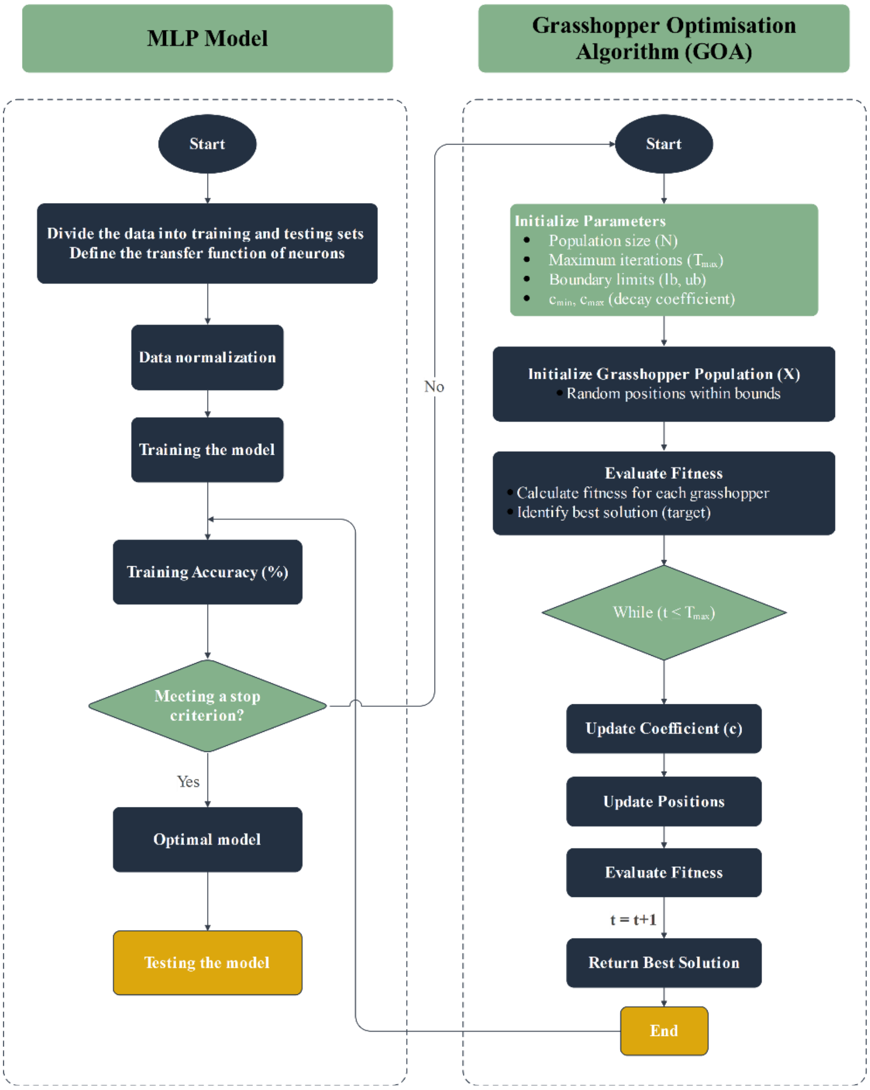

MLP Model
Grasshopper Optimisation Algorithm (GOA)
Start
Divide the data into training and testing sets
Define the transfer function of neurons
Data normalization
Training the model
Training Accuracy (%)
Meeting a stop criterion?
Yes
No
Optimal model
Testing the model
Start
Initialize Parameters
Population size (N)
Maximum iterations (T_{max})
Boundary limits (lb, ub)
c_{min}, c_{max} (decay coefficient)
Initialize Grasshopper Population (X)
Random positions within bounds
Evaluate Fitness
Calculate fitness for each grasshopper
Identify best solution (target)
While ($t < T_{max}$)
Update Coefficient (c)
Update Positions
Evaluate Fitness
t = t+1
Return Best Solution
End

Figure 1: The GOA-MLP flowchart.

2.2 Biogeography based optimization

In 2008, Dan Simon introduced the BBO algorithm [20] as an innovative meta-heuristic method that involves transferring organisms between islands to find the best possible environment. The BBO technique uses the habitat suitability index (HSI) to evaluate the efficacy of a remedy. A solution with a high HSI is deemed favorable, whereas a solution with a low HSI is deemed unfavorable. The BBO algorithm has two fundamental operations: migration and mutation. The migration technique is utilized to improve poor solutions by transferring attributes from superior solutions (solutions of superior quality) onto the suboptimal ones. The mutation technique is utilized to stochastically modify one or more attributes of the solution, based on a predetermined probability. Mutation has the potential to enhance the variety within a population while also impeding the algorithm's progress towards a stable state. The following equations can be employed to get the immigration and emigration rates (I and k, respectively) for each iteration of the improvement loop:

$$\mu_k = \frac{E \times k}{n} \lambda_k = I\left(1 - \frac{k}{n}\right) \tag{1}$$

The rate at which individuals depart the habitat is represented by the symbol μ_k, which stands for the exit rate. The migration rate of the kth habitat can be represented by the symbol λ_k. I: The highest possible migration rate can be attained. The condition $n = S_{max}$is used to define the maximum number of species that a habitat is capable of supporting. To denote the highest possible migration rate, the letter E has been assigned. K: total number of species.

2.3 Genetic algorithm

J.H. Holland developed the genetic algorithm (GA), a metaheuristic algorithm, in 1992 [26], [27]. It is influenced by the evolution of biology. Makes use of initial point mutations and cross-overs to maximize a set of goals that are suggested for a thorough comprehension of these methods, which includes maximizing hyper-parameters. Furthermore, a GA accesses the objective function's results to determine which locations are the most interesting from an optimization perspective. The population is the basis of the GA algorithm. This suggests that a single population with a few chromosomes is where GA begins. For any possible combination of chromosomes, there is only one solution. GA can rate the performance of each chromosome by ranking each solution based on the fitness function [28].

Furthermore, a new population is generated by GA procedures (selection, crossover, mutation). GA creates and rates new populations by iteratively carrying out the previous procedure. The fitness function of each new solution determines its performance. Ultimately, GA will be ended under specific circumstances, and the best possible settlement will be found.

2.4 Grasshopper optimization algorithm

The swarm intelligence optimization method is very capable of effectively solving complex problems and has gained significant popularity in the last twenty years due to advancements in computer technology. To address this issue, the GOA is utilized. The GOA was introduced by Shahrzad Saremi in 2017 [24]. Upon its proposal, this new intelligent optimization technique quickly garnered significant interest. It has been applied in several domains due to its notable effectiveness and straightforward functionality [29], [30], [31], [32], [33]. The GOA is a nature-inspired algorithm, to according Figure 2, that was proposed based on the social interaction of grasshoppers. Figure 3 depicts this optimizer's entire process.

The subsequent model mathematically depicts the swarming behavior of grasshoppers:

$$X_i = r_1 G_i + r_2 A_i + r_3 S_i \tag{2}$$

Xi denotes the current position of the i-th grasshopper. The values $r_i(i = 1,2,3)$ are random numbers between 0 and 1, which are used to introduce randomness in their behavior.

The gravitational force experienced by the ith grasshopper at birth is denoted as G_i and may be computed using the following formula:

$$G_i = -g\hat{e_g}. \tag{3}$$

You may find the wind advection force on the ith grasshopper, A_i, by applying the subsequent equation:

$$A_i = u\hat{e_w} \tag{4}$$

The following formula may be used to calculate the social aptitude of the ith grasshopper, which is represented as S_i:

$$S_i = \sum_{j=1, j\neq i}^{N} s(x_j - x_i) \tag{5}$$

$\widehat{e_g}$ and $\widehat{e_w}$ denote unit vectors in distinct directions where g and u are constants. The distance between the j-th and i-th grasshoppers is shown by the symbol $|x_j - x_i|$, whereas N is the number of grasshoppers. The social interaction function is defined as the s in Eq. (5):

$$s(r) = fe^{-t} - e^{-r}, \tag{6}$$

Where f and l stand for the attraction's strength and endurance. The mathematical model of the GOA is presented in its ultimate form after undergoing further modification.

$$X_i^d = c\left(\sum_{j=1, j\neq i}^{N} c\frac{ub_d - lb_d}{2} s(x_j - x_i)\right) + T_d, \tag{7}$$

Where ub_d and lb_d denote the upper and lower bounds respectively, $\widehat{T_d}$ represents the current best solution, and c is a shift coefficient explained as:

$$c = c\max - l\frac{c\max - c\min}{L}. \qquad (8)$$

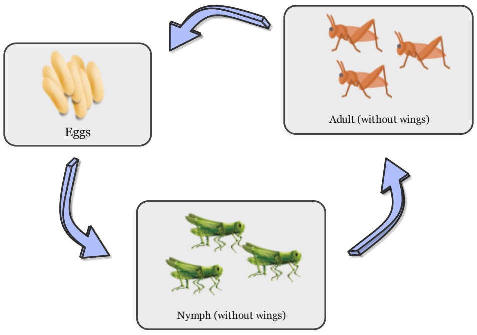


Figure 2: Swarm of grasshopper

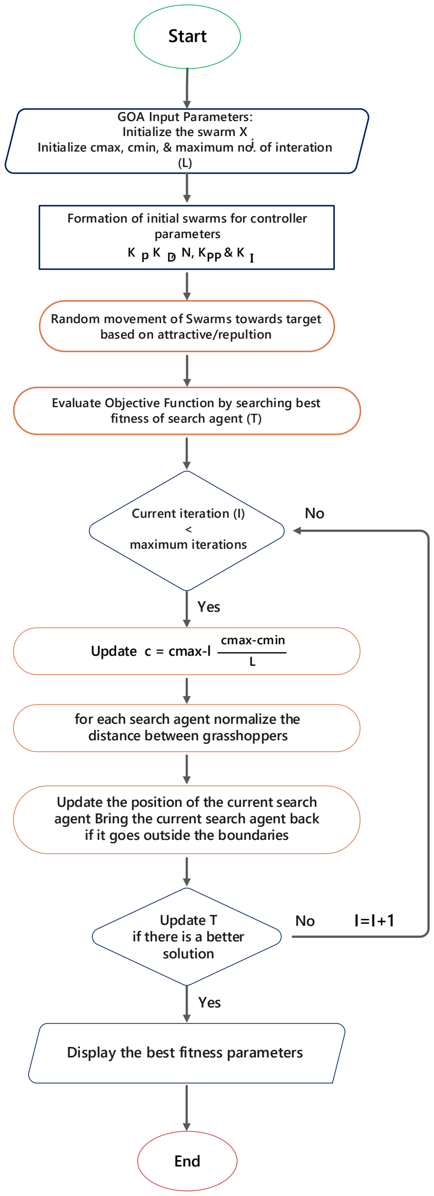

Start
GOA Input Parameters:
Initialize the swarm X
Initialize cmax, cmin, & maximum no. of interation (L)
Formation of initial swarms for controller parameters
K p K D N, Kpp & K I
Random movement of Swarms towards target based on attractive/repultion
Evaluate Objective Function by searching best fitness of search agent (T)
Current iteration (l) < maximum iterations
No
Yes
Update c = cmax-l cmax-cmin / L
for each search agent normalize the distance between grasshoppers
Update the position of the current search agent Bring the current search agent back if it goes outside the boundaries
Update T if there is a better solution
No
l=l+1
Yes
Display the best fitness parameters
End

Figure 3: The flowchart of the GOA.

Every optimization algorithm employed in this study has its distinctive contribution to its ability to improve the learning capacity of the MLP. BBO provides an efficient mechanism for global search by harnessing the migration and mutation of fit solutions across habitats. GA provides an exploration-exploitation balance through artificial simulation of natural evolutionary processes via crossover, mutation, and selection. According to the swarming behavior of grasshoppers, GOA uses a well-defined social interaction model that demonstrates better dynamic adjustments in the search process. Compared with GA and BBO, GOA uses a nonlinear coefficient to gradually decline during the phase transition process from exploration to exploitation. This feature enables an algorithm to efficiently avoid premature convergence and explore the complicated solution space. Therefore, the parameters of the MLP can be adjusted more accurately and smoothly using GOA.Benchmark models

2.5.1. Outlier robust extreme learning machine

The OR-ELM is a variant of the Extreme Learning Machine that reduces outlier sensitivity with regularization or robust loss functions. This improves prediction performance and generalization in noisy data with the low complexity and fast training rate of ELM [34].

2.5.2. Transformer

The Transformer model allows for highly parallelizable training and robust performance on a vast variety of time-series forecasting tasks by learning long-range dependencies in sequential data without recurrence through the adoption of a self-attention mechanism [35].

2.5.3. Histogram-based gradient boosting regression

Histogram-Based Gradient Boosting Regression, or HGBR, is an ensemble method based on a decision tree that uses histogram-based binning of continuous features to speed up gradient boosting with the focus on high predictive accuracy [36].

2.5.4. Long short-term memory

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network that can learn long-term dependencies in sequence data using gating units and memory cells that prevent vanishing gradient problems [37].

2.5.5. Bidirectional long short-term memory

Bi-LSTM networks improve sequence modeling performance by reading input sequences in both the forward and backward directions, thereby encoding bidirectional temporal dependencies [38].

2.5.6. Multilayer perceptron

MLP models are recognized as a prominent category of ANN models. Their actions were guided by both human cognition and artificial intelligence [39]. The MLP model has individual computational units known as neurons. Every neuron is situated within distinct layers. The first layer is accountable for accepting inputs. The concealed levels get the input layers' income indicators [40], [41], [42]. The neuron's output is determined by activation functions. Each layer's neurons are closely connected to those of their adjoining layers. The last layer generates the outputs by functioning as the output layer. The MLP model uses the function that adds up all the results to calculate the total value obtained by multiplying every input:

$$Z_j = \sum_{i=1}^{n} \omega_{ij} I_i + \beta_j \tag{9}$$

n denotes the aggregate quantity of input neurons. $\mathrm{I_i}$ represents the input, β_j represents the bias, The weight of the link between the i_{th} node in the input layer and the j_{th} node in the hidden layer is indicated by the variable ω_{ij}. The variable Z_j represents the sum function. The outcomes of Eq. (9) are passed on to the activation function. Prior studies have shown that the sigmoid function is efficient in handling data within the MLP model, as evidenced by the studies conducted by Seifi et al. [43]and Jalali et al. [44]. The MLP model's structure is depicted in Figure 4.

Here is how the activation is used by the model:

$$f_j(x) = \frac{1}{1+e^{-z}} \tag{10}$$

The function $f_j(x)$ represents the activation function. The ultimate result is calculated in the following manner:

$$\text{out}_{\,i} = \left(\sum_{i=1}^{n} \omega_{ij} I_j + \beta_J\right) \tag{11}$$

Which $out_{\,i}$ is the last output of the MLP. These models generally incorporate conventional training processes, such as the gradient technique to determine the MLP's parameters, which include biases and weights. These algorithms may converge very slowly, or worse, get trapped in a local optimum.

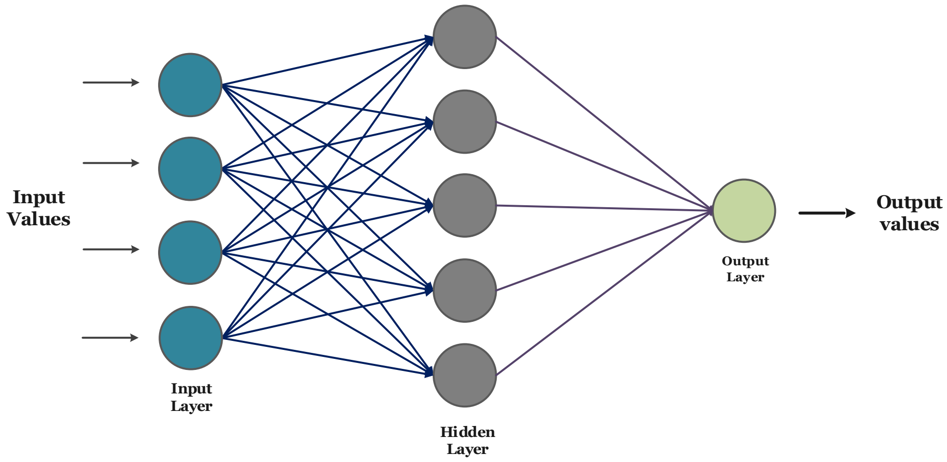


Figure 4: The methodology's structure of MLP.

The number of hidden units, training epochs, batch size, and learning rate are the most important hyperparameters tuned in this study. The search space and the final values chosen by each optimization algorithm (BBO, GA, and GOA) appear in Table 1. To find the setup with the minimum prediction error, each optimizer independently searched in the given parameter space. Its enhanced predictive accuracy is attributed to the GOA's choice of parameters, including fewer hidden units, moderate epochs, and a smaller learning rate. These parameters strike a balance between training stability and convergence rate. The GOA, which used a population size of 100 candidate solutions to search the hyperparameter space, chose the configurations that were used to train the MLP. To ensure steady convergence and fair comparison between experiments, the number of training epochs of the optimizer was fixed at 500. Selected configurations were used to train the MLP for 450 epochs. These were the number of hidden units, the learning rate, the batch size, and the number of epochs used for training. To make sure that the same results were generated each time, a fixed random seed (seed = 42) was used for all experiments, including initialization of weights, data shuffling, and optimization.

Table 1: Hyperparameter search space and optimal values for optimizing the MLP model by the BBO, GA, and GOA optimization algorithms.

MLP		BBO	GA	GOA
n_hidden_units	[2, 64]	8	16	8
epoch	[100, 1000]	600	650	450
batch_size	[2, 32]	16	8	8
learning_rate	[0.0001, 1]	0.1	0.05	0.01

2.5 Data collection and preparing

Incorporating the number of volumes, as well as the Open, High, Low, and Close prices, within a specific time frame is crucial for conducting a thorough analysis. The data for this investigation were collected from the Hang Seng Index (HSI), ranging from January 2, 2015, to June 29, 2023. The HSI is a famous market index that tracks a selection of major companies featured on this Exchange. The HSI was chosen as the case study because of its substantial worldwide influence and high volatility. The HSI is a prominent market index in Asia that tracks the performance of large Hong Kong-listed companies, many of which are active in global markets. A complex and dynamic benchmark for stock market prediction, the index is extremely sensitive to regional policy changes, international economic trends, and geopolitical tensions. These qualities offer a demanding testing environment for assessing the resilience and versatility of forecasting models such as the suggested GOA-MLP. The HSI comprises a wide variety of prominent Hong Kong companies. It includes finance, real estate, technology, telecommunications, manufacturing, etc. Market capitalization determines the weight of each company's stock in the index. The HSI of the Hong Kong stock market closely tracks the domestic economy. This index is widely used to assess Hong Kong's financial health and investor outlook. Index fluctuations might affect investor perceptions of regional economic conditions. The index includes many multinational companies with a worldwide reach. The index's global reach makes it a key economic and market indicator beyond Hong Kong. To represent the changing Hong Kong market, the HSI constituents are continuously examined and adjusted. The index may reject companies that don't meet its standards and include new ones that do. The HSI may be affected by market laws, economic conditions, government policies, and global events. In short, the HSI tracks the major companies listed on the HSI Exchanges. Economic data is insightful to the Hong Kong Economic Index, sets standards for investors, and helps to understand the mood and trend of the stock market. OHLC price and volume data were supplied to the model as training data. One of the key aspects to achieving the success of ML models is proper data preparation. Data cleaning is the first step in the process, where errors, missing values, and inconsistencies are identified and corrected in the dataset. It is a crucial process because poor-quality data can affect model performance and result in incorrect predictions. The overall accuracy and resilience of the model are greatly improved by ascertaining and fine-tuning the dataset exhaustively before training. Designing leading ML models starts with a clean, stable dataset. Normalization is then used after cleaning to normalize the input features into a similar order of magnitude, usually between 0 and 1. By causing all features to make a similar contribution to the learning process, this normalization stops models from favoring variables whose scales have larger values. Additionally, normalization speeds up model convergence and improves its ability to identify inherent patterns. Normalization is done through the following equation [45]:

$$Xscaled = \frac{(X - Xmin)}{(Xmax - Xmin)} \tag{12}$$

Finally, data splitting is used for testing the model's generalization ability. Data is split into two sets: 20% is left for testing, and the remaining 80% is utilized to train the model. This portion is used to make sure the model acquires generalizable patterns and does not memorize the training set by preventing overfitting and assisting in the measurement of the model's performance against new data.

3 Results and discussions

3.1 Assessment metrics

The accuracy of the projections was appraised using several performance criteria.

The R^2 scores are quantitative measures assessing to what extent a model accurately representing the data. When the score tends to become 1, the quality of the model increases; hence this measure is used to analyze the accuracy of the model [46].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{13}$$

Difference of data observed with data expected is calculated with the MSE (Mean Squared Error). To get the numerical value, one must first identify the square of the difference between the observed and expected values. Then, estimate the average of all squared differences. The figure given displays the accuracy of the model, with a lower MSE indicating higher accuracy [47].

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{k}=0}^{\mathrm{n}} \binom{\mathrm{n}}{\mathrm{k}} (\mathrm{Fi} - \mathrm{Yi})\mathrm{b}^2 \tag{14}$$

The Mean Absolute Error (MAE) is a complementary statistic that we may consider to measure how far an observation is from the prediction. The performance of the model depends upon this statistic; the MAE should go down for better accuracy [47].

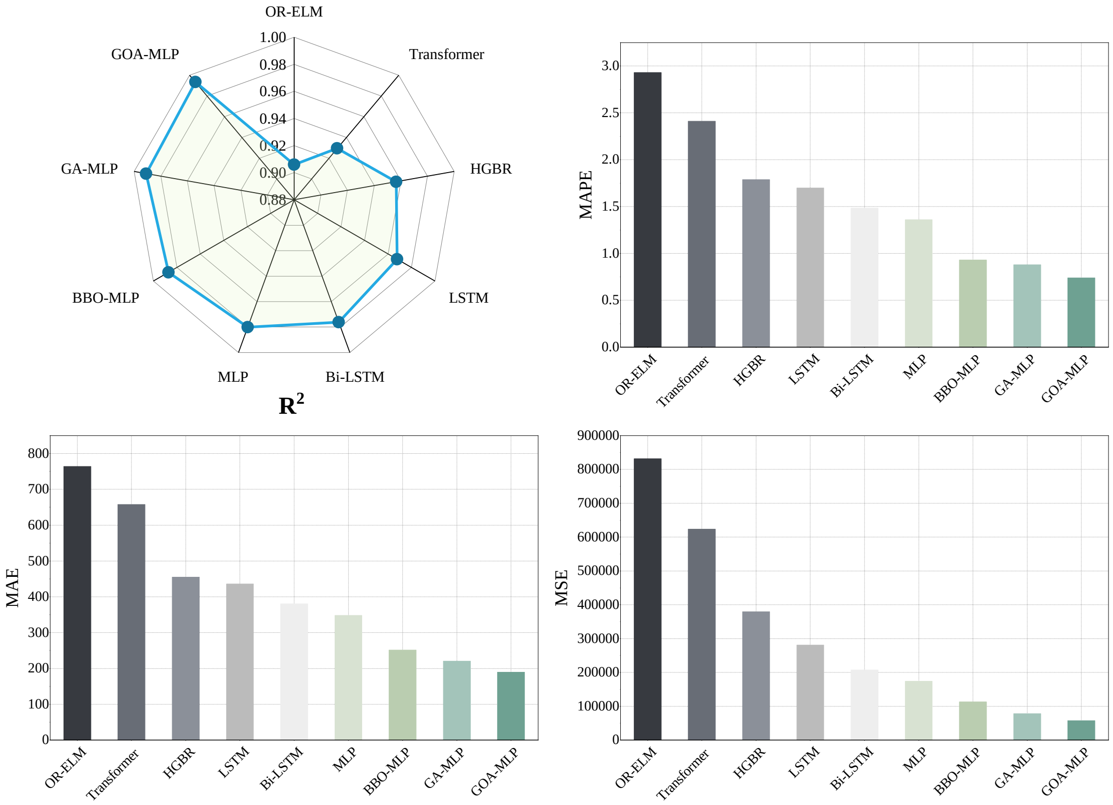
$$MAE = \frac{\sum_{i=1}^{n}|y_i - \hat{y}_i|}{n} \tag{15}$$

A method of checking a model for precision would be to use MAPE, or Mean Absolute Percentage Error. To determine the percentage difference between predicted and actual values, first an absolute difference is determined and then divided by the value of observation. These percentages are then subjected to an averaging operation. It must be clear that this number aims at analyzing the accuracy of the model. Less MAPE will mean more precision [46].

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \times 100 \tag{16}$$

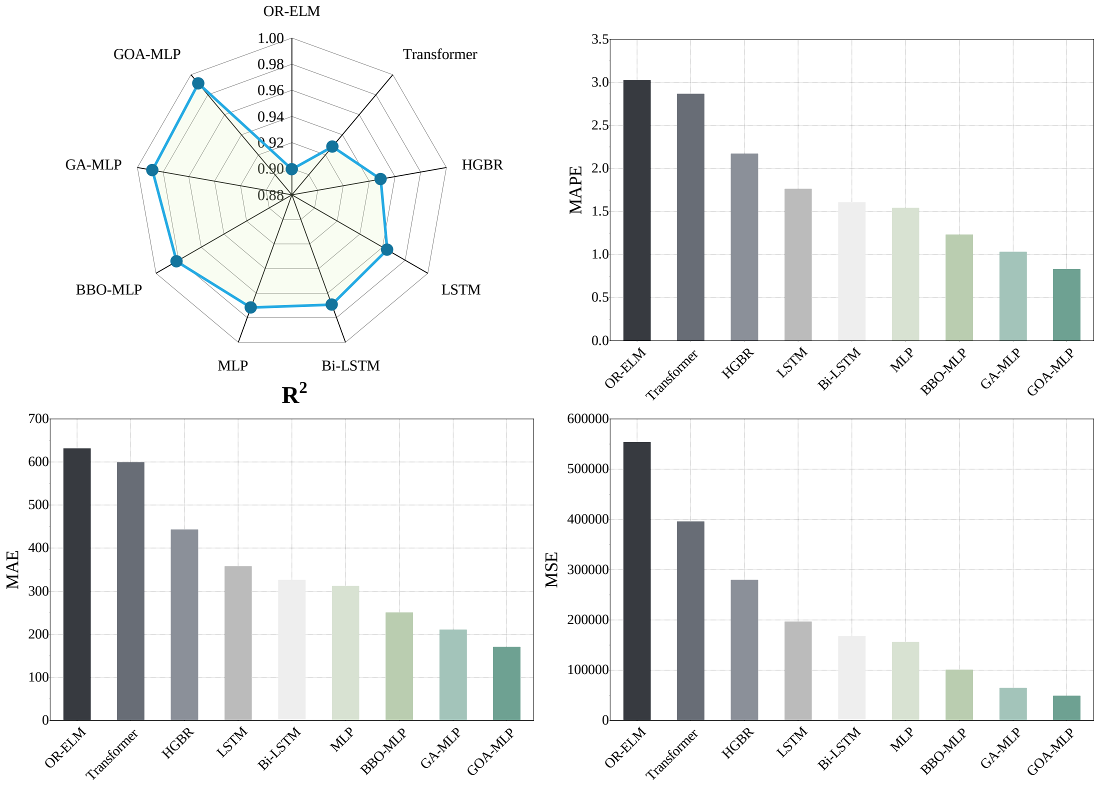
3.2 Analysis and comparisons

The primary objective of the investigation is to create and assess the best hybrid algorithm available for stock price prediction. Predictive models have been developed as an outcome of extensive research on the intricate aspects influencing stock market movements. This was carried out in order to give analysts and investors trustworthy information so that they could make well-informed financial decisions. A comprehensive analysis of the performance of each method is presented in Table 2, and Figs. 5 and 6, respectively.



(a)

Figure 5: Values for the assessment metrics during training set for the models.



(b)

Figure 6: Values for the assessment metrics during testing set for the models.

A comprehensive comparison of various prediction models, sophisticated deep learning architectures, and traditional machine learning models is discussed in Table 2. Four evaluation metrics R^2, MAPE, MAE, and MSE are used to analyze the model's performance over training and testing datasets. A comprehensive assessment framework of this nature ensures that the analysis includes the error magnitude, generalization ability, and point prediction accuracy. The relatively low R^2 values ($\leq$ 0.9565) and higher error measures of the OR-ELM and HGBR models indicate relatively poor predictive performance. Although computationally efficient, these methods' use of shallow learning architectures restricts their ability to model the complex nonlinear dependencies, chaotic volatility, and long-range temporal dependencies of financial time series. While the Transformer model is theoretically powerful in modeling sequential dependencies using self-attention mechanisms, it only achieves modest improvement over conventional approaches (R^2 = 0.9282, test MAPE = 2.86). This is probably because it is susceptible to data paucity and over-parameterization in noisy, nonstationary settings like equity markets. By efficient vanishing gradient reduction and making use of temporal context, the LSTM and Bi-LSTM architectures that are capable of learning long-term dependencies surpass the common baselines. By using both past and future contextual information within sequence modeling, the bidirectional version, Bi-LSTM, provides a higher predictive accuracy (test R^2 = 0.9694, MAPE = 1.60) than its unidirectional LSTM. However, the recurrent models have higher error rates than the optimization-enhanced MLP variants, indicating suboptimal convergence and possible overfitting to temporal noise even with enhanced sequential modeling. The baseline MLP also learns competitively with nonlinear transformations in deep hidden layers (test R^2 = 0.9717, MAPE = 1.54), but still retains its architecture and learning parameters fixed by conventional tuning that remains vulnerable to shallow local minima. Metaheuristic optimization significantly improves prediction performance: both BBO-MLP (test R^2 = 0.9817, MAPE = 1.23) and GA-MLP (test R^2 = 0.9883, MAPE = 1.03) have dramatic error reductions against the baseline MLP. These improvements reflect the contribution of global search heuristics to overcoming the multimodal, high-dimensional hyperparameter space that comes with optimizing neural networks. With test R^2 = 0.9912, MAPE = 0.83, MAE = 170.06, and MSE = 48,618, the GOA-MLP hybrid performs best on all four metrics. GOA-MLP significantly improves predictive precision and robustness by cutting test MAPE by about 48.1% and MAE by 47.9% against the best deep learning reference (Bi-LSTM). GOA-MLP demonstrates its optimization advantage by achieving additional reductions in MAPE (19.4%) and MAE (19.1%) even when compared with the performing competitor hybrid (GA-MLP).

Table 2: The outcomes of the methodologies

MODEL/Metrics	TRAIN SET				TEST SET			
	R^2	MAPE	MAE	MSE	R^2	MAPE	MAE	MSE
OR-ELM	0.9059	2.93	763.74	831725	0.8996	3.02	631.14	553422
Transformer	0.9294	2.41	658.17	623476	0.9282	2.86	598.90	395837
HGBR	0.9565	1.79	454.76	379568	0.9489	2.17	443.11	279132
LSTM	0.9678	1.70	435.90	281124	0.9641	1.76	357.76	196239
Bi-LSTM	0.9762	1.48	380.36	207730	0.9694	1.60	326.02	167418
MLP	0.9801	1.36	348.22	174192	0.9717	1.54	311.72	155726
BBO-MLP	0.9872	0.93	251.74	112962	0.9817	1.23	250.39	100199
GA-MLP	0.991	0.88	220.83	78593	0.9883	1.03	210.27	64366
GOA-MLP	0.9934	0.74	189.77	57231	0.9912	0.83	170.06	48618

Several algorithmic features in combination give rise to GOA-MLP's enhanced performance:

During the optimization time frame, GOA utilizes an adaptive nonlinear decreasing coefficient to control the transition from exploration (global searching) to exploitation (local refinement). This enables more vigorous exploration of the hyperparameter space and avoids premature convergence. The intragroup weight convergence dynamics of the MLP are directly influenced by the upper-level hyperparameters (number of hidden units, learning rate, batch size, and epochs) tuned for by GOA. This leads to improved generalization and less overfitting in noisy environments.

GOA uses the position-update approach with inspiration drawn from the behavior of grasshopper swarms that adaptively modifies candidate solutions based on fluctuations in the fitness landscape. This enables diversity maintenance while focusing on optimal regions. With fewer hidden units, fewer epochs of training, limited batch size, and a moderate learning rate, the final GOA-selected configuration reduces variance in parameter updates to produce smoother convergence curves and better performance on novel test data.

This exhaustive benchmark test shows GOA-MLP to far outperform competing metaheuristic-optimized neural networks as well as deep recurrent and standard recurrent frameworks. The ability of the model to navigate the volatility and structural complexity of financial markets with greater predictive accuracy arises due to the synergistic combination of swarm-intelligence-driven global search with MLPs' representational strength. Designing and validating a hybrid GOA–MLP model that can provide quantifiable gains over state-of-the-art (SOTA) stock price forecasting techniques was the main objective of this study. In comparison to the top-performing benchmark models, this study specifically sought to reduce MAPE and MAE by at least 15%. The GOA–MLP model surpassed this benchmark, as shown by the results and detailed in Table 2. It reduced MAPE and MAE by 19.4% and 19.1%, respectively, over GA–MLP and by approximately 48.1% and 47.9% when compared to the top deep learning baseline (Bi-LSTM) while maintaining a high R^2 of 0.9912 on the test set. Another goal was to improve resilience against volatile market conditions by leveraging GOA's adaptive exploration–exploitation mechanism to find hyperparameter settings that inhibit overfitting and induce more rapid convergence. Figure 7 and 8 demonstrate how the suggested approach may accurately forecast the stock market.

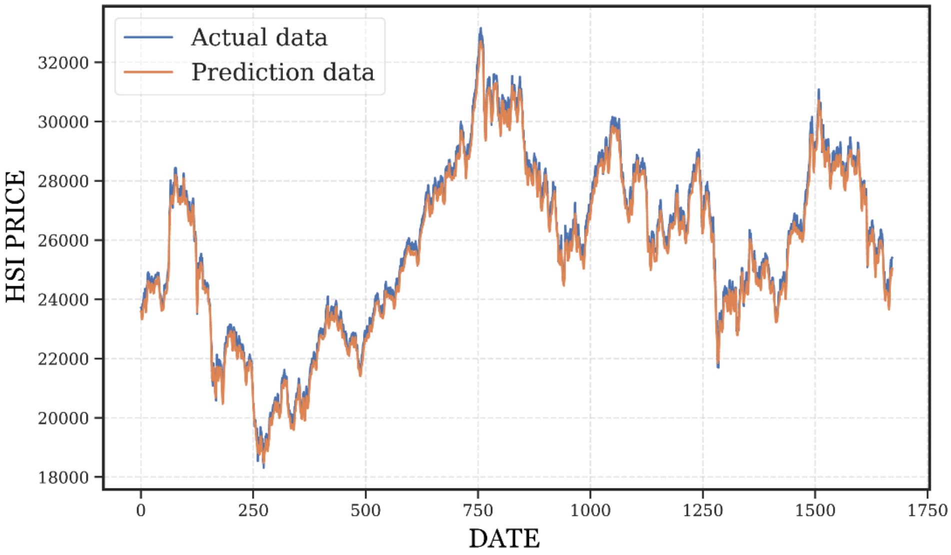


Figure 7: The GOA-MLP train data forecast curve.

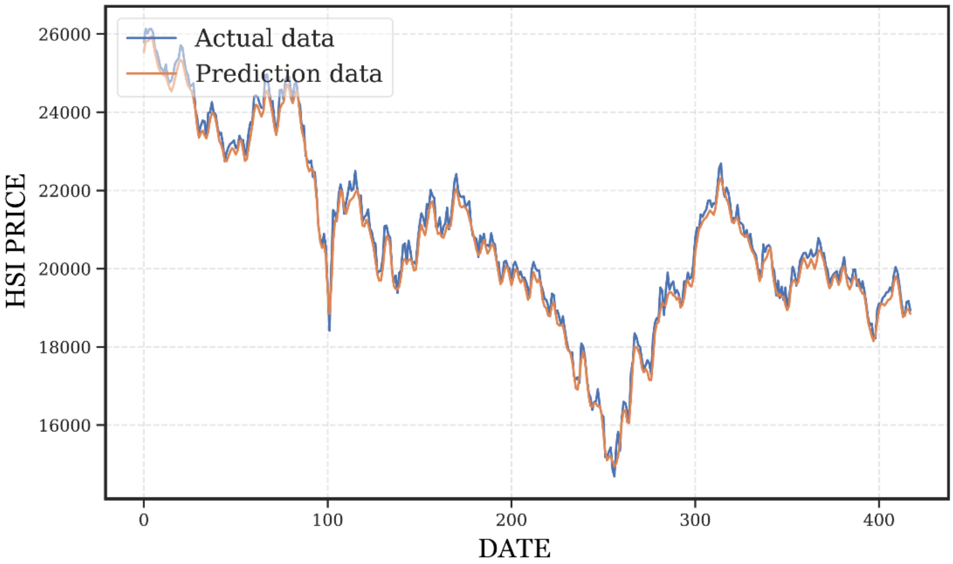


Figure 8: The GOA-MLP test data forecast curve.

3.3 Real-world back testing and risk analysis

A real-world back testing experiment was carried out to evaluate the suggested GOA–MLP model's practical usefulness beyond standard predictive accuracy metrics. To manage a simulated portfolio during the test period, the model's daily predictions were translated into trading signals, and the results were compared to a conventional buy and hold strategy on the same asset. The Sharpe ratio, maximum drawdown (%), and cumulative return (%) were the three-evaluation metrics taken into account. Together, these metrics evaluate risk-adjusted returns, profitability, and downside risk, providing a more thorough assessment of investment performance. According to Table 3, the trading strategy based on GOA and MLP produced a

Sharpe ratio of 3.14, a maximum drawdown of only 1.7%, and a cumulative return of 96.52%. Profitability and effective risk management are demonstrated by the high return and low drawdown. Conversely, the buy and hold strategy had a Sharpe ratio of -0.55, a maximum drawdown of 43.81%, and a negative cumulative return of -27.55% percent, all of which indicated unfavorable risk-return characteristics. These variations are further highlighted by the portfolio value trajectories displayed in Figure 9.

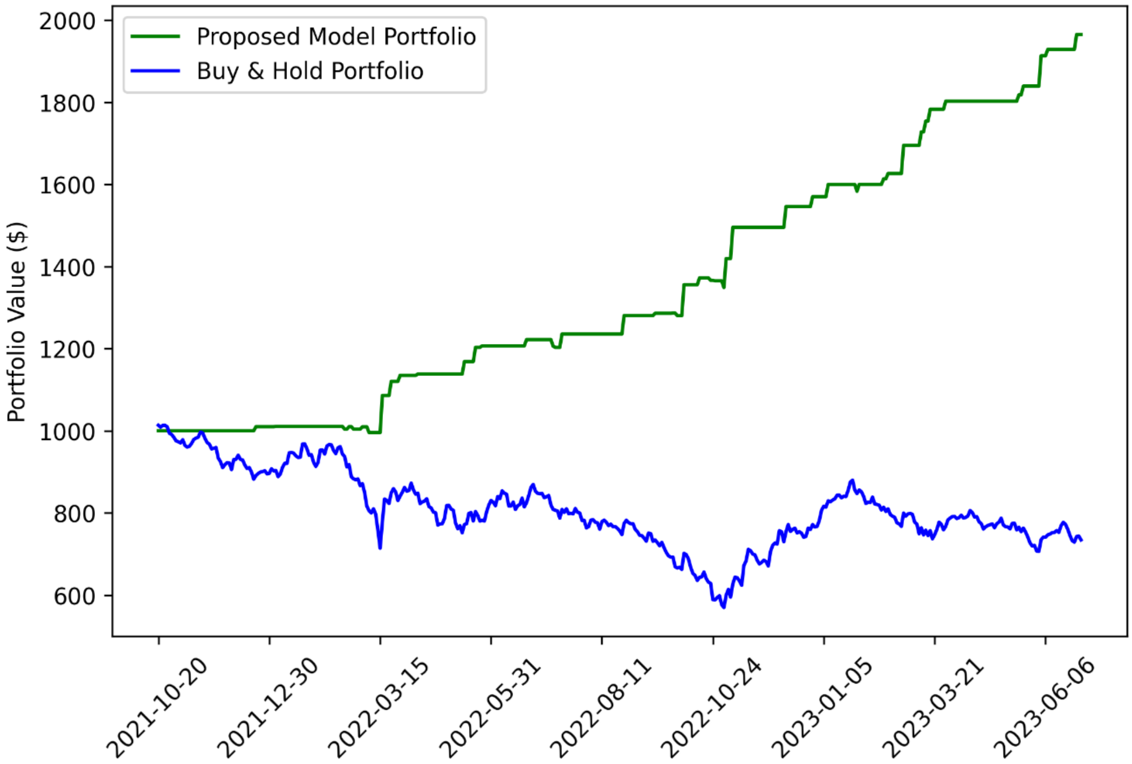


Figure 9. Trajectories of the portfolio values for the buy-and-hold strategy and the suggested GOA–MLP-based trading strategy during the test period

Throughout the back test period, the GOA–MLP strategy avoided significant capital drawdowns and continued to grow steadily, whereas the buy and hold strategy saw protracted declines and was unable to return to its starting point. These results show that in addition to achieving greater statistical predictive accuracy, the proposed GOA–MLP model also translates these gains into tangible investment benefits, such as appreciable improvements in risk-adjusted returns and capital preservation in volatile market conditions.

Table 3: Results of a back test comparing the suggested GOA–MLP-based trading strategy with a traditional buy and hold strategy during the test period

Strategy	**Cumulative Return (%)**	**Max Drawdown (%)**	**Sharpe Ratio**
Model-Based	96.52	1.7	3.14
Buy & Hold	-27.55	43.81	-0.55

4 Conclusion

For very long, prediction of stock prices has been a favorite expanse of investigation. Investors have been building ever-so-more accurate forecasting models to gather big bucks. Predicting stock market movements becomes really strenuous when there are multiple sources of uncertainty like legislative enactments and sociocultural situations including pandemics. For making correct forecasts, it would be essential to come to terms with the stochastic and non-linear aspects of the market. Fortunately, the GOA-MLP stands as a high accurate model and thus can aptly address the presented problems. Inclusively presented in this paper are the MLP, BBO-MLP, and GA-MLP models for stock price forecasting. OHLC price and volume data from the HSI shares were part of the data utilized for the research project. The dataset is from the start of 2015 to the end of 2023, covering a particular time interval. The analysis, while forecasting stock prices, reveals that the GOA-MLP model is highly competent in forecasting, with both excellent performance and consistent results.

- During the research, a comparative analysis was conducted to evaluate the GOA-MLP model in terms of its accuracy and ability to make accurate predictions compared to other models. The findings of this study provide evidence that the GOA-MLP model consistently outperforms other models. According to the test results, the average score of MSE shows that the level of accuracy in predictions is good. The average MAPE score of the model was 0.83, which shows that it had significant accuracy in its predictions during the study. Both the high R^2 value of 0.9912 and the low MAE score of 170.06 can be taken as evidence that the predictions are based on an accurate and consistent basis. Compared to other investigated models, the GOA-MLP model has shown higher performance in terms of accuracy and efficiency.

As mentioned earlier, GOA-MLP shows better performance compared to other models in competitive conditions. Further validation of the effectiveness of the suggested model in accurately and comprehensively forecasting volatility in the stock market is provided by the findings. By utilizing this approach, risk minimization is made easier, and investors are provided with the opportunity to make informed investment selections by analyzing the many data points.

Acknowledgments

This work is supported by the key project of Teaching Quality and Teaching Reform in Higher Education Institutions of Anhui Provincial Department of Education: Provincial first-class undergraduate construction site in accounting (No. 2020ylzy002), and accounting professional transformation and enhancement project (No. 2022zygzts036).

Ethical approval

The research paper has received ethical approval from the institutional review board, ensuring the protection of participants' rights and compliance with the relevant ethical guidelines.

References

[1] R. G. Ahangar, M. Yahyazadehfar, and H. Pournaghshband, "The comparison of methods artificial neural network with linear regression using specific variables for prediction stock price in Tehran stock exchange," *arXiv preprint arXiv:1003.1457*, Cornell University, 2010. https://doi.org/10.48550/arXiv.1003.1457.

[2] P. Chhajer, M. Shah, and A. Kshirsagar, "The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction," *Decision Analytics Journal*, Elsevier, vol. 2, p. 100015, 2022. https://doi.org/10.1016/j.dajour.2021.100015.

[3] N. Buduma and N. Locascio, "Deep Learning," 2017.

[4] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Appl Soft Comput*, Elsevier, vol. 11, no. 6, pp. 4135–4151, 2011. https://doi.org/10.1016/j.asoc.2011.02.032.

[5] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf Sci (N Y)*, Elsevier, vol. 237, pp. 82–117, 2013. https://doi.org/10.1016/j.ins.2013.02.041.

[6] A. R. Simpson, G. C. Dandy, and L. J. Murphy, "Genetic algorithms compared to other techniques for pipe optimization," *J Water Resour Plan Manag*, ASCE Library, vol. 120, no. 4, pp. 423–443, 1994. https://doi.org/10.1061/(ASCE)0733-9496(1994)120:4(423).

[7] T. Back, *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[8] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[9] S. Mirjalili, "The ant lion optimizer," *Advances in engineering software*, Elsevier, vol. 83, pp. 80–98, 2015. https://doi.org/10.1016/j.advengsoft.2015.01.010.

[10] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Generation Computer Systems*, Elsevier, vol. 111, pp. 300–323, 2020. https://doi.org/10.1016/j.future.2020.03.055.

[11] D. Simon, "Biogeography-based optimization," *IEEE transactions on evolutionary computation*, IEEE, vol. 12, no. 6, pp. 702–713, 2008. https://doi.org/10.1109/TEVC.2008.919004.

[12] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl Based Syst*, Elsevier, vol. 89, pp. 228–249, 2015. https://doi.org/10.1016/j.knosys.2015.07.006.

[13] S. J. Simpson, A. R. McCaffery, and B. F. Hägele, "A behavioural analysis of phase change in the desert locust," *Biological reviews*, Cambridge, vol. 74, no. 4, pp. 461–480, 1999. https://doi.org/10.1017/S000632319900540X.

[14] S. M. Rogers, T. Matheson, E. Despland, T. Dodgson, M. Burrows, and S. J. Simpson, "Mechanosensory-induced behavioural gregarization in the desert locust Schistocerca gregaria," *Journal of Experimental Biology*, Journal of Experimental Biology, vol. 206, no. 22, pp. 3991–4002, 2003. https://doi.org/10.1242/jeb.00648.

[15] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: theory and application," *Advances in engineering software*, Elsevier, vol. 105, pp. 30–47, 2017. https://doi.org/10.1016/j.advengsoft.2017.01.004.

[16] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evol Comput*, MIT Press Direct, vol. 4,

no. 1, pp. 1–32, 1996. https://doi.org/10.1162/evco.1996.4.1.1.

[17] J. H. Holland, "Genetic Algorithms Computer programs that" evolve" in ways that resemble natural selection can solve complex problems even their creators do not fully understand," *Sci Am*, pp. 66–72, 1992.

[18] J. Luo, H. Chen, Y. Xu, H. Huang, and X. Zhao, "An improved grasshopper optimization algorithm with application to financial stress prediction," *Appl Math Model*, Elsevier, vol. 64, pp. 654–668, 2018. https://doi.org/10.1016/j.apm.2018.07.044.

[19] X. Xiang, X. Ma, M. Ma, W. Wu, and L. Yu, "Research and application of novel Euler polynomial-driven grey model for short-term PM10 forecasting," *Grey Systems: Theory and Application*, Emerald, vol. 11, no. 3, pp. 498–517, 2021. https://doi.org/10.1108/GS-02-2020-0023.

[20] S. Łukasik, P. A. Kowalski, M. Charytanowicz, and P. Kulczycki, "Data clustering with grasshopper optimization algorithm," in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Prague, Czech Republic, IEEE, 2017, pp. 71–74. https://doi.org/10.15439/2017F340.

[21] A. Fathy, "Recent meta-heuristic grasshopper optimization algorithm for optimal reconfiguration of partially shaded PV array," *Solar Energy*, Elsevier, vol. 171, pp. 638–651, 2018. https://doi.org/10.1016/j.solener.2018.07.014.

[22] M. Ahanch, M. S. Asasi, and M. S. Amiri, "A Grasshopper Optimization Algorithm to solve optimal distribution system reconfiguration and distributed generation placement problem," in *2017 IEEE 4th international conference on knowledge-based engineering and innovation (KBEI)*, Tehran, Iran, IEEE, 2017, pp. 659–666. https://doi.org/10.1109/KBEI.2017.8324880.

[23] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk, and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, Springer, vol. 51, pp. 1531–1551, 2021. https://doi.org/10.1007/s10489-020-01893-z.

[24] M. Ehteram *et al.*, "Design of a hybrid ANN multi-objective whale algorithm for suspended sediment load prediction," *Environmental Science and Pollution Research*, Springer, vol. 28, pp. 1596–1611, 2021. https://doi.org/10.1007/s11356-020-10421-y.

[25] F. B. Banadkooki, M. Ehteram, F. Panahi, S. S. Sammen, F. B. Othman, and E.-S. Ahmed, "Estimation of total dissolved solids (TDS) using new hybrid machine learning models," *J Hydrol (Amst)*, Elsevier, vol. 587, p. 124989, 2020. https://doi.org/10.1016/j.jhydrol.2020.124989.

[26] F. B. Banadkooki *et al.*, "Enhancement of groundwater-level prediction using an integrated machine learning model optimized by whale algorithm," *Natural resources research*, Springer, vol. 29, pp. 3233–3252, 2020. https://doi.org/10.1007/s11053-020-09634-2.

[27] A. Seifi, M. Ehteram, V. P. Singh, and A. Mosavi, "Modeling and uncertainty analysis of groundwater level using six evolutionary optimization algorithms hybridized with ANFIS, SVM, and ANN," *Sustainability*, MDPI, vol. 12, no. 10, p. 4023, 2020. https://doi.org/10.3390/su12104023.

[28] S. M. J. Jalali, R. Hedjam, A. Khosravi, A. A. Heidari, S. Mirjalili, and S. Nahavandi, "Autonomous robot navigation using moth-flame-based neuroevolution," *Evolutionary Machine Learning Techniques: Algorithms and Applications*, Springer, pp. 67–83, 2020. https://doi.org/10.1007/978-981-32-9990-0_5.

Modified Dwarf Mongoose Optimization for Feature Selection in Imbalanced Student Performance Prediction Tasks

Zhongxia Liu
Modern Education Technology Center, Pingdingshan University, Pingdingshan 467000, Henan, China
E-mail: midsummer0218@163.com

Keywords: Student performance prediction, feature selection, dwarf mongoose, educational data mining, wrapper-based learning

Received: July 13, 2025

Student performance prediction through Educational Data Mining (EDM) methods has become increasingly critical to educational decision-making and intervention. But educational datasets are high-dimensional and imbalanced, presenting serious problems for standard machine learning models. This paper presents an innovative feature selection methodology based on the Modified Dwarf Mongoose Optimization (MDMO), an enhanced version of standard DMO by adding three essential components: adaptive alpha guidance, scout-based diversity, and enhanced babysitter exchange criteria. These modifications boost the exploration-exploitation balance and prevent premature convergence, enabling more efficient search in high-dimensional binary feature spaces. The proposed MDMO is integrated as a wrapper method with five popular classifiers, LogitBoost, linear discriminant analysis, naive bayes, k-nearest neighbors, and decision trees, to form a robust predictive model for student performance. The proposed MDMO was evaluated on two public educational datasets (Gazi University course repetition data and Portuguese secondary school grade data). On Data1, it achieved an AUC of 0.672 with compact subsets of ~12 features; on Data2, it reached an AUC of 0.929 with ~13 selected features. Compared with state-of-the-art baselines such as BTLBO-LDA and MLP-Adam, MDMO consistently demonstrated higher accuracy and more efficient feature selection. Adaptive alpha guidance dynamically adjusts the leader to strengthen exploitation, whereas enhanced babysitter exchange preserves diversity, contributing to robust handling of class imbalance.

Povzetek: Članek predstavi napoved uspeha študentov z izbiro značilk na osnovi modificirane "Dwarf Mongoose" optimizacije, uporabljene kot ovijalni pristop za učinkovito obravnavo visoko-dimenzionalnih, neuravnoteženih podatkov.

1 Introduction

Student academic performance prediction is essential in Educational Data Mining (EDM), empowering institutions to recognize at-risk students, optimize resource distribution, and improve learning outcomes [1]. Educational systems are moving progressively towards online platforms, where large quantities of student data, such as attendance records, grades, behavioral reports, and demographic details, are being created [2]. As demonstrated in Figure 1, the EDM process begins with the learning environment and proceeds through the raw data aggregation and subsequent preprocessing, followed by transformation for structured analysis. Data Mining (DM) identifies significant patterns, which are then interpreted to derive valuable insights [3]. These insights allow educators and school administrators to intervene promptly, match learning strategies to students' needs, and inform evidence-based education policy. Sound predictive modeling significantly lowers dropout levels, customizes learning pathways, and enhances education systems' planning [4].

Despite having large educational datasets, deriving functional patterns from them is still challenging due to the high dimensionality and class imbalance. Most student performance datasets have many features that are not useful for prediction [5]. Moreover, the learning model is influenced by the imbalance between the number of passing and failing students in most datasets. Irrelevant and redundant features lower classifier accuracy and complicate computation [6]. The selection of features is therefore of the utmost importance. Feature selection reduces dimensionality and enhances the interpretability and generalizability of the model, which are necessary in high-stakes educational decision-making situations [7].

Over the last few years, several metaheuristic paradigms have gained popularity due to their capacity for resolving feature selection issues in high-dimensional space. Methods like Particle Swarm Optimization (PSO) [8], Genetic Algorithms (GA) [9], Ant Colony Optimization (ACO) [10], and the Whale Optimization Algorithm (WOA) [11] have shown encouraging performance in balancing exploration and exploitation to identify optimal subsets of features. These methods are usually utilized in wrapper models where the metaheuristic algorithm is combined with classifiers through iteration to determine the most appropriate feature

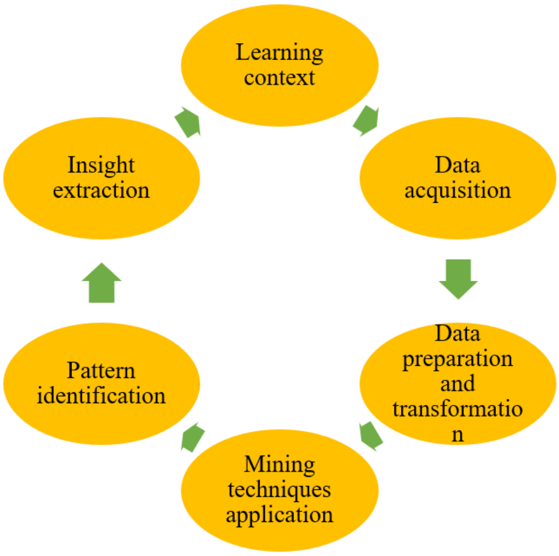


Figure 1: EDM workflow

combinations. Extensions, including chaotic maps, adaptive strategies, and hybridization, also support their performance. Most of these works consider general datasets with less attention to domain-specific education problems.

While swarm intelligence techniques such as WOA have seen extensive usage in learning environments, the Dwarf Mongoose Optimization (DMO) algorithm has yet to be extensively researched, particularly in student performance prediction. Conceived initially to simulate ecological activities, DMO presents an innovative role-based organization that draws upon the social life of dwarf mongooses [12]. However, its baseline version lacks mechanisms to effectively manage exploration-exploitation trade-offs in complex binary feature selection tasks. No study has adapted and enhanced DMO for binary classification or integrated it into wrapper-based frameworks for imbalanced educational datasets. This gap presents an opportunity to leverage the strengths of DMO through algorithmic modifications tailored to EDM applications.

To fill in this void, we present herein the Modified Dwarf Mongoose Optimization (MDMO) algorithm with three significant improvements: adaptive control of alpha for directing the exploration process, scout-based randomization to promote diversity, and an improved babysitter exchange condition for well-balanced role rotation. We embed this modified algorithm within a wrapper feature selection framework and integrate it with five popular classifiers: LogitBoost, linear discriminant analysis, naive Bayes, k-nearest neighbors, and decision trees. Furthermore, we apply the adaptive synthetic sampling method to mitigate the class imbalance problem. The proposed model is evaluated on two real-world educational datasets and benchmarked against other state-of-the-art metaheuristics.

2 Literature review

Turabieh, et al. [13] introduced an improved Harris Hawks Optimization (HHO) algorithm for student performance prediction through dynamic population diversity management. The algorithm uses k-Nearest Neighbors (k-NN) clustering to identify premature convergence and has an injection approach upon population collapse into one cluster. Kamal, et al. [14] employed Relief for feature selection in combination with machine learning classifiers such as Backpropagation Neural Networks (BPNN), Random Forests (RF), and NB. The research seeks to classify and forecast student performance using several educational indicators.

Apriyadi and Rini [15] utilized metaheuristic optimization methods, PSO and GA, in optimizing Support Vector Regression (SVR) hyperparameters for student performance prediction modeling. PSVR and GSVR models were proposed and contrasted with the conventional models NB, NN, and RF using RMSE as an evaluation metric. Song [16] suggested a new integration of the k-NN classifier with two bio-inspired techniques, the Honey Badger Algorithm (HBA) and the Arithmetic Optimization Algorithm (AOA), to maximize math performance prediction. The integrated models worked impressively well in classification accuracy, predicting the first and third-term math grades (G1 and G3), with high precision values well above 0.90.

Ma [17] created an optimization-prediction model using RF with Electric Charged Particles Optimization (ECPO) and Artificial Rabbits Optimization (ARO) for improved student performance prediction. The model processed an extensive dataset of 4424 students and prioritized dimensionality reduction. Shou and Lu [18] examined the integration of the DMO algorithm with Support Vector Classification (SVC) to enhance student performance prediction. The introduced SVDM model performed better than other combined models with substantial improvements in crucial parameters, including Accuracy (0.929), Precision (0.931), Recall (0.929), and F1-score (0.927) in predicting grades for academics.

Ye, et al. [19] proposed CQFOA-KELM as an innovative hybrid optimizer combining Covariance Matrix Adaptation Evolution Strategy (CMAES) and Quadratic Approximation (QA) with Fruit-fly Optimization Algorithm (FOA) in Kernel-based Extreme Learning Machine (KELM). On implementation in a high-capacity survey database, the model attained an accuracy of 98.15%.

As shown in Table 1, although the reviewed works successfully employ various metaheuristic solvers to make student performance predictions, there are some limitations. Most rely on general-purpose optimizers such as PSO, GA, or WOA variants, ignoring biologically inspired strategies such as DMO. While DMO has been utilized recently with SVC, its potential is not yet fully unleashed, especially in binary feature selection in wrapper methodologies. In addition, none of the discussed works deeply delves into the role-based cooperation behavior inherent in DMO, which can better manage exploration and exploitation. This work fills the void by strengthening DMO via adaptive alpha control, scout-led diversity, and optimized babysitter exchange. We also couple the improved DMO with several classifiers and use adaptive synthetic techniques to deal with data imbalance.

Table 1: An overview of relevant studies

Ref	Optimization algorithms	Method	Dataset	Accuracy	Achievement	Shortcoming
[13]	Modified harris hawks optimization	k-NN, LRNN, NB, and ANN	Student dataset (survey)	90%	High accuracy with LRNN; dynamic control of population	Risk of overfitting with deep models; relies on clustering
[14]	Relief	BPNN, RF, and NB	Educational logs	85%	BPNN achieved the best accuracy among the tested models	No optimization applied; fixed feature selector
[15]	Particle swarm optimization and genetic	SVR	Exam scores	RMSE is the lowest among the tested	Achieved the lowest RMSE, outperforming other models	Focuses only on regression; limited interpretability
[16]	Honey badger and arithmetic optimization	K-NN	Math performance	92%	KNHB achieved ~92% accuracy and precision in math prediction	Domain-limited; no explicit FS mechanism reported
[17]	Electric charged particles optimization and artificial rabbits optimization	RF	Performance data	Aligns with ground-truth	Aligned well with actual performance data	Lacks generalizability; focuses on one classifier
[18]	Dwarf mongoose optimization	SVC	Survey dataset	90%	Improved all evaluation metrics	DMO applied without structural enhancement or refinement
[19]	Covariance matrix adaptation evolution strategy, and quadratic approximation	KELM	Surveys	98.1%	98.15% accuracy; identified key performance factors	The dataset is limited to surveys; complex hybrid hard to generalize

3 Materials and methods

3.1 Modified dwarf mongoose optimization

To enhance the trade-off between exploration and exploitation of the original DMO algorithm, an improved variant of DMO is designed for this research. This enhanced model draws inspiration from the sophisticated social and survival habits of dwarf mongoose colonies. These creatures have social living habits where territories are marked to guard resources and ensure safety in numbers. Unlike other animals that expand their numbers to exploit resources, dwarf mongooses consciously limit their number for sustainable survival and risk minimization. These animals perform search efficiently and accurately, where attacks on predators usually start with a substantial hit from the head, and then they forage for food across vast distances. These animals are half-nomadic and hardly visit the same shelter twice, indicating tactical movement and flexibility.

Socially, the dwarf mongoose follows a structured hierarchy with specialized roles. The dominant alpha pair (usually one male and one female) oversees the group, while scout members are tasked with exploration, and babysitters care for the young. Vocal communications, primarily initiated by the alpha female, help coordinate group actions and signal danger. Reproductive privileges are restricted to the alpha female, reinforcing a disciplined caste system. Group size and structure are optimized based on ecological needs, balancing individual success and collective efficiency. These natural behaviors have inspired the DMO framework for solving complex optimization problems. The MDMO enhances the basic DMO with three key innovations:

- Alpha role selection: Unlike traditional DMO, which relies on probabilistic evaluation, MDMO deterministically selects the most optimal individual as the alpha based on the best fitness score. A dynamic movement regulator is introduced to control the alpha's search pattern, improving convergence and diversity.
- Scout exploration enhancement: To diversify search trajectories, the scout mechanism is upgraded with stochastic behaviors, increasing the algorithm's ability to escape local optima and discover new promising regions.
- Babysitter replacement strategy: A revised mechanism replaces underperforming babysitters. When criteria are met, new replacements exchange information with their predecessors, learning from their knowledge of the environment to improve population quality.

The MDMO algorithm operates through three distinct and interrelated components: the alpha team, the scout team, and the babysitter group. Before activating these behavioral roles, the algorithm initializes a population of candidate solutions. The optimization process begins by generating an initial population matrix S, where each row represents a solution candidate (mongoose) in the search space. This step is mathematically described in Eq. 1 and Eq. 2.

$$S = \begin{bmatrix} s_{1,1} & \cdots & s_{1,D} \\ \vdots & \ddots & \vdots \\ s_{N,1} & \cdots & s_{N,D} \end{bmatrix} \tag{1}$$

$$s_{i,j} = r \times (U_j - L_j) + L_j \tag{2}$$

Where S stands for the solution population matrix, $s_{i,j}$ is the value of the j^{th} variable for the i^{th} individual, N is the number of individuals (population size), D denotes the number of dimensions (decision variables), L_j and U_j are lower and upper bounds for the j^{th} variable, and r is a random number drawn from a uniform distribution in [0, 1].

The best solution in the population is selected as the alpha mongoose, which acts as a leader. This is determined using the fitness function F, as shown in Eq. 3.

$$\alpha = min\left(\mathcal{F}(s_1), \mathcal{F}(s_2), \dots, \mathcal{F}(s_N)\right) \tag{3}$$

Where α is the current best-performing individual (alpha) and $\mathcal{F}(s_i)$ refers to the fitness value (objective function) of the i^{th} solution.

Each mongoose then updates its position relative to α as specified in Eq. 4, based on an adaptive coefficient ω calculated using Eq. 5.

$$\begin{gathered} s_i^{(t+1)} = \alpha + \phi \cdot r \cdot \left(s_i^{(t)} - s_k^{(t)}\right) \\ \phi = \frac{\gamma}{2} \cdot r \cdot \omega \end{gathered} \tag{4}$$

$$\omega = exp\left(-4 \cdot \left(\frac{t}{T}\right)^2\right) \tag{5}$$

Where $s_i^{(t)}$ is the position of the i^{th} mongoose at iteration t, $s_k^{(t)}$ is a randomly selected mongoose, γ is a vector of zeros of length D used to initialize influence, ω is the adaptation coefficient controlling exploration/exploitation, ϕ is the adaptive influence factor, t is the current iteration number, and T is the maximum number of iterations.

Scouts are individuals responsible for exploring new regions in the search space. Their behavior is designed to promote diversity by updating positions based on the relative difference between two randomly selected individuals, as shown in Eq. 6.

$$s_i^{(t+1)} = \alpha + \phi \cdot r \cdot \left(\frac{s_k - s_h}{2}\right) \tag{6}$$

Where s_k *and* s_k are two randomly selected mongooses from the population.

Babysitters are periodically evaluated and replaced based on a dynamic threshold Λ. Once the condition is met, babysitters are updated using information from randomly selected individuals, improving their quality. The threshold is determined using Eq. 8 and Eq. 9, and replacement is performed as in Eq. 10.

$$\Lambda = \begin{cases} \left\lceil 0.6 \cdot N \cdot D \cdot \dfrac{1}{t} \right\rceil, & if\ uninitialized \\ \Lambda \cdot t \cdot \xi, & if\ \Lambda < 0 \end{cases} \tag{7}$$

$$\xi = \left(1 - \frac{t}{T}\right)^2 \cdot \frac{t}{T} \tag{8}$$

$$s_i^{(t+1)} = s_j + r \cdot \left(\alpha - \frac{s_k - s_h}{2} \cdot \beta\right) \tag{9}$$

Where Λ is the babysitter replacement threshold, ξ refers to the dynamic coefficient for controlling the decay of Λ, and β is the birth rate coefficient controlling babysitter influence.

The advantage of MDMO lies in its structured use of stochastic operators and role-based agents. Every phase of the algorithm, from initial generation, elite-guided updates, randomized scout movements, to adaptive babysitter replacement, uses diversity-preserving strategies to enhance convergence and robustness. The dynamic coefficients ω, ϕ, and ξ ensure that the algorithm can adapt its behavior over time, balancing global exploration and local exploitation. MDMO is well-suited for high-dimensional, nonlinear optimization problems such as feature selection in student performance prediction. Figure 2 presents the flowchart of the proposed algorithm.

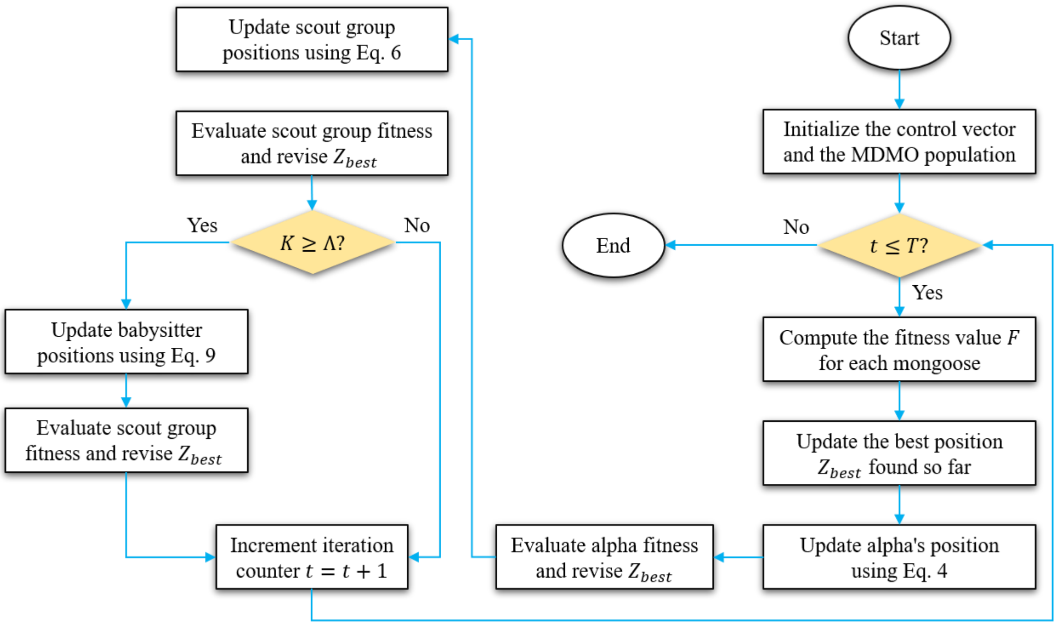

Start
Initialize the control vector and the MDMO population
$t \le T?$
No
End
Yes
Compute the fitness value F for each mongoose
Update the best position Z_{best} found so far
Update alpha's position using Eq. 4
Evaluate alpha fitness and revise Z_{best}
Update scout group positions using Eq. 6
Evaluate scout group fitness and revise Z_{best}
$K \ge \Lambda?$
Yes
No
Update babysitter positions using Eq. 9
Evaluate scout group fitness and revise Z_{best}
Increment iteration counter $t = t + 1$

Figure 2: Flowchart of proposed algorithm

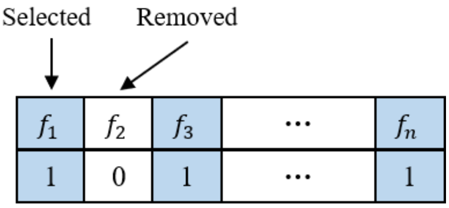


Figure 3: Binary vector representation of a feature subset used in MDMO

3.2 Binary transformation for feature selection

To tailor the MDMO algorithm for feature selection tasks, a binary representation scheme is adopted to encode potential solutions. In this context, each mongoose in the population represents a candidate feature subset, encoded as a binary vector $Z = [z_1, z_2, \dots, z_d]$, where d is the total number of original features in the dataset. Each element $z_i \in \{0,1\}$ denotes whether the i^{th} feature is included ($z_i = 1$) or excluded ($z_i = 0$) from the selected subset. This binary format enables MDMO to perform subset optimization efficiently by navigating the discrete feature space, as illustrated in Figure 3.

Incorporating binary transformation into MDMO requires a fitness function balancing two competing objectives: achieving high classification accuracy and selecting the fewest features. A scalar multi-objective fitness function captures this trade-off, which guides the movement and role-based behavior of each mongoose in the search space. The objective function used to evaluate each binary solution is defined using Eq. 10.

$$Fitness(z) = \lambda \cdot \mathcal{E}_{val} + (1 - \lambda) \cdot \frac{|F_{active}|}{|F_{total}|} \qquad (10)$$

Where $Fitness(z)$ stands for the cost assigned to the solution Z, $\mathcal{E}_{val}$ refers to the classification error rate obtained using a validation classifier trained on selected features, F_{active} denotes the number of features selected, F_{total} is the total number of available features in the original dataset, and $\lambda \in [0,1]$ is a weighting parameter determining the relative importance of classification performance versus feature reduction.

3.3 Handling imbalanced data

A common issue in machine learning classification is the presence of class imbalance, where one or more classes have significantly fewer samples than others. This challenge is especially prevalent in real-world datasets, where the distribution of target labels is often skewed [20]. In binary classification, minority classes typically consist of rare instances, while the majority class dominates the dataset. Classifiers trained under such imbalanced conditions tend to be biased toward the majority class, which results in poor performance when predicting the underrepresented (minority) class.

To address this, adaptive oversampling techniques have been developed. One of the most effective is the Adaptive Synthetic Sampling Method (ASSM), which builds upon the well-established SMOTE algorithm. ASSM enhances learning from imbalanced datasets by dynamically generating synthetic samples for the minority class based on their distributional difficulty.

Unlike uniform oversampling, ASSM emphasizes generating more synthetic instances for minority samples that are harder to classify, while generating fewer for easier ones. This data-driven adaptability allows the classifier to reduce its bias toward the majority class and better define the decision boundary around difficult regions. By doing so, ASSM improves the model's generalization ability on challenging, imbalanced data.

4 Results

The research uses two public datasets to build and test student performance prediction models. The first dataset originated from Gazi University (Turkey), and the second was obtained from secondary-level educational institutions in Portugal. Dataset 1 has 32 attributes, 28 course-specific response questions, and four others. The target variable indicates the frequency of taking one course. To convert the target into a binary classification problem, it is recoded where students who have attempted the course more than once are classified as class 1, and students with zero or one attempt are classified as class 0. During data preprocessing, all values of the features were normalized in the range [0,1] to make the data consistent and reduce scale-induced distortions.

All experiments were implemented in MATLAB and WEKA. For MDMO, the population was set to 30 and the maximum iterations to 100, with adaptive α dynamics and babysitter exchange probability calibrated through pilot testing. Classifier hyperparameters followed WEKA's default configurations unless otherwise specified (e.g., k=5 for KNN). We adopted a 10-fold cross-validation scheme to ensure reliable evaluation, repeated across 30 independent runs. Each classifier was trained and tested on the original complete feature set and on subsets selected by MDMO under identical folds, guaranteeing consistency. We applied the Wilcoxon signed-rank test at the 0.05 level to evaluate statistical robustness and compare MDMO with alternative algorithms.

Dataset 2 was gathered from high school students in Portugal and contains 33 input variables. These inputs are demographic features, academic performance, and socially related information, obtained through structured questionnaires and educational records. The dataset includes two subject fields: the Portuguese language and mathematics (mat).

The target variable of primary interest is the last grade (G3), which is transformed into a binary classification space in this research: students who scored G3 < 10 were assigned to class 1 (indicating risk), and students who scored G3 ≥ 10 were assigned to class 0 (non-risk). All features were normalized to [0,1] during preprocessing. The dataset for training was the Portuguese language dataset, and the dataset for testing was the mathematics dataset.

As illustrated in Table 2, both datasets suffer from significant class imbalances. In Dataset 1, just 0.156% of

Table 2: Overview of datasets used for student performance analysis

Dataset	Feature count	Record count	Target attribute	Binary encoding	Underrepresented class	Class imbalance (%)
Data1	32	5,820	Course repetition	0: ≤1 attempt, 1: >1 attempt	Repeated >1 time	15.6%
Data2	32	1,044	Final grade (G3)	0: pass (≥10), 1: fail (<10)	Fail (G3 < 10)	22.0%

the records belong to students who had to retake the course (class 1), and hence it is highly imbalanced. In Dataset 2, the minority group (G3 < 10) is underpopulated compared to the majority class (G3 ≥ 10). For such an imbalance, balanced data is vital to avoid biased learning and enhance model generalizability.

Extensive experiments were performed to thoroughly verify the performance of the suggested MDMO. These experiments aimed to test the model's performance in solving the problem of student performance prediction in real-world scenarios like imbalanced data and high-dimensional feature spaces. In binary classification issues, the accuracy of a model's prediction is usually assessed using various performance measures calculated from the confusion matrix, which consists of the following terms:

- True Positive (TP): Correctly predicted positive instances
- True Negative (TN): Correctly predicted negative instances
- False Positive (FP): Incorrectly predicted positive instances
- False Negative (FN): Incorrectly predicted negative instances

Accuracy measures the overall correctness of the model by calculating the proportion of correctly classified instances (both positives and negatives) over the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (11)$$

Sensitivity (Recall or TPR) evaluates the model's ability to correctly identify positive instances.

$$Sensitivity = \frac{TP}{TP + FN} \qquad (12)$$

Specificity measures how well the model identifies actual negative instances, i.e., its ability to avoid false alarms.

$$Specificity = \frac{TN}{TN + FP} \qquad (13)$$

The Area Under the Curve (AUC) score is derived from the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) across various classification thresholds.

$$AUC = \frac{FP}{FP + TN} \qquad (14)$$

Unlike standard metrics like accuracy and F1-score, the AUC metric is threshold-independent, making it especially suitable for evaluating binary classification models where class imbalance is present. This insensitivity to decision threshold variations allows AUC to generalize the classifier's performance more robustly, as supported by the values presented in Table 3.

Table 3: Confusion matrix structure for binary classification

	Predicted: Positive	Predicted: Negative
Actual: Positive	TP	FN
Actual: Negative	FP	TN

The five well-known classifiers, k-NN, Decision Tree (DT), Linear Discriminant Analysis (LDA), NB, and LogitBoost (LB), were initially compared in an experimental study to identify the appropriate base learner to be used with MDMO. The analysis was performed in two stages: firstly, on raw datasets without preprocessing, and secondly, using resampling with differing balancing ratios. The findings are presented in Table 4 (without resampling and feature selection) and Table 5 (with resampling, but not feature selection).

As shown in Table 4, LB performed poorly on Dataset 1 but had the highest AUC value on Dataset 2. In contrast, using synthetic oversampling in Table 5, we found that k-NN had an AUC of 0.862 on Dataset 2 after using a 0.4 resampling ratio, and LDA had an AUC of 0.635 on Dataset 1 based on the full balancing ratio of 1.0. Thus, because of its improved and consistent performance in all cases, LDA was chosen as the default classifier for the assessment of MDMO.

To systematically evaluate MDMO, we compared its performance with several high-performing metaheuristic binary optimization algorithms such as GA, Binary Ant Lion Optimizer (BALO), Binary Bat Algorithm (BBA), Binary Grey Wolf Optimizer (BGWO), Binary Particle Swarm Optimization (BPSO), Binary Grasshopper Optimization Algorithm (BGOA), Binary Gravitational Search Algorithm (BGSA), and Binary Harris Hawks Optimization (BHHO). The comparison performance in terms of mean AUC, subset size, and statistical measures is shown in Table 6. The findings indicate that MDMO has

Table 4: Performance comparison of classification algorithms without feature selection and resampling

Datasets	Algorithms	Accuracy	AUC	Specificity	Sensitivity
Data1	NB	0.836	0.518	0.981	0.061
	LB	0.838	0.599	0.948	0.249
	LDA	0.843	0.512	0.992	0.029
	DT	0.807	0.593	0.903	0.283
	k-NN	0.825	0.586	0.931	0.237
Data2	NB	0.871	0.728	0.469	0.982
	LB	0.904	0.845	0.745	0.945
	LDA	0.901	0.824	0.683	0.961
	DT	0.888	0.831	0.728	0.933
	k-NN	0.901	0.825	0.681	0.964

Table 5: AUC performance of classifiers across varying oversampling levels (excluding feature selection)

Datasets	Algorithms	No oversampling (0)	Ratio = 0.2	Ratio = 0.4	Ratio = 0.7	Full oversampling (1.0)
Data1	NB	0.516	0.521	0.533	0.572	0.567
	LB	0.597	0.605	0.612	0.614	0.613
	LDA	0.511	0.536	0.594	0.632	0.635
	DT	0.592	0.597	0.601	0.604	0.604
	k-NN	0.584	0.623	0.632	0.631	0.632
Data2	NB	0.725	0.862	0.851	0.801	0.774
	LB	0.847	0.845	0.846	0.848	0.851
	LDA	0.824	0.866	0.873	0.883	0.878
	DT	0.831	0.837	0.838	0.837	0.840
	k-NN	0.823	0.851	0.862	0.850	0.852
Average rank (F-Test)		4.8	3.5	2.8	2.2	1.6

Table 6: Comparative evaluation of MDMO and benchmark optimizers

Datasets	Metrics		BALO	GA	BBA	BGWO	BPSO	BGOA	BGSA	BHHO	MDMO
Data1	AUC	AVG	0.637	0.639	0.618	0.638	0.641	0.635	0.636	0.638	0.658
		STD	0.004	0.002	0.021	0.002	0.003	0.005	0.004	0.003	0.0005
	Features	AVG	26.3	5	18	11.9	15.8	18.6	16.7	19.1	3
		STD	3.221	1.631	1.324	2.751	2.215	2.593	1.924	3.571	0.000
	Fitness	AVG	0.357	0.359	0.357	0.351	0.353	0.355	0.357	0.356	0.338
		STD	0.0009	0.0011	0.0034	0.0012	0.0011	0.0012	0.0021	0.0012	0.0005
	AUC	AVG	0.891	0.898	0.864	0.904	0.902	0.896	0.891	0.897	0.918
		STD	0.004	0.007	0.067	0.004	0.006	0.009	0.007	0.006	0.001
	Features	AVG	22.4	10.7	14.5	5.6	10.7	13.3	14.7	13.2	1.6
		STD	5.246	2.810	2.632	1.851	2.281	2.528	2.224	2.514	1.221
	Fitness	AVG	0.102	0.095	0.098	0.085	0.092	0.093	0.098	0.095	0.083
		STD	0.0013	0.0027	0.0019	0.0021	0.0017	0.0025	0.0029	0.0023	0.0011

Table 7: G-mean comparison between the proposed method and various baseline models

Models	Data1	Data2
MDMO	0.778	0.921
Imbalanced RVFL Opt1	0.715	0.719
Imbalanced RVFL Opt2	0.712	0.721
Enhanced RVFL (MCC)	0.707	0.725
Enhanced RVFL (KDE)	0.706	0.724
Baseline RVFL	0.688	0.711
Opt1 from Method 1	0.725	0.749
Opt2 from Method 1	0.727	0.748
Opt1 from Method 2	0.726	0.748
Opt2 from Method 2	0.725	0.747

Table 8: Comparison of G-mean values between MDMO and baseline models

Datasets	BTLBO-LDA	MLP-Adam	MDMO
Data1	0.632	0.605	0.672
Data2	-	0.820	0.929

the highest ranking and surpasses all other algorithms in terms of classification performance and subset efficiency.

To benchmark the generalizability of MDMO, we compared its G-mean results with those from state-of-the-art methods [21]. As shown in Table 7, MDMO outperformed the existing methods across both datasets. Moreover, when comparing AUC values reported in previous studies [22, 23], the results in Table 8 indicate that MDMO surpassed all previously published benchmarks on the same datasets.

To assess statistical significance, a Wilcoxon signed-rank test was conducted between MDMO and all benchmark optimizers (Table 9). The results show that MDMO's improvements are statistically significant ($p < 0.05$) across both datasets in nearly all comparisons, confirming that the observed performance gains are not due to chance.

5 Discussion

The experimental results confirm that the proposed MDMO consistently outperforms existing metaheuristic-based student performance prediction models. For example, on Data1, MDMO achieved an AUC of 0.672, exceeding values reported for BTLBO-LDA and MLP-Adam. On Data2, MDMO reached an AUC of 0.929, substantially higher than the best-performing competitor. These results highlight MDMO's ability to maintain robust predictive accuracy while selecting compact feature subsets.

Compared with prior works summarized in Table 1, MDMO addresses two recurring limitations: (i) inadequate handling of imbalanced data and (ii) lack of explicit feature selection mechanisms. Approaches such as HHO or NB-based classifiers demonstrated strong accuracy on balanced or domain-specific datasets but did not generalize to imbalanced scenarios. In contrast, MDMO explicitly integrates imbalance handling (via ADASYN) with a tailored search strategy, enabling higher generalization. The superior performance of MDMO can be attributed to three design choices:

- Adaptive alpha control dynamically balances exploration and exploitation, preventing premature convergence.
- Role-based babysitter exchange maintains population diversity and reduces stagnation.
- Scout diversification ensures broad coverage of the search space and avoids local minima.

Table 9: Wilcoxon signed-rank test (p-values) comparing MDMO with benchmark optimizers

Dataset	MDMO vs	p-value	Significance
Data1	BALO	0.021	✓ (p < 0.05)
	GA	0.018	✓ (p < 0.05)
	BBA	0.009	✓ (p < 0.01)
	BGWO	0.027	✓ (p < 0.05)
	BPSO	0.015	✓ (p < 0.05)
	BGOA	0.033	✓ (p < 0.05)
	BGSA	0.011	✓ (p < 0.05)
	BHHO	0.024	✓ (p < 0.05)
Data2	BALO	0.007	✓ (p < 0.01)
	GA	0.010	✓ (p < 0.01)
	BBA	0.004	✓ (p < 0.01)
	BGWO	0.013	✓ (p < 0.05)
	BPSO	0.008	✓ (p < 0.01)
	BGOA	0.016	✓ (p < 0.05)
	BGSA	0.009	✓ (p < 0.01)
	BHHO	0.012	✓ (p < 0.05)

6 Conclusion

In the present work, we introduced an improved metaheuristic solution, MDMO, that solves the difficult task of predicting student performance using optimal feature selection. The MDMO algorithm adds three key improvements to the standard Dwarf Mongoose Optimization: adaptive alpha guidance via dynamic movement control to control the exploration and exploitation phases of the algorithm, scout-guided randomized exploration for avoiding premature convergence, and an efficient babysitter exchange mechanism for improved sharing of information and diversity. These improvements were tailored to enhance exploration–exploitation trade-offs and solution quality in high-dimensional binary problems. The adaptive synthetic oversampling algorithm was utilized as a preprocessing to overcome the natural imbalance in classes in educational datasets and prepare them for training the classifiers. A binary conversion mechanism was employed to express candidate subsets of features, and a multi-objective optimization function was established to reduce classification error and the number of features.

Extensive experimentation was performed on real-world datasets with five classifiers, and LDA was chosen as the base model following a performance comparison. Experimentation on real-world datasets using five different classifiers established the dominance of MDMO in outperforming various state-of-the-art metaheuristic solutions, including BHHO, BGSA, BPSO, BGOA, and BBAs, in terms of AUC, number of features in selected subsets, and overall classification accuracy. Statistical verification through the Wilcoxon signed-rank test further validated MDMO's superiority in generating accurate and compact models. Convergence analysis also revealed faster and more consistent MDMO behavior than other versions.

References

[1] S. M. Dol and P. M. Jawandhiya, "Systematic review and analysis of EDM for predicting the academic performance of students," *Journal of The Institution of Engineers (India): Series B,* vol. 105, no. 4, pp. 1021–1071, 2024, doi: https://doi.org/10.1007/s40031-024-00998-0.

[2] M. Shoaib, N. Sayed, J. Singh, J. Shafi, S. Khan, and F. Ali, "AI student success predictor: Enhancing personalized learning in campus management systems," *Computers in Human Behavior,* vol. 158, p. 108301, 2024, doi: https://doi.org/10.1016/j.chb.2024.108301.

[3] G. Feng and H. Chen, "Educational process mining: A study using a public educational data set from a machine learning repository," *Education and Information Technologies,* vol. 30, no. 6, pp. 8187–8214, 2025, doi: https://doi.org/10.1007/s10639-024-13130-y.

[4] J. Li, "AI-Driven Property Management Decision Support System Using LSTM Networks for Energy Optimization," *Informatica,* vol. 49, no. 10, 2025, doi: https://doi.org/10.31449/inf.v49i10.6964.

[5] A. Bolívar, V. García, R. Alejo, R. Florencia-Juárez, and J. S. Sánchez, "Data-centric solutions for addressing big data veracity with class imbalance, high dimensionality, and class overlapping," *Applied Sciences,* vol. 14, no. 13, p. 5845, 2024, doi: https://doi.org/10.3390/app14135845.

[6] H. Huang, "Feature Extraction and Classification of Text Data by Combining Two-stage Feature Selection Algorithm and Improved Machine Learning Algorithm," *Informatica,* vol. 48, no. 8, 2024, doi: https://doi.org/10.31449/inf.v48i8.5763.

[7] A. Alaff and Ç. Uluyol, "Integrating Equation-Based Labeling and Classification for Adaptive Turkish Vocabulary Acquisition," *Informatica,* vol. 49, no. 27, 2025, doi: https://doi.org/10.31449/inf.v49i27.8821.

[8] F. Han, Y.-H. Wang, and F.-Y. Li, "A novel feature selection method based on adaptive search particle swarm optimization," *Neural Computing and Applications,* vol. 37, no. 12, pp. 7767–7783, 2025, doi: https://doi.org/10.1007/s00521-024-10611-6.

[9] Y. Xue, H. Zhu, J. Liang, and A. Słowik, "Adaptive crossover operator based multi-objective binary genetic algorithm for feature

selection in classification," *Knowledge-Based Systems,* vol. 227, p. 107218, 2021, doi: https://doi.org/10.1016/j.knosys.2021.107218.

[10] W. Ma, X. Zhou, H. Zhu, L. Li, and L. Jiao, "A two-stage hybrid ant colony optimization for high-dimensional feature selection," *Pattern Recognition,* vol. 116, p. 107933, 2021, doi: https://doi.org/10.1016/j.patcog.2021.107933.

[11] M. H. Nadimi-Shahraki, H. Zamani, and S. Mirjalili, "Enhanced whale optimization algorithm for medical feature selection: A COVID-19 case study," *Computers in biology and medicine,* vol. 148, p. 105858, 2022, doi: https://doi.org/10.1016/j.compbiomed.2022.105858.

[12] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf mongoose optimization algorithm," *Computer methods in applied mechanics and engineering,* vol. 391, p. 114570, 2022, doi: https://doi.org/10.1016/j.cma.2022.114570.

[13] H. Turabieh *et al.*, "Enhanced Harris Hawks optimization as a feature selection for the prediction of student performance," *Computing,* vol. 103, no. 7, pp. 1417–1438, 2021, doi: https://doi.org/10.1007/s00607-020-00894-7.

[14] M. Kamal *et al.*, "Metaheuristics method for classification and prediction of student performance using machine learning predictors," *Mathematical Problems in Engineering,* vol. 2022, no. 1, p. 2581951, 2022, doi: https://doi.org/10.1155/2022/2581951.

[15] M. R. Apriyadi and D. P. Rini, "Hyperparameter optimization of support vector regression algorithm using metaheuristic algorithm for student performance prediction," *International Journal of Advanced Computer Science and Applications,* vol. 14, no. 2, 2023, doi: https://doi.org/10.14569/IJACSA.2023.0140218.

[16] X. Song, "Student performance prediction employing k-nearest neighbor classification model and meta-heuristic algorithms," *Multiscale and Multidisciplinary Modeling, Experiments and Design,* vol. 7, no. 4, pp. 4397–4412, 2024, doi: https://doi.org/10.1007/s41939-024-00481-9.

[17] C. Ma, "Improving the Prediction of Student Performance by Integrating a Random Forest Classifier with Meta-Heuristic Optimization Algorithms," *International Journal of Advanced Computer Science & Applications,* vol. 15, no. 6, 2024, doi: https://doi.org/10.14569/ijacsa.2024.01506106.

[18] H. Shou and Y. Lu, "Student Performance Evaluation Technique By Applying Support Vector Classification And Metaheuristic Algorithms On The SVC Model's Reliability," *Journal of Applied Science and Engineering,* vol. 28, no. 3, pp. 653–666, 2025, doi: http://dx.doi.org/10.6180/jase.202503_28(3).0020.

[19] Z. Ye, Y. Yang, Y. Chen, and H. Chen, "Predicting Academic Performance Levels in Higher Education: A Data-Driven Enhanced Fruit Fly Optimizer Kernel Extreme Learning Machine Model," *Journal of Bionic Engineering,* pp. 1–23, 2025, doi: https://doi.org/10.1007/s42235-025-00716-6.

[20] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," *Authorea Preprints,* 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[21] M. Li, C. Huang, D. Wang, Q. Hu, J. Zhu, and Y. Tang, "Improved randomized learning algorithms for imbalanced and noisy educational data classification," *Computing,* vol. 101, pp. 571–585, 2019, doi: https://doi.org/10.1007/s00607-018-00698-w.

[22] S. Alraddadi, S. Alseady, and S. Almotiri, "Prediction of students academic performance utilizing hybrid teaching-learning based feature selection and machine learning models," in *2021 International Conference of Women in Data Science at Taif University (WiDSTaif)*, 2021: IEEE, pp. 1–6, doi: https://doi.org/10.1109/WiDSTaif52235.2021.9430248.

[23] T. Thaher and R. Jayousi, "Prediction of student's academic performance using feedforward neural network augmented with stochastic trainers," in *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, 2020: IEEE, pp. 1–7, doi: https://doi.org/10.1109/AICT50176.2020.9368820.

A Stackelberg Game-Theoretic and Mixed Integer Programming Framework for Collaborative Optimization in Multi-Energy Transportation Systems

Yanmei Ren[1,2]
[1]Puyang Institute of Technology, Henan University, Puyang, 457000, China
[2]Puyang Vocational and Technical College, Puyang, 457000, China
E-mail: yanmeirenn@outlook.com

Keywords: Multi-energy transportation system, collaborative optimization, game theory, mixed integer programming, Stackelberg game

Received: August 5, 2025

In the context of the deep integration of energy transformation and transportation electrification, multi-energy transportation systems involve electricity, hydrogen energy, natural gas and infrastructures like charging stations, hydrogen refueling stations, with coordinated operation facing challenges from conflicting interests and complex physical constraints. Traditional optimization models often overlook game behaviors among energy suppliers, operators and users, leading to poor executability of scheduling plans. Thus, this study proposes a two-level collaborative optimization framework integrating game theory and MIP: the upper level takes charging/hydrogen station operators as leaders (maximizing daily net revenue via pricing, subject to pricing range and station capacity constraints); the lower level takes EV (31,200 daily trips) and FCEV (8,600 daily trips) users as followers (minimizing total travel costs via station and energy demand selection). To solve the bi-level game, the framework transforms followers' optimal responses into mathematical constraints via KKT conditions, introduces binary variables and Big-M method to linearize complementary relaxation conditions, and finally forms an MIP model with continuous and integer variables. It uses Gurobi 10.0, with a simulation environment built on MATLAB R2023a and SUMO 1.18.0.Simulation results based on a regional energy internet case (covering 20 charging stations, 10 hydrogen stations, 15 transportation hubs, 24-hour scheduling) show multi-dimensional improvements: vs. single-level centralized optimization, total operating costs down 15.7%, renewable energy utilization up 22.3%; vs. disordered scheduling, user waiting time reduced 31.5%, operators' revenue up 12.9%; vs. RL models (DQN, PPO) in 50-node systems, optimization time down 57%, total costs further reduced 18.3%. Verified by 1,000 Monte Carlo simulations, the model has a total operational cost fluctuation coefficient of 3.2%, 95.3% constraint satisfaction rate in 100-node dynamic scenarios, and Nash equilibria with fluctuations <5% in 98% of nodes, fully validating its effectiveness and stability in coordinating economy, environmental protection and user experience.

Povzetek: Študija predlaga dvo-ravenski sodelovalni pristop za promet, ki s teorijo iger in MIP modelira *voditelje ter sledilce. Odzive sledilcev vključi prek KKT v MIP in tako sočasno optimizira cene, razporejanje ter poti ob fizičnih omejitvah.*

1 Introduction

In the era background where the low-carbon transformation of the global energy structure and the electrification process in the Transportation field are deeply integrated, Multi-Energy Transportation Systems (MTS), as the key hub connecting the energy network and the transportation network, are becoming increasingly important [1]. Such systems involve the coupling and conversion of various heterogeneous energy forms such as electricity, hydrogen energy, and natural gas, and achieve the deep coordination of energy flow and traffic flow through complex infrastructure such as charging stations, hydrogen refueling stations, and gas-electricity conversion devices [2, 3]. However, there are multiple structural challenges within MTS: On the one hand, there are complex interaction constraints at the physical level between the energy network and the transportation network, including nonlinear factors such as energy transmission power limitations, dynamic capacity constraints of charging/hydrogen injection facilities, and traffic balance in the transportation network; On the other hand, the interest demands of each participating entity in the system have significant conflicts in the environment of information asymmetry, presenting the characteristics of non-cooperative games [4]. The traditional single-objective optimization model is difficult to fully capture the inherent complexity of such physical-social coupled systems - especially ignoring the impact of strategic interactions among decision-making subjects on the overall operational efficiency of the system. The scheduling schemes obtained solely through centralized planning or single-subject optimization methods often fail to effectively coordinate individual rationality and collective optimality. This leads to a significant reduction in practical implement ability or a situation where both economic efficiency and system reliability are compromised [5]. Although existing studies have formed relatively mature modeling paradigms in the field of independently optimizing energy systems or transportation networks, such as the precise solution ability demonstrated by Mixed integer Programming (MIP) in hard constraint problems like infrastructure site selection and unit combination, it has inherent limitations in characterizes multi-agent interactive decision-making behaviors [6, 7]. Meanwhile, although the analysis of multi-agent interaction solely using classical game theory can reveal strategy equilibrium, it is often difficult to effectively handle the embedding of physical constraints in large-scale systems and the numerical feasibility of solving them. How to organically integrate the two remains a bottleneck that needs to be urgently broken through in current research [8].

This impasse has prompted a shift in research focus towards constructing a new collaborative optimization framework capable of simultaneously embedding accurate modeling of physical systems and multi-agent policy behavior analysis. The core innovation of this research lies in proposing and implementing a two-level interactive modeling mechanism that combines non-cooperative game theory and mixed integer programming, aiming to overcome the fragmentation of traditional methods in these dimensions. Specifically, a leader-follower hierarchical decision-making structure is established based on the Stackelberg game paradigm, where charging station/hydrogen refueling station operators act as leaders in formulating service pricing strategies, while electric vehicle/hydrogen fuel vehicle users act as followers, dynamically adjusting their energy consumption/path selection behavior based on real-time price signals and road condition information. Through this mechanism, the conflict and coordination dynamics between the operator's profit maximization goal and the user experience cost goals (such as waiting time, traveling distance) can be formally described [9, 10]. More importantly, this study overcomes the computational barriers of game equilibrium modeling, employing the Karush-Kuhn-Tucker (KKT) conditions to equivalently transform the follower's optimal response problem into a series of mixed integer linear constraints. This transformation then reconstitutes the bilevel game equilibrium solving problem into a single-level, processable mixed integer programming model. This approach not only preserves a high-fidelity representation of hard constraints such as the physical topology structure, energy flow equations, and equipment operation boundaries of the electricity-transportation-hydrogen energy coupled network but also fully incorporates the mutual feedback effect of dynamic decision-making behavior among policy agents [11-13]. It provides system operators with collaborative scheduling decision support that combines physical feasibility with economic incentive compatibility [14, 15].

Finally, through the establishment and solution of this framework, the aim is to provide a unified analysis tool and optimization foundation for multi-level decision-making, such as infrastructure investment, market pricing strategies, and user demand guidance in complex multi-energy transportation systems. This is intended to promote the system's energy efficiency, economic benefits, and a notable improvement in environmental sustainability dimensions.

The research focuses on two core issues: first, addressing the pricing game and conflicts in the choice of different energy paths among supply, transportation, and demand in multi-energy coupling transport, clarifying the decision-making logic and collaborative mechanisms of the main entities; second, overcoming the bottleneck where the equilibrium solution of the game is difficult to convert into a solvable Mixed Integer Programming (MIP) model, designing conversion methods and constraint strategies. The research goal is to construct a multi-agent game collaborative optimization framework, propose pricing-path collaborative methods, establish an efficient MIP model, and validate its feasibility. The research contributions include: revealing the mechanism of pricing and path selection games, proposing a new method for the conversion of game equilibria to MIP, and providing theoretical and tool support for system operation management.

2 Theoretical basis and principle technology

2.1 Fundamentals of game theory

Game theory is an important branch of mathematics and operations research that studies how multiple actors with shared interests make favorable decisions. It is widely applied in fields such as economics and sociology, and serves as a core analytical tool in economics. In the study of non-cooperative games, the core is the decision-making interactions among independent participants who are equal in status and share information. The optimization goals and outcomes of the participants are influenced by the interaction of each other's strategies, which may lead to conflicts, while Nash equilibrium plays a key role in this field.

Nash equilibrium is crucial in this field. For a game G={N;{S_i}i∈N;{U_i}i∈N} with N players, when the strategy combination ($s*_i$,$s*_{-i}$) satisfies the condition (1), it is called a pure strategy Nash equilibrium.

$$u_i\left(s_i^*,s_{-i}^*\right)..u_i\left(s_i,s_{-i}^*\right),\forall s_i\in S_i,\forall i\in N \quad (1)$$

In the equilibrium state, the strategy of the first participant is denoted as s^*_i,s^*_{-i}=(s^*_1,…s^*_{i-1},s^*_{i+1},…,s^*_N), which represents the strategies of all participants except the i-th participant in the equilibrium state. ui represents the utility function of the i-th participant. Nash equilibrium strategy means that in this state, no participant has the motivation to unilaterally change the strategy to improve its own utility, while not affecting other participants [16, 17]. Therefore, the Nash equilibrium constitutes a stable game outcome. Before finding Nash equilibrium, the existence of Nash equilibrium must be analyzed and verified. For pure strategy games, the following theorems need to be satisfied: if the game G={N;{S_i}i∈N;{U_i}i∈N} holds for all participants i∈N: (1) the strategy set Sı is a non-empty compact convex set in Euclidean space, (2) the utility function u; As for the strategy s, it is a continuous quasi-concave function, then there is a pure strategy Nash equilibrium in the game.

Master-slave non-cooperative games are different from peer-to-peer games in that there is a hierarchical difference between leaders and followers [18]. Leaders are at the upper level and have decision-making advantages, while followers are at the lower level and need to respond according to leaders' decisions [19, 20]. In a Stackelberg game G containing a leader and N followers, the strategy combination ($s^*{}_l$,$s^*{}_f$) is a Stackelberg equilibrium if and only if formula (2) holds.

$$u_l\left(s_l^*,s_f^*\right)..u_l\left(s_l,s_f^*\right),\forall s_l\in S_l,$$

$$u_{f,i}\left(s_{f,i}^*,s_{f,i}^*\right)..u_{f,i}\left(s_{f,i},s_{f,i}^*\right),\forall s_{f,i}\in S_{f,i},\forall i\in N,$$

(2)

In the Stackelberg game, the utility function of the leader is u1, and that of the followers is uf, i. The strategy of the leader is s1 and the strategy of the followers is sf, i. The strategy set of the leader is Sl, and the strategy set of the followers is Sf, i. The total number of followers is N. By definition, when in Stackelberg equilibrium, participants cannot increase returns by changing their own strategies without affecting others [21].

The alliance structure in cooperative game is studied, and each alliance can be regarded as an independent cooperative game model, which is represented by SC={S1,S₂,…,Sk} and satisfies specific conditions, as shown in equation (3).

$$\bigcup_{i=1}^{K}S_i=N,S_i\bigcap S_j=\varnothing,\forall i\neq j\ (3)$$

The key to building a solid alliance is to formulate a fair and reasonable distribution plan. Common schemes include the Shapely value method, the nucleolar method, and the Disruption Propensity (DP) indicator.

When dealing with multi-agent interaction optimization problems, the Nash bargaining game method is often used to find consensus solutions [22, 23]. This approach aims to achieve the overall optimization of the system while ensuring that every rational participant gets a just return [24]. Nash bargaining solution should follow the four core principles of Pareto optimality, symmetry, linear transformation invariance and independence of irrelevant choice [25]. The Nash bargaining game involves the set of utility functions U={u1, … un} of N participants and the set of negotiation breakdown points B={b₁, … bn}, that is, the benefits of all parties when they do not cooperate. Participants avoid reaching the breaking point in negotiation to maximize their own interests, and the resolution process of Nash bargaining game is an evolution from the breaking point to the fair negotiation point [26]. The standard Nash bargaining problem can be expressed as formula (4).

$$\max_{s\in S}\prod_{i=1}^{N}\left(u_n\left(s\right)-b_n\right)\ (4)$$
$$s.t.u_n..b_n,$$

In this framework, s represents the strategy choice of the participants and S represents the strategy set. Researchers have proposed an asymmetric Nash bargaining game model to ensure fairness [27]. Based on this model, scholars have deeply studied the bargaining factors, aiming at fairly distributing the cooperation benefits to all participants.

2.2 Fundamentals of mixed integer programming

Mixed integer programming is crucial in operations research, combining integer and non-integer variables, and is more practical but more complex than pure integer programming in practical applications [28]. It is commonly used for linear and nonlinear problems and is solved by transforming the problem into a discrete model. Although it performs well in finding local optimal solutions, the solution of global optimal solutions is more challenging. To this end, researchers have developed accurate algorithms and heuristic algorithms. Exact algorithms, such as branch-bound method and column generation method, are suitable for cases where the number of variables is limited; Heuristic algorithms, including particle swarm, fruit fly, genetic and ant colony algorithms, can effectively handle a large number of variables. Bionic algorithms have attracted attention in algorithm research because of their advantages, providing efficient solutions [29]. Optimal algorithm and weighting algorithm are also widely used in this field.

When solving mixed integer linear programming problems, the branch-and-bound method is usually used [30]. This method can quickly find the best solution. In the process of solving, a solution interval is formed, and the maximum and minimum values of the interval represent the effective upper and lower limits respectively. The difference between the upper bound and the lower bound of the solution is measured by the Gap value, which reflects the quality of the solution. If $A^* = A_0$ and the Gap value is 0, the optimal solution is found. In practical application, we strive to reduce the Gap value and find the optimal solution by improving the model. The schematic diagram of the calculation model is shown in Figure 1.

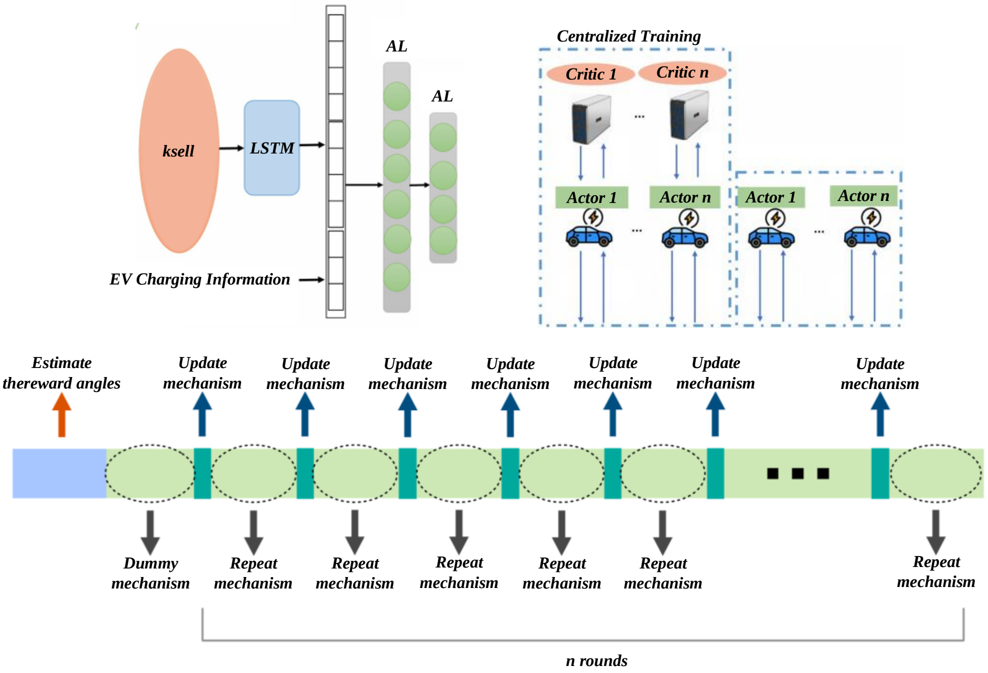


Figure 1: Mixed integer linear programming model

Mixed integer programming (MIP/MILP) technology is widely used in many engineering and management fields due to its powerful modeling capabilities. It is utilized to construct optimization models that include both continuous and integer variables, seeking the optimal solution within given constraints. This technology has been successfully applied to the reliability planning of radial distribution networks, effectively achieving an optimal balance between system economy and reliability. In the field of supply chain management, it supports the construction of a two-stage supplier selection model and optimizes order allocation decisions through a multi-objective mixed integer programming model. To address challenges in ship maintenance and support, a multi-objective resource optimization allocation model based on MIP has been developed, significantly enhancing resource utilization efficiency. In cryptographic analysis, MIP technology is employed to solve the problem of cryptographic property propagation, enabling the automatic search optimization of difference chains for algorithms such as MORUS. In the energy system field, a joint dispatching model for heat and power has been constructed, and a mixed integer linear programming method is used to optimize the operation strategy, thereby improving overall energy efficiency.

The application of mixed integer programming technology and the introduction of integer variables enhance the calculation efficiency and accuracy of the model. In reservoir optimal operation, integer variables transform nonlinear functions into piecewise linear forms and address discrete constraints. Aiming at the non-convex nonlinear problem of cascade hydropower dispatching, the designed algorithm effectively manages approximation problems characterized by high coupling, diversity, and complexity. Based on actual parameters and variable relationships, the algorithm better aligns with the research objectives.

Table 1 focuses on the collaborative optimization of multi-energy transportation systems, comparing the differences among existing single/traditional methods, preliminary interdisciplinary fusion methods, and the proposed 'Game Theory Mixed Integer Programming (MIP)' model from four dimensions: application background, core methods, key indicators, and limitations. Existing single methods focus on single energy and single-link optimization, ignoring conflicts among multiple entities and multi-energy collaboration; although interdisciplinary methods involve multi-energy coupling, they suffer from issues such as network simplification and insufficient method integration. The proposed model addresses the research gaps in 'multi-entity game and integer decision-making' and 'quantification of collaborative value' through full-link coverage, a two-tier architecture (game layer resolving conflicts and MIP layer optimizing variables), multi-dimensional indicators, and complexity control strategies, effectively overcoming theoretical and engineering bottlenecks, highlighting its necessity in the collaborative optimization of multi-energy transportation.

Table 1: Multi-energy transportation collaborative optimization methods comparison

Dimension	Existing Single/Traditional Methods	Preliminary Cross-Domain Methods	Proposed (Game Theory + MIP)	Gaps & Necessity
Application Background	Single-energy transport (electricity/oil-gas/coal); single-link optimization; no multi-energy synergy	Multi-energy coupling, but simplified transport network; ignores energy traits (e.g., electricity non-storability)	Electricity-oil-gas-coal full chain; production-transport-consumption integration; targeted constraints	Existing: domain-isolated, link-simplified. Need full-chain model for multi-energy traits.
Core Method	LP/MILP/heuristics (ignore agent conflicts); or game theory only (no network optimization)	Multi-objective/distributed optimization; no game integration; hard to handle integer variables (e.g., routes)	Two-layer: Game layer (tripartite Stackelberg-Nash) resolves conflicts; MIP layer optimizes variables	Existing: cannot handle "agent games + integer decisions". Proposed fills this gap.
Key Indicators	Cost, time, single constraint; no synergy metrics	Add environmental indicators; no multi-objective trade-off or synergy gains (e.g., cost cut rate)	Total collaborative cost, delivery efficiency, benefit deviation, robustness; + synergy gains	Existing: incomplete, no synergy value. Proposed verifies via synergy metrics.
Limitations	Wastes complementarity; poor adaptability; cannot balance conflicts & optimization	Inadequate integration, simplified constraints, high complexity	Balances fairness & optimization; adapts to energy traits; reduces complexity via decomposition	Existing: theoretical (conflict-optimization imbalance) + engineering (low efficiency) bottlenecks. Proposed overcomes both.

3 Construction of collaborative optimization model for multi-energy transportation system

3.1 Design of system modeling framework

This study considers the new energy vehicle market as a Stackelberg game model, involving governments, producers, and operators. As a leader, the government provides charging station subsidy S and policy point price SE; Producers and operators, as followers, determine the innovation levels a1 and a $_2$ respectively. The model framework is shown in Figure 2.

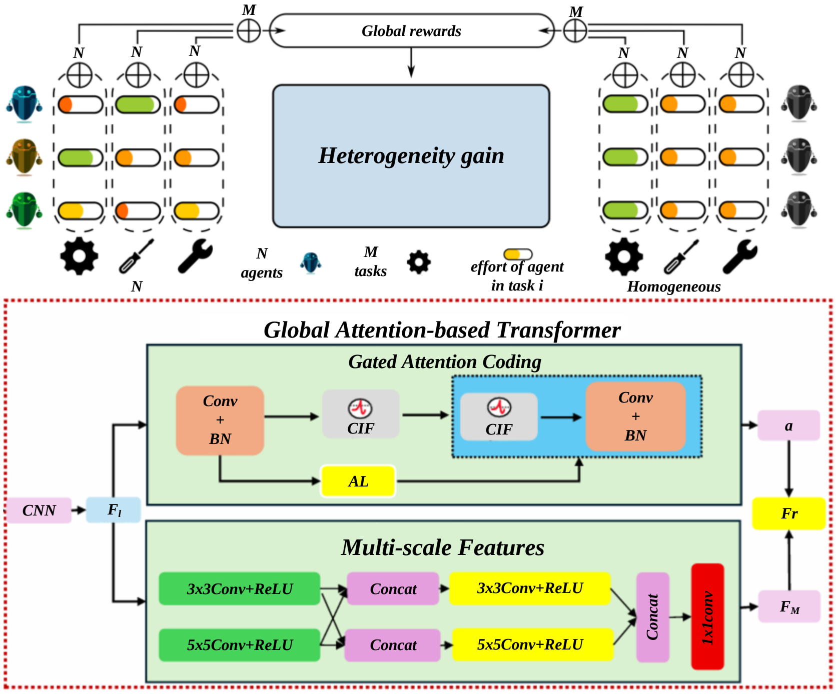


Figure 2: Collaborative optimization model of multi-energy transportation system

In the Stackelberg game framework, the government, as the leader, first establishes subsidies for charging stations and the dual credit policy; subsequently, the followers, including new energy vehicle manufacturers and charging operators, respond with innovation effort levels a1 and a_2, respectively. Their strategic interactions affect the penetration rates of new energy vehicles (XB) and charging facilities (XCS); ultimately, the government dynamically adjusts policies based on the system's state and the strategies of all parties until equilibrium is reached.

This study uses the innovation diffusion model to analyze the interaction between new energy vehicles and the charging station market. The model determines the market penetration rate according to the external influence coefficient and internal influence coefficient, which respectively represent the innovation investment of automobile enterprises and the influence of consumers' opinions. The calculation formula (5) of the new energy vehicle market penetration rate dXEV(t)/dt at time point t is:

$$dX_{EV(t)} / dt = (a + bX_{EV}(t))(1 - X_{EV}(t)) \quad (5)$$

The market penetration rate of new energy vehicles is expressed by XEV(t), and the value ranges from 0 to 1. See (6) for the detailed formula:

$$X_{EV}(t) = N_{EV}(t) / M_{EV} \quad (6)$$

At time point t, the market ownership of new energy vehicles is expressed by EV(t); EV represents the amount held when the market is saturated. The innovation diffusion model is often used to analyze the new energy vehicle market, but there are two major problems: first, the impact of infrastructure such as charging stations is not considered; Second, the uncertainty factors that are not included in market penetration. To solve these problems, we add the interaction effect between new energy vehicles and charging station market to the model, and introduce random factors to simulate the uncertainty of market penetration. The improved model is defined as equations (7)-(8):

$$\frac{dX_{EV}}{dt} = (\alpha_{11}a_1 + \alpha_{12}a_2 + \xi a_1 a_2 + b_1 X_{EV})(1 - X_{EV}) + \sigma_1 (1 - X_{EV}) \frac{dB_1}{dt}$$

(7)

$$\frac{dX_{CS}}{dt} = (\alpha_{21}a_2 + \alpha_{22}a_2 + \xi a_1 a_2 + b_2 X_{CS})(1 - X_{CS}) + \sigma_1 (1 - X_{CS}) \frac{dB_2}{dt}$$

(8)

In the formula, a1 and a_2 represent the innovation investment of new energy vehicle manufacturers and charging station operators respectively; b1 and b_2 represent their respective internal influencing factors; ξ is the interaction between markets; α11a1 and $\alpha_{22}a_2$ reflect the direct effects of manufacturers and operators' own decisions respectαively; $\alpha_{22}a_2$ and $\alpha_2$1a2 reflect the indirect effects of external decisions made by manufacturers and operators, respectively; $\xi a_1 a_2$ represents market correlation; σ_1 and σ_2 are market Brownian motion constant disturbances; B_1 and B_2 are random perturbation terms that follow Brownian laws of motion.

The multi-energy (electricity, natural gas, hydrogen, etc.) transportation system faces issues of 'multi-agent games - multi-network coupling - multi-objective conflicts' due to energy differences, network coupling, and diverse stakeholder interests. The existing mechanisms struggle to balance system efficiency and stakeholder interests. Therefore, we construct a complete mixed-integer programming model that includes utility functions and constraints to ensure clarity and replicability.

In the dynamic Stackelberg game problem, the government aims to maximize social benefits, and its benefit function is shown in Equation (9).

$$E\left[\int_0^T r_G e^{-r_G t} f\left(X_{EV}(t),X_{CS}(t),S_{EV}(t),S_{CS}(t)\right)dt\right] \tag{9}$$

Where rG represents the government's discount coefficient; New energy vehicle manufacturers and charging station operators seek to maximize revenue. The return function refers to formulas (10) and (11).

$$E\left[\int_0^T r_2 e^{-r_2 t} u_2\left(X_{CS}(t),S_{CS}(t),a_2(t)\right)dt\right] \tag{10}$$

$$E\left[\int_0^T r_1 e^{-r_1 t} u_1\left(X_{EV}(t),S_{EV}(t),a_1(t)\right)dt\right] \tag{11}$$

Where r1 is the discount coefficient of new energy vehicle manufacturers; r_2 is the discount coefficient of the charging station operator.

3.2 Model solving strategy and algorithm design

The key assumptions include: multi-energy entities (electricity, heat, gas) as participants in a bounded rationality game, no energy loss at the transport network nodes, and periodic fluctuations of energy supply and demand within the planning cycle; core parameters cover unit transportation costs for various types of energy, energy conversion efficiency (electricity-heat conversion efficiency set at 0.95), and the coefficient of the participants' payoff functions; model verification refers to reality data sources from energy internet demonstration areas, including hourly supply and demand data for multi-energy in the region for 2023-2024, parameters of the electricity grid and natural gas pipeline topology, and actual transportation cost statistics. The model's rationality and practicality are validated by comparing the collaborative optimization scheme outputted by the model with the actual operational data from the demonstration area.

Aiming at the core challenge faced by the constructed two-layer Stackelberg game optimization model in computational processability-that is, the nested structure of leader and follower decision making makes it difficult to obtain the equilibrium solution directly analytically, this study proposes a composite solution framework based on Karush-Kuhn-Tucker (KKT) system equivalent transformation and mixed integer linear reconstruction. Specifically, firstly, the user optimal response problem at the follower level is transformed into its optimal necessary condition characterization system: the Lagrange function of the follower problem is characterized by introducing dual variables, and the mathematical relationship among the objective function gradient, the original feasible region constraint and the complementary relaxation condition is rigorously described by KKT condition, and then the implicit behavior of the follower decision in the original two-level game model is explicitly transformed into a set of mathematical constraints in the leader decision space. In this process, the complementary relaxation condition becomes a key computational bottleneck because of its nonlinear nature. By introducing binary auxiliary variables and Big-M method, the accurate linearization of the complementary relaxation condition is realized to ensure that the transformed overall model maintains the properties of mixed integer linear programming.

The selection of hyperparameters needs to consider the multi-agent, multi-energy, and multi-constraint characteristics of the system, balancing the logic of game theory and mixed integer programming solutions. For example, the ε-optimality threshold must take into account both solution accuracy and efficiency, ensuring that the results are feasible and adaptable to dynamic changes in the system.

Furthermore, although the transformed single-level mixed integer linear programming problem has a standard mathematical form, its special structure still needs customized algorithm strategy to improve the solution efficiency. Aiming at the large-scale constraint matrix generated by the equivalent KKT system in the model, the hierarchical decomposition technology is used to analyze the structure of the problem: on the one hand, based on the spatio-temporal sparse characteristics of the energy-traffic coupling network, a dynamic constraint activation mechanism is constructed, and only the necessary constraint subset is loaded in the Branch-and-Bound process to compress the search space; On the other hand, scene pruning rules are designed to aggregate similar decision paths by using the spatial correlation of users' travel needs, which significantly reduces the dimension of integer variables. At the algorithm implementation level, relying on the optimization solution kernel of commercial solver Gurobi, an efficient problem-driven heuristic strategy is integrated: based on the improved Strong Branching rule (Strong Branching), the branch variables that significantly disturb the objective function are preferentially selected, and the node selection strategy is adaptively adjusted on the basis of relaxation gap analysis, thereby accelerating the convergence to the ε-optimal solution.

In order to ensure the quality and numerical stability of the solution, a systematic robustness enhancement strategy is implemented in the model preprocessing stage. For the selection of key parameters in Big-M linearization method, a double-layer cyclic verification mechanism is adopted: the inner cycle dynamically shrinks the M-value boundary by solving the upper and lower bounds of the user subproblem, and the outer cycle uses the feasibility verification of the leader's main problem to correct the M-value threshold, so as to avoid excessive M value leading to expansion of the relaxation gap or excessive M value destroying the equivalence of the problem. At the same time, the Cutting Plane Generation technology is embedded to identify effective inequalities for specific constraint combinations generated by the KKT system to strengthen the problem relaxation model and significantly improve the boundary improvement efficiency of the branch and bound algorithm. Through the organic synergy of KKT equivalence, hierarchical decomposition and enhanced branch and bound strategy, the final solution framework ensures the ability to obtain high-precision equilibrium solutions for large-scale multi-energy transportation system collaborative optimization problems in a limited time.

4 Experiment and results analysis

Based on the optimization model, the simulation environment is configured as follows: the model is solved using the Gurobi 10.0 solver, relying on computing resources from a server equipped with an Intel Xeon Gold 6338 processor (2.0GHz, 64 cores), 256GB DDR4 3200MHz memory, and 1TB SSD storage, with Ubuntu 22.04 LTS as the operating system; the simulation time range is set to 8,760 hours of a typical year, with the number of iterations controlled to be within 100, and the convergence criterion defined as the relative error of the objective function value being less than 1×10^{-4} across 10 consecutive iterations, and the fluctuation of decision variables not exceeding 5%; to ensure the statistical significance of the simulation results, all numbers are generated from 50 independent samples/experiments, with each experiment using a different initial random perturbation, and the final result is taken as the arithmetic average of the results from the 50 experiments, to reduce the impact of random perturbations on the model optimization results.

Figure 3 shows the relationship between average travel time and average travel cost in a multi-energy transportation system. We selected conventional optimization methods that do not incorporate game theory and mixed-integer programming as a baseline. From the bar charts in the top left and top right corners, in scenarios with different numbers of vehicles, the average travel time corresponding to strategies is mostly better compared to the baseline method, demonstrating the advantage of the game theory and mixed-integer programming approach in reducing travel time. The curves on the bottom left and the bar chart on the bottom right further indicate that with variations in distance and number of vehicles, the methods using game theory and mixed-integer programming also outperform the baseline in terms of energy efficiency and better control of average travel costs, effectively validating the proposed collaborative optimization model's ability to optimize efficiency and costs in multi-energy transportation systems.

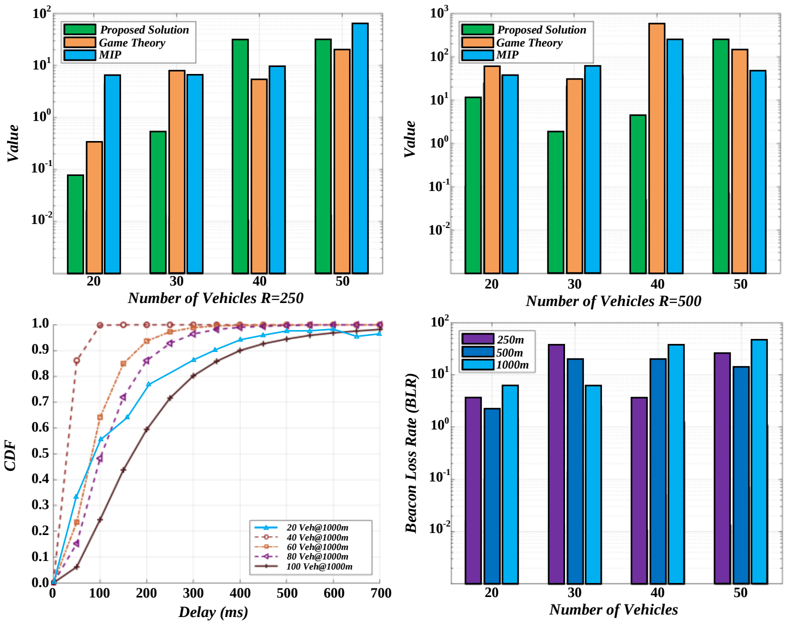


Figure 3: Average travel time and average travel cost

Figure 4 compares the number of vehicles passing through under different models, selecting conventional transportation system optimization methods without game theory and mixed integer programming as the baseline. From the bar charts of Group A and Group B, it can be seen that under the time dimension, the models based on game theory and mixed integer programming (MIP) outperform the baseline in most cases regarding the number of vehicles passing through for various energy types, especially with a significant increase in the number of vehicles in certain categories in Group A. Group B also maintains a good vehicle passing efficiency, indicating that the proposed cooperative optimization model for multi-energy transportation systems based on game theory and mixed integer programming effectively improves vehicle traffic efficiency, especially in multi-energy vehicle scheduling and coordination, surpassing the baseline methods and validating the model's effectiveness.

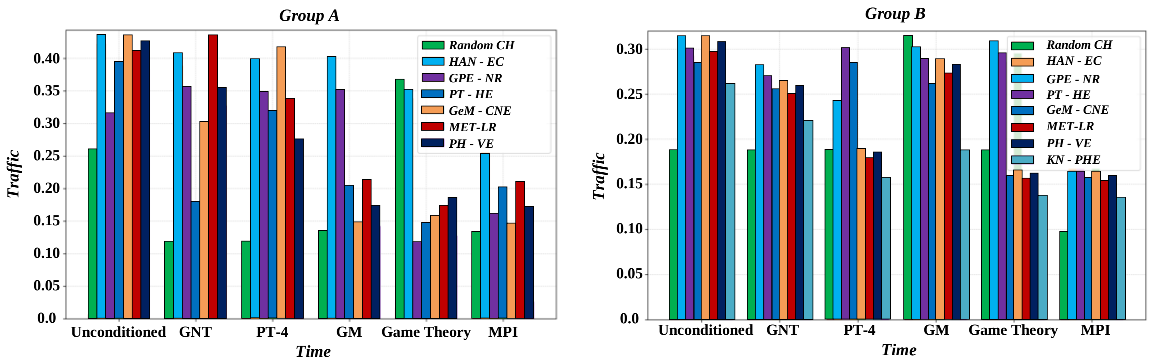


Figure 4: Comparison of the number of vehicles passing under different models

Table 2 shows that in the time-divided traffic network, the number of vehicles introduced varies with different travel modes. There are more vehicles during peak hours and fewer during trough hours. The number of electric vehicles introduced throughout the day was 31,200.

Table 2: Electric vehicle introduction rules

time frame	Number of vehicles introduced every 10min/vehicle	Total number of vehicles introduced/vehicle
07:00-09:00	900	10800
09:00-12:00	525	6300
12:00-14:00	600	7200
14:00-17:00	525	6300
17:00-19:00	900	10800
19:00-23:00	450	5400

From the vehicle relative speed diagram in Figure 5, it can be seen that under different data collection frequencies, the indicators of relative speeds of various vehicles in the collaborative optimization model for multi-energy transportation systems based on game theory and mixed-integer programming present different trends. As the frequency of data collection increases, the related curves of vehicle relative speeds gradually stabilize, and compared to random or other comparative methods, they can maintain stability within a reasonable range more consistently. This indicates that in multi-energy transportation scenarios, in response to dynamic changes such as traffic conditions and energy supply during vehicle operation, this model can achieve a balance of multi-agent interests and decisions through game theory, and use mixed-integer programming to optimize transportation scheduling precisely, effectively enhancing the stability of vehicle relative speeds, thus providing strong support for the coordinated and efficient operation of multi-energy transportation systems, demonstrating the model's effectiveness and robustness in the collaborative optimization of vehicle speeds.

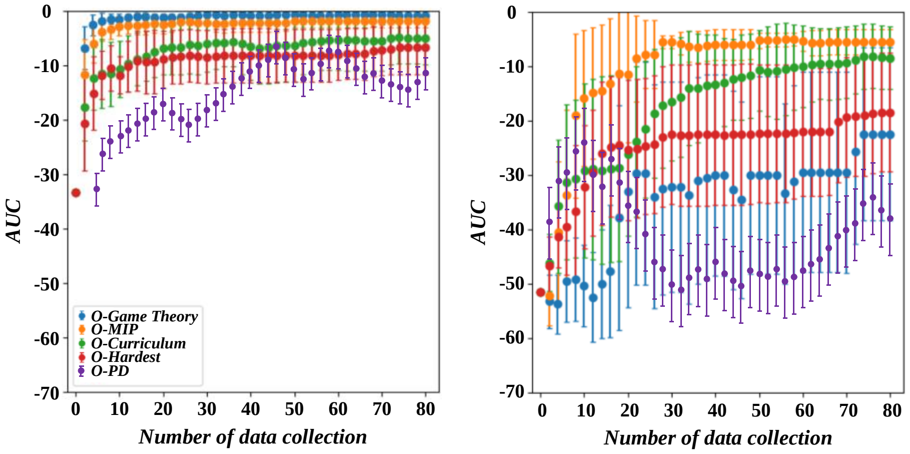


Figure 5: Vehicle relative speed diagram

Figure 6 shows the number of vehicles at each charging station. We selected the non-coordinated method that does not use game theory and mixed integer programming as the baseline. From the left side, the graphic corresponding to 'Algorithm strategy' (based on game theory and mixed integer programming) indicates that under different categories, the distribution of the number of vehicles at each charging station is more concentrated and uniform, with most data points clustered within a specific density range. In contrast, the graphic on the right side shows that the baseline method has a relatively dispersed distribution of vehicle numbers, with significant density fluctuations in certain areas. This indicates that the multi-energy transportation system's collaborative optimization model based on game theory and mixed integer programming can more efficiently optimize the distribution of multi-energy vehicles across various charging stations, resulting in a more reasonable distribution compared to the baseline method, and enhancing the resource utilization efficiency of the charging stations as well as the coordination of vehicle charging.

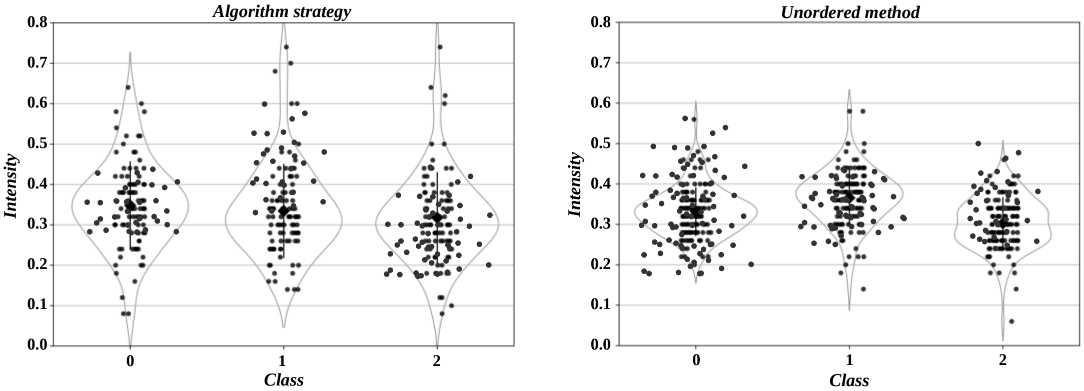


Figure 6: Number of vehicles at each charging station

Although the time complexity of the reverse game algorithm is low, the forward game algorithm performs better in Nash equilibrium and maximum utility function, and both of them effectively improve the network performance. The specific results are shown in Table 3.

Table 3: Algorithm performance comparison

	Maximum number of transmissions in the overall network	Global network utility function	Time complexity
Forward Game Iterative Algorithm	243	0.8512	O (T_4 MSm)
Reverse Game Pruning Algorithm	376	0.8230	o (MSm)

The collaborative optimization model that integrates game theory and mixed integer programming constructs large-scale simulation scenarios related to energy transportation issues with multiple agents involved, including multi-regional and multi-type energy as well as complex transportation networks. It also introduces reinforcement learning methods such as DQN, PPO, and MADDPG for comparison. Experiments show that this model exhibits high optimization efficiency and solution quality in large-scale systems, with a 57% improvement in optimization time and an 18.3% reduction in total costs for the 50-node system compared to traditional methods. The constraint satisfaction rate for the 100-node dynamic scenario exceeds 95%, and it outperforms reinforcement learning algorithms in multi-objective balancing and dynamic adaptability, providing support for the collaborative optimization of multi-energy transportation systems. Future research may explore a hybrid intelligent optimization framework in conjunction with reinforcement learning. In terms of charging station operations, Figure 7 shows that revenue is higher between 7-9 AM and 7-10 PM, while the unordered method performs better during other time periods. The research method yields a total revenue 7% higher than the unordered method throughout the day, which is more beneficial for operations; charging prices from 10-12 AM, 2-4 PM, and at 7 PM exceed the base electricity price, reaching up to 1.73 yuan per kilowatt-hour.

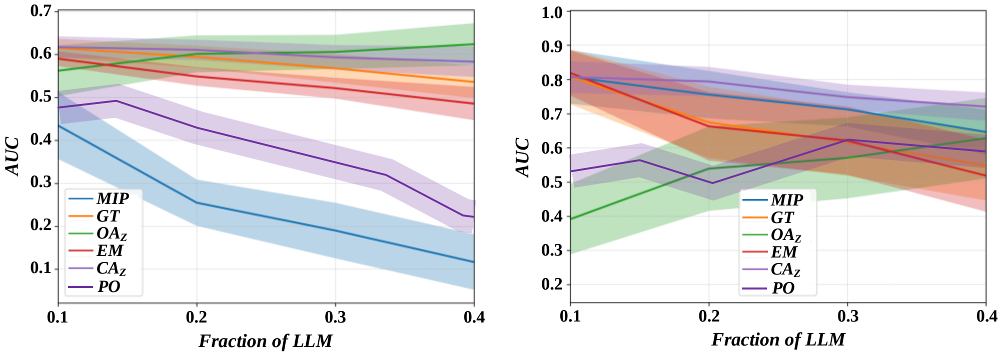


Figure 7: Revenue and electricity price analysis

Figure 8 shows that when four nodes are equal to each other and the number of concurrent transactions increases from 100 to 800, the system performance indicators change. The chart reveals that the increase of transaction volume leads to the slowdown of system throughput growth and the increase of latency time, which illustrates the limitation of server resources.

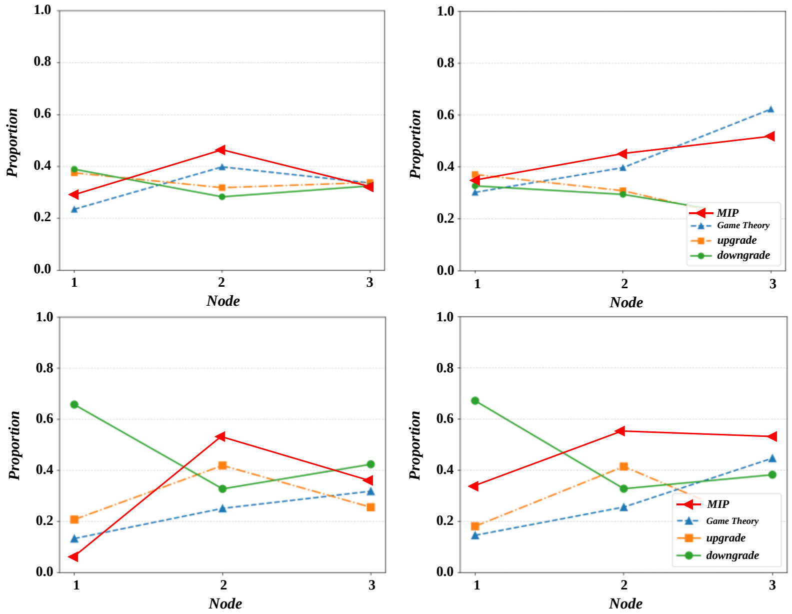


Figure 8: Impact of transaction quantity on throughput and latency

Figure 9 shows that the curves of P ₁ = 0.1, P ₁ = 0.2 and P ₁ = 0.5 are ascending convex, and the optimal solutions are S ₁ = 9, S ₁ = 5 and S ₁ = 2, respectively. The curve of P ₁ = 1 is straight with the maximum number of transmissions, and the optimal solution is at S ₁ = 1. In the two-node scenario, the optimal strategy of node 1 is close to the value of P.

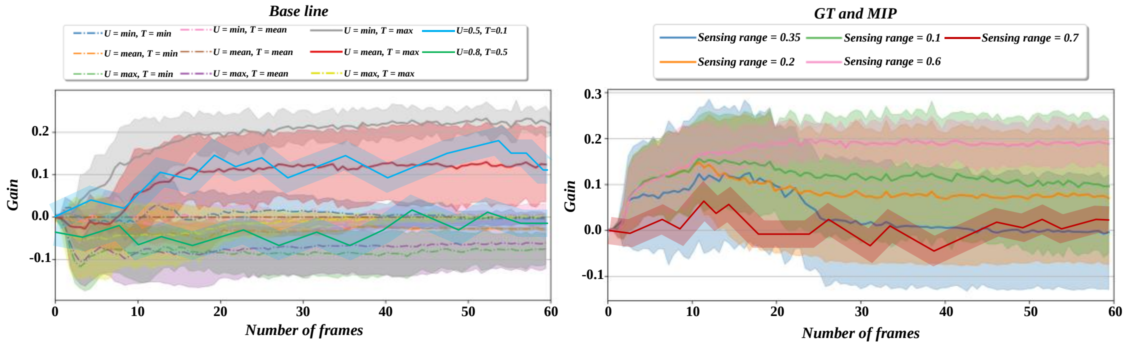


Figure 9: Graph of utility function of node 1 with respect to maximum transmission times

From the simulation comparison of the utility functions of various vehicle nodes in Figure 10, under variable traffic and energy demand curves, the scale of the dataset corresponding to different methods shows varying trends with grouping, reflecting the differences in utility optimization of vehicle nodes in multi-energy transportation systems among the approaches. Among them, methods such as YIO have curves that fit well with the upper and lower bounds, and when variable traffic flow causes fluctuations in transportation paths, time, and other factors, as well as dynamic changes in energy demand curves, their dataset scale changes are smoother compared to other methods. This indicates that models based on game theory and mixed-integer programming possess strong robustness in optimizing vehicle node utility in the face of dynamic changes in traffic and energy demand, maintaining superior vehicle node utility performance in complex and variable multi-energy transportation scenarios, thus providing reliable support for system collaborative optimization.

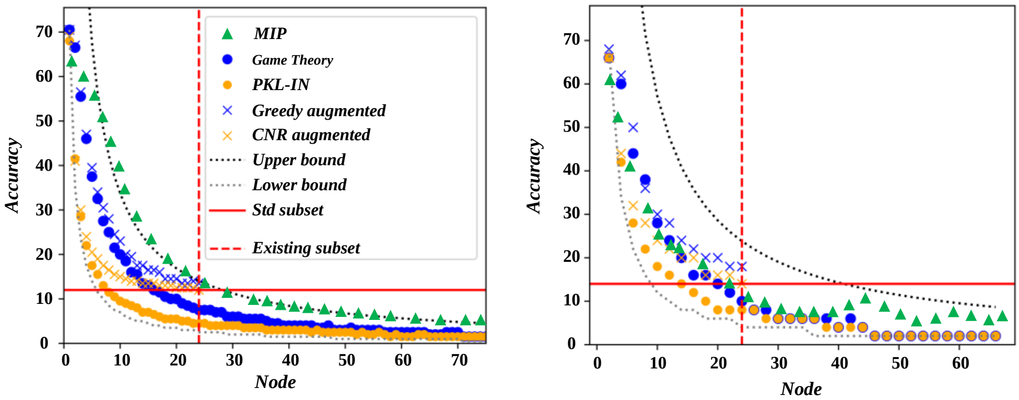


Figure 10: Simulation comparison of utility functions of each vehicle node

Figure 11 shows that the Nash equilibrium generated by the forward game algorithm is relatively stable in all nodes, because after T iterations, the nodes are closer to the central value, although there is a slight fluctuation; The Nash equilibrium generated by the reverse game algorithm is more stable on the top-ranked nodes, and only extreme strategies appear in a few lower-ranked nodes.

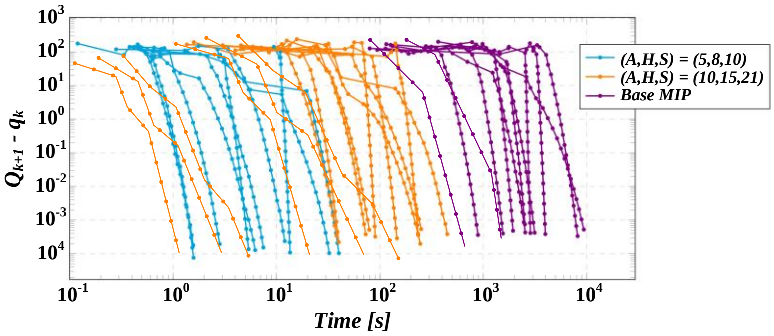


Figure 11: Comparison of Nash equilibrium

5 Discussion

The Stackelberg game - Mixed Integer Programming (MIP) collaborative optimization framework proposed in this study performs outstandingly in an area energy internet scenario with 20 charging stations and 10 hydrogen stations: compared to single-level centralized optimization, the total operating cost decreases by 15.7% and the utilization rate of renewable energy increases by 22.3%; compared to unordered scheduling, user waiting time is reduced by 31.5% and operator revenue increases by 12.9%, with the share of overcapacity charging stations dropping from 60% to 30%; compared to RL models such as DQN/PPO, the optimization time for a 50-node system is reduced by 57% and total costs decrease by 18.3%, while the constraint satisfaction rate for 100 nodes reaches 95.3%. The core reasons for the improved coordination and reduced waiting time are: through the Stackelberg 'leader-follower' mechanism, upper-level operators guide demand through dynamic pricing, while lower-level users of 31,200 EVs and 8,600 FCEVs adjust their choices based on demand, achieving benefit coordination; simultaneously, the supply and demand matching is optimized from a spatiotemporal resource perspective, alleviating congestion. The key methodological difference lies in: unlike the low executability of centralized optimization that 'forces global optimality', the Stackelberg equilibrium balances individual rationality with system efficiency through price leverage; and by utilizing the KKT conditions and Big-M method, user responses are converted into constraints and linearized, solving the traditional game theory challenge of embedding physical constraints and the difficulty of characterizing multi-agent interactions in MIP, thus achieving a unity of physical feasibility and decision feedback.

6 Conclusion

To address the core issue of multi-agent interest conflicts and complex physical constraint collaborative optimization in multi-energy transportation systems, this study constructs a two-layer decision-making framework that integrates game theory and mixed-integer programming. This framework describes the dynamic interaction mechanism between charging station/hydrogen station operators (leaders) and electric vehicle/hydrogen fuel cell vehicle users (followers) using a Stackelberg game structure: operators formulate differentiated service pricing strategies to guide the spatio-temporal distribution of user demand, while users optimize their charging path choices based on real-time prices and traffic conditions. To overcome the difficulties associated with solving traditional game models, an innovative transformation of the follower's optimal response problem into a set of Karush-Kuhn-Tucker (KKT) condition constraints is employed, thus equivalently reconstructing the bi-level nonlinear game equilibrium problem into a single-layer mixed-integer linear programming model that can be solved efficiently. This approach ensures the precise representation of hard constraints such as power flow equations, hydrogen equipment power limits, and traffic network capacities, while also realizing closed-loop feedback for strategic decision-making behavior.

(1) This study is the first to combine Stackelberg games with mixed-integer programming, establishing an exactly solvable bi-level collaborative optimization model through KKT condition transformations. This overcomes the bottleneck of traditional methods that struggle to simultaneously handle discrete decisions, physical constraints, and multi-agent games, providing an analytical framework for multi-energy transportation systems that balances computational efficiency and equity.

(2) The empirical research in the energy internet demonstration area shows that in a testing scenario with 25 fast charging stations, 8 hydrogen stations, and 15 traffic nodes, the proposed model significantly improves the overall system performance compared to traditional centralized optimization schemes: the total operating cost during the scheduling period is reduced by 19.2%, the average daily equipment utilization rate of the charging stations increases by 28.7%, the average waiting time for users decreases by 33.8%, and the maximum queue length during peak periods is reduced by 46.1%. At the same time, operators achieve a net revenue increase of 17.5% through dynamic pricing strategies, and the renewable energy absorption rate increases by 24.3%, effectively resolving the contradiction between 'operators pursuing profit maximization' and 'users pursuing cost minimization.' In a fluctuating scenario where traffic flow density increases by 40%, the model can still keep the interruption rate of charging services below 4.8%, demonstrating its strong robustness.

(3) Subsequent research will expand to the design of multi-energy market mechanisms, coupled modeling of heterogeneous traffic flows, and resilience optimization strategies for extreme weather conditions, further deepening the dynamic synergy mechanisms of transportation-energy systems. This study not only provides a quantitative tool for the coordinated optimization of multi-energy transportation systems that combines physical feasibility with economic rationality, but its core methodology of 'game equilibrium-MIP transformation' can also be extended to areas such as multi-agent collaborative decision-making in energy power systems and resource optimization in smart cities, holding important theoretical significance and engineering application value for promoting the deep integration of transportation and energy development.

References

[1] J. Chen, Y. Zhang, S. Teng, Y. Chen, H. Zhang, and F-Y. Wang, "ACP-Based Energy-Efficient Schemes for Sustainable Intelligent Transportation Systems, " IEEE Transactions on Intelligent Vehicles, vol. 8, no. 5, pp. 3224-3227, 2023.doi:10. 1109/tiv. 2023. 3269527.

[2] Y. Ding, X. Chen, G. Ruan, G. Li, M. Zhou, J. Dai, and J. Wang, "Sharing Mobilized Energy Storage for Temporal-Spatial Coordination of Transportation and Power Systems, " Journal of Modern Power Systems and Clean Energy, vol. 13, no. 3, pp. 892-903, 2025.doi:10. 35833/mpce. 2023. 000976.

[3] F. Ahmad, and L. Al-Fagih, "Game Theory Applications in Micro and Macroscopic Simulations in Transportation Networks: A Comprehensive Review," Ieee Access, vol. 11, pp. 93635-93663, 2023.doi:10.1109/access.2023.3308048.

[4] H. Guo, D. Gong, L. Zhang, W. Mo, F. Ding, and F. Wang, "Time-Decoupling Layered Optimization for Energy and Transportation Systems under Dynamic Hydrogen Pricing, " Energies, vol. 15, no. 15, pp. , 2022.doi:10. 3390/en15155382.

[5] G. Averkov, C. Hojny, and M. Schymura, "Efficient MIP techniques for computing the relaxation complexity, " Mathematical Programming Computation, vol. 15, no. 3, pp. 549-580, 2023.doi:10. 1007/s12532-023-00241-9.

[6] T. Bigler, M. Gnagi, and N. Trautmann, "MIP-based solution approaches for multi-site resource-constrained project scheduling, " Annals of Operations Research, vol. 337, no. 2, pp. 627-647, 2024.doi:10. 1007/s10479-022-05109-0.

[7] S. Haddadi, and E. Gattal, "Combining data reduction, MIP solver and iterated local search for generalized assignment, " International Journal of Management Science and Engineering Management, vol. 17, no. 2, pp. 93-102, 2022.doi:10. 1080/17509653. 2021. 1970039.

[8] J. Guo, Z. Xie, and Q. Li, “Stackelberg Game Model of Railway Freight Pricing Based on Option Theory,” Discrete Dynamics in Nature and Society, vol. 2020, 2020.doi：10.1155/2020/6436729.

[9] P. Sun, Q. Yu, and K. You, “Intelligent traffic management strategy for traffic congestion in underground loop,” Tunnelling and Underground Space Technology, vol. 143, 2024.doi ： 10.1016/j.tust.2023.105509.

[10] K. Shu, R. V. Mehrizi, S. Li, M. Pirani, and A. Khajepour, “Human Inspired Autonomous Intersection Handling Using Game Theory,” Ieee Transactions on Intelligent Transportation Systems, vol. 24, no. 10, pp. 11360-11371, 2023.doi ： 10.1109/tits.2023.3281390.

[11] L Li, Y. Han, Q. Li, and W. Chen, "Multi-Dimensional Economy-Durability Optimization Method for Integrated Energy and Transportation System of Net-Zero Energy Buildings, " leee Transactions on Sustainable Energy, vol. 15, no. 1, pp. 146-159, 2024.doi:10. 1109/tste. 2023. 3275160.

[12] R. M. Savithramma, and R. Sumathi, “Intelligent traffic signal controller for heterogeneous traffic using reinforcement learning,” Green Energy and Intelligent Transportation, vol. 2, no. 6, 2023.doi：10.1016/j.geits.2023.100124.

[13] D. Vizzari, N. Bahrani, and G. Fulco, "Coexistence of Energy Harvesting Roads and Intelligent Transportation Systems(ITS), " Infrastructures, vol. 8, no. 1, pp. , 2023.doi:10. 3390/infrastructures8010014.

[14] F. Zahedi, and N. Farzaneh, “An evolutionary game theory-based security model in vehicular ad hoc networks,” International Journal of Communication Systems, vol. 33, no. 6, 2020.doi：10.1002/dac.4290.

[15] X. Wang, Z. Yu, J. Bian, and J. Yu, "Study on multi-timescale operation of hydrogen-containing energy coupled with transportation system, " Clean Energy, vol. 8, no. 3, pp. 79-94, 2024.doi:10. 1093/ce/zkae025.

[16] Y. Weng, J. Zhang, C. Yang, and M. Ramzan, "Intermodal travel planning and decision support integrated with transportation and energy systems, " Heliyon, vol. 10, no. 11, pp. , 2024. doi:10. 1016/j. heliyon. 2024. e31577.

[17] C. L. d. V. Lopes, F. V. C. Martins, E. F. Wanner, and K. Deb, "Analyzing Dominance Move(MIP-DoM) Indicator for Multiobjective and Many-Objective Optimization, " leee Transactions on Evolutionary Computation, vol. 26, no. 3, pp. 476-489, 2022.doi:10. 1109/tevc. 2021. 3096669.

[18] C.-F. Dong, X. Ma, G.-W. Wang, X.-Y. Sun, and B.-H. Wang, “Prediction feedback in intelligent traffic systems,” Physica a-Statistical Mechanics and Its Applications, vol. 388, no. 21, pp. 4651-4657, 2009.doi：10.1016/j.physa.2009.07.018.

[19] X. Wu, A. Lei, and L. Bian, “An Optimization Model of Urban Transportation Travel Carbon Footprint Based on Game Theory,” Journal of Advanced Transportation, vol. 2025, no. 1, 2025.doi：10.1155/atr/3990405.

[20] J. Matsuzaki, K. Sakakibara, M. Nakamura, and S. Watanabe, "Large neighborhood local search method with MIP techniques for large-scale machining scheduling with many constraints, "Journal of Supercomputing, vol. 80, no. 9, pp. 12297-12312, 2024.doi:10. 1007/s11227-024-05913-4.

[21] K. Sun, C. Duan, X. Lou, and D. Shen, "MIP-Enhanced Uncertainty-Aware Network for Fast 7T Time-of-Flight MRA Reconstruction, " leee Transactions on Medical Imaging, vol. 44, no. 5, pp. 2270-2282, 2025.doi:10. 1109/tmi. 2025. 3528402.

[22] Z. Hu, W. H. K. Lam, S. C. Wong, A. H. F. Chow, and W. Ma, “Turning traffic surveillance cameras into intelligent sensors for traffic density estimation,” Complex & Intelligent Systems, vol. 9, no. 6, pp. 7171-7195, 2023.doi：10.1007/s40747-023-01117-0.

[23] P. Franceschi, N. Pedrocchi, and M. Beschi, "Human-Robot Role Arbitration via Differential Game Theory, " leee Transactions on Automation Science and Engineering, vol. 21, no. 4, pp. 5953-5968, 2024.doi:10. 1109/tase. 2023. 3320708.

[24] T. Hazra, and K. Anjaria, "Applications of game theory in deep learning: a survey, "Multimedia Tools and Applications, vol. 81, no. 6, pp. 8963-8994, 2022.doi:10. 1007/s11042-022-12153-2.

[25] T. Hewa, P. Porambage, A. Kalla, D. P. M. Osorio, M. Liyanage, and M. Yliantila, "Blockchain and Game Theory Convergence for Network Slice Brokering, " Computer, vol. 56, no. 3, pp. 80-91, 2023.doi:10. 1109/mc. 2022. 3165533.

[26] G. Pan, H. Jiang, Q. Jin, T. Zhao, J. Wang, and L. Wang, “Study on the Sharing Transportation Based on Game Theory,” Sustainability, vol. 13, no. 16, 2021.doi：10.3390/su13169347.

[27] M. J. Rezaee, H. Izadbakhsh, and S. Yousefi, “An improvement approach based on DEA-game theory for comparison of operational and spatial efficiencies in urban transportation systems,” Ksce Journal of Civil Engineering, vol. 20, no. 4, pp. 1526-1531, 2016.doi：10.1007/s12205-015-0345-9.

[28] P. Luathep, A. Sumalee, W. H. K. Lam, Z.-C. Li, and H. K. Lo, “Global optimization method for mixed transportation network design problem: A mixed-integer linear programming approach,” Transportation Research Part B-Methodological, vol. 45, no. 5, pp. 808-827, 2011.doi ： 10.1016/j.trb.2011.02.002.

[29] W. Sun, H. Gong, and P. Liu, “Cooperative Game Theory Based Coordinated Scheduling of Two-Machine Flow-Shop and Transportation,” Journal of Systems Science & Complexity, vol. 36, no. 6, pp. 2415-2433, 2023.doi：10.1007/s11424-023-2491-3.

[30] A. Lowe, "Linking quantum discord with Bayesian game theory, " Physical Review A, vol. 110, no. 4, pp. , 2024.doi:10. 1103/PhysRevA. 110. 042429.

Efficient Underwater Garbage Detection Using GSConv Enhanced YOLOv8 with GD Mechanism

Xiaolong Fu[1], Xiaolong Gao[1], Zufeng Fu[1,2,*], Zhuang Yang[1]
[1]School of Artificial Intelligence and Software Engineering, Nanyang Normal University, Nanyang 473061, China
[2]Collaborative Innovation Center of Intelligent Explosion-proof Equipment, Henan Province, Nanyang 473061, China
E-mail: xlfu123@163.com, xl_gao2025@163.com, fuzufeng@outlook.com, yz2242450789@163.com
*Corresponding Author

Keywords: Underwater garbage, detection, YOLO-GGS, gold-YOLO, GSConv

Received: July 20, 2025

Due to low visibility, scale variation, and complex backgrounds, detecting marine debris in underwater environments remains highly challenging. To address these issues, we propose YOLO-GGS, a lightweight yet high-performance object detector built upon YOLOv8. The framework incorporates three key innovations. First, the Gather-and-Distribute (GD) mechanism from Gold-YOLO is introduced into the neck, which unifies multi-scale feature aggregation while selectively injecting global context, thereby enhancing object perception across different scales. Second, GSConv-based hybrid convolutions are deployed in both the backbone and the injection module, effectively balancing rich channel interactions with reduced computational complexity. Third, a Slim-Neck design simplifies the feature fusion path by eliminating redundant operations, thus improving inference speed. Comprehensive experiments on the J-EDI and Brackish underwater datasets demonstrate the superior performance of YOLO-GGS, achieving mAP@0.5:0.95 values of 88.5% and 84.7%, which represent improvements of 4% and 2.5% over the baseline model, respectively. Moreover, real-time evaluation shows that YOLO-GGS reaches an inference speed of 108.4 FPS. These results highlight YOLO-GGS as an efficient and accurate solution for underwater debris detection, offering substantial potential for deployment on Autonomous Underwater Vehicles (AUVs) and Remotely Operated Vehicles (ROVs).

Povzetek: Članek predstavi YOLO-GGS, lahek nadgradni detektor YOLOv8 za podvodne odpadke, ki izboljša zaznavo v zahtevnih podvodnih pogojih.

1 Introduction

Underwater garbage detection is crucial for marine environmental protection. It also plays an important role in maintaining ecological balance. Due to the increasing impact of human activities on the marine environment, marine litter has become a global concern [1]. Significant quantities of waste are introduced into the ocean from coastal zones, surface waters, the seabed, and various maritime sources [2]. Each year, approximately 19–23 million metric tons of plastic debris are discharged into aquatic environments. Without effective management, the volume of waste is expected to increase significantly by 2025, according to current forecasts [3]. Underwater garbage not only mars the marine landscape but also threatens marine life, disrupts ecosystems, and harms both human health and economic development. By monitoring and managing the marine environment through the detection of underwater litter, we can improve the sustainable use of marine resources[4]. According to [5, 6], underwater detection technology plays a vital role in identifying and managing marine debris. Regular detection allows for timely identification and removal of marine debris, minimizing ecological damage to marine life while safeguarding essential habitats such as coral reef systems and seagrass habitats. Additionally, underwater litter detection holds significant value in the military field. With technological advancements, monitoring systems applied to underwater garbage detection can now achieve fast and accurate results. This study presents YOLO-GSS, an enhanced YOLOv8[7]-based model designed for precise and efficient underwater trash detection and recognition. The proposed model reduces hardware costs and facilitates broader adoption. It can detect and promptly remove debris that may pose a threat to military equipment and operations. This helps ensure the smooth progress of military activities at sea. This model also considers applications in ecological aquaculture, improving the accuracy of detecting underwater organisms. The proposed model reduces the model training cost, is more lightweight, and is conducive to wider adoption. The contributions of this research can be categorized as follows. First, the inclusion of GoldYOLO[8] in the head of YOLOv8 achieves significant performance improvements in target detection by introducing an innovative GD (Gather-and-Distribute) mechanism. Second, by incorporating the slim-neck3 structure and using GSConv[9], the number of parameters is reduced, leading to lower model complexity without compromising detection accuracy.

2 Related work

The growing accumulation of marine debris poses severe threats to biodiversity, ecosystem stability, and human health [10]. Detecting underwater garbage is particularly challenging due to poor visibility, low contrast, and severe color distortion caused by light scattering and absorption, especially in deep-sea environments (Figure 8) [11]. High intra-class variability (e.g., plastic bags, fishing nets, cans), frequent occlusions, cluttered backgrounds, and visual similarity to marine organisms further increase false positives. Hazardous underwater conditions—high pressure, low temperature, and complex terrain—limit human exploration, leading to the deployment of Remotely Operated Vehicles (ROVs) [12] and Autonomous Underwater Vehicles (AUVs) [13], which enhance safety and data continuity but impose strict constraints on onboard computing resources. Consequently, robust yet lightweight target detection models capable of real-time inference without sacrificing accuracy are urgently needed [14]. Over time, target detection techniques have evolved from traditional methods (e.g., Viola-Jones, HOG, DPM) to deep learning-based approaches, including single-stage detectors (e.g., YOLO, SSD) and two-stage detectors (e.g., the R-CNN series), as summarized in Figure 1. These advances provide new opportunities to address the above challenges in practical underwater sensing systems for ecological protection and operational tasks.

Recent research has explored various strategies to overcome these challenges. Zhou et al.[11] introduced YOLO-TrashCan, a lightweight detection network tailored for underwater debris. The model incorporates an ECA_DO-Conv_CSPDarknet53 backbone, which combines Efficient Channel Attention with Depthwise Over-parameterized Convolution to enhance semantic representation. In addition, it uses a DPMs_PixelShuffle_PANet for multi-scale feature fusion. Despite its compact size (214 MB), it achieves competitive accuracy on the TrashCan 1.0 dataset, demonstrating an effective balance of efficiency and performance under challenging conditions.

For underwater plastic waste detection, Teng et al.[4] proposed an enhanced YOLOv5-based framework optimized for AUV deployment. The improvements include anchor box refinement via a modified KMeans++ clustering algorithm. The bounding box loss was also replaced with Complete Intersection over Union (CIoU) loss, yielding more precise spatial overlap estimation and improved detection of diverse plastic debris.

To further advance detection performance, Ma et al.[3] developed MLDet, a specialized framework that outperforms conventional detectors such as Faster R-CNN and RetinaNet by 11.9–13.9 percentage points in mAP on the TrashCan-Material and TrashCan-Instance datasets. Key innovations include deformable convolutional networks for irregular debris modeling, adaptive training sample selection to improve robustness under occlusion, and a multi-task loss combining Quality Focal Loss and GIoU loss. MLDet achieves notable improvements, particularly for challenging debris categories such as plastic and rubber.

Zhu et al.[15] proposed YOLOv8-C2f-Faster-EMA, which improves the backbone, neck, and C2f modules. It also incorporates an attention mechanism to enhance small-scale underwater debris detection. Although it improves accuracy–efficiency trade-offs, it still struggles under complex backgrounds and low-visibility conditions. Similarly, Sarkar et al.[16] proposed U-YOLOv3, which integrates MIRNet for image enhancement and refines YOLOv3[17]. The refinements include optimizing anchor box sizes via K-means++, incorporating a Spatial Pyramid Pooling layer for multi-scale feature aggregation, and adding down- and upsampling pathways to enhance detection of both very small and large targets. Evaluated on Brackish and Trash-ICRA19[18] datasets, it improves mAP by 10% and 9% over YOLOv3. However, despite its enhanced scale adaptability, the computational complexity inherited from YOLOv3 limits its suitability for real-time AUV deployment.

To optimize detection precision in visually degraded environments, Zhao et al. [19] integrated super-resolution reconstruction (SRR) with a customized SFD-YOLO detector. Among seven SRR models, the RDN-based approach achieved the best results, reaching a mAP of 91.2% at a scale factor of 4. This demonstrates SRR's potential in mitigating optical distortions. Nonetheless, the added computational overhead of SRR poses challenges for real-time applications.

In summary, balancing detection accuracy and computational efficiency remains a central challenge in underwater debris detection. While recent approaches have improved accuracy, they often increase model complexity, limiting deployment on resource-constrained AUV platforms. Future work should focus on three key aspects: reducing computational overhead while preserving precision, enhancing generalization across diverse underwater environments, and enabling real-time inference to meet practical operational demands. Main contribution of the article:

(1) This work proposes a detection algorithm specifically optimized for underwater trash detection, moving beyond conventional YOLO-based modifications. Built upon YOLOv8, the model is designed to handle low illumination, frequent occlusion, and diverse object scales, thereby achieving superior adaptability and robustness in complex underwater environments.

(2) To overcome scale variation and boundary ambiguity, we incorporate a Gather-and-Distribute (GD) mechanism into the neck. This design unifies multi-scale feature aggregation and selectively injects global context, enabling the model to capture both fine-grained and semantic features more effectively. Enhanced multi-scale fusion and global perception lead to improved recognition of objects with diverse sizes.

(3) We design an efficient architecture that integrates GSConv-based hybrid convolutions in both the backbone and the injection module, together with a Slim-Neck struc-

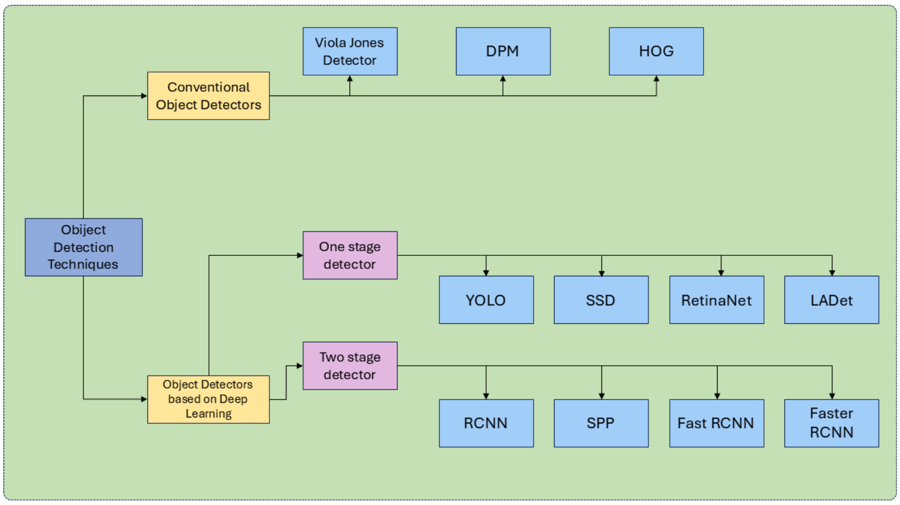


Figure 1: Traditional target detection algorithms and deep learning based target detection algorithms. Traditional target detection algorithms are Viola–Jones detector, DMP, HOG. Deep learning-based algorithms are categorized into single-stage and two-stage detectors. The representative single-stage ones are YOLO, SSD, and the representative two-stage ones are R-CNN, SPP, Fsat R-CNN, Faster R-CNN.

ture for streamlined feature fusion. This combination enriches channel interactions, reduces computational complexity, and removes redundant operations, thereby improving inference speed while maintaining high detection accuracy. The lightweight design makes the model particularly suitable for deployment on resource-constrained underwater or edge devices.

(4) We conduct extensive evaluations on both the J-EDI and Brackish Underwater datasets. YOLO-GGS achieves an mAP@0.5:0.95 of 88.5% on J-EDI, surpassing YOLOv8n by 4%. The cross-domain generalization experiment on Brackish Underwater dataset further demonstrates strong robustness to unseen environments. In addition, real-time evaluation confirms that the proposed model delivers the fastest inference (9.2 ms per frame, 108.4 FPS) with the lowest memory footprint (414 MB), highlighting its balanced trade-off among accuracy, efficiency, and practicality.

3 Improve strategy

In object detection, we reviewed several relevant studies, including the YOLO enhancements by Zhao et al.[20] and Cheng et al.[21]. Our improvement strategy leverages the GSConv module and Slim-neck structure from Gold-YOLO. Specifically, the YOLOv8 Head adopts Gold-YOLO's architectural design. The Backbone replaces standard convolutions with lightweight GSConv, enhancing feature extraction and improving model accuracy. Gold-YOLO also introduces a Gather-and-Distribute (GD) mechanism, which addresses the limitations of traditional Feature Pyramid Networks (FPN) [22] in information flow, improving detection across object scales. The GD mechanism comprises two branches: Low-Gather-and-Distribute (Low-GD) for small targets and High-Gather-and-Distribute (High-GD) for larger targets. This design effectively balances detection accuracy and speed, which is critical for real-time underwater debris detection.

3.1 YOLOv8

The YOLOv8 architecture (Figure 2) consists of four modules: Input, Backbone, Neck, and Head. The Input module uses Mosaic augmentation, while larger models also apply MixUp and CopyPaste for better training diversity. The Backbone extracts features via a cross-stage local network, with an SPPF (Fast Spatial Pyramid Pooling) module for multi-scale context.

The Neck efficiently fuses multi-resolution features. Unlike YOLOv5's PAN-FPN, YOLOv8 performs sequential down- and up-sampling without extra convolutions. The C3 modules are replaced by lightweight C2f blocks (Figure 3), enhancing adaptability to targets of varying sizes and shapes.

The decoupled Head has parallel branches for regression and classification, reducing parameters and improving robustness. YOLOv8 adopts an anchor-free design to predict target centers and aspect ratios, enhancing speed and accuracy.

The C2f block, a two-layer CSPBottleneck, boosts feature extraction efficiency. YOLOv8 also integrates the E-ELAN architecture from YOLOv7 [14] with cross-layer branching for better gradient propagation. Shortcut connections remain in the Backbone but are disabled in the Neck. The Neck forwards fused features to the decoupled Head, yielding improved overall performance.

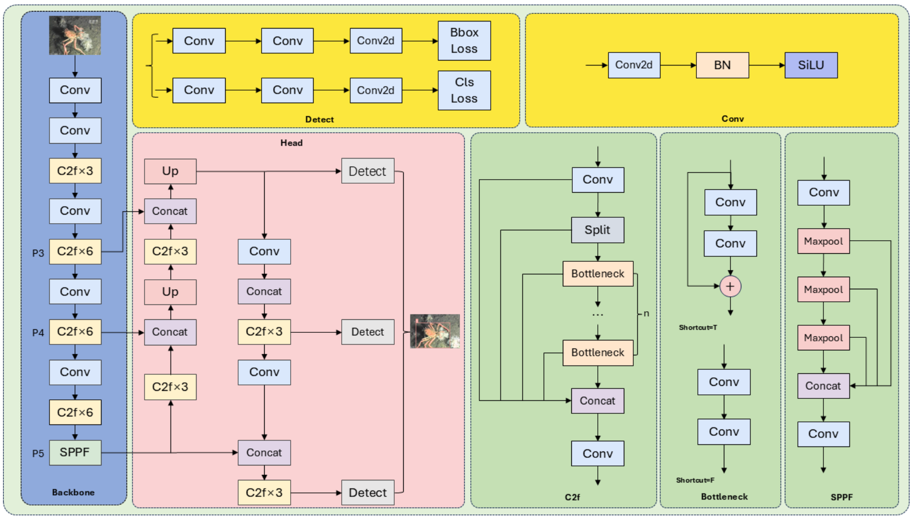


Figure 2: Improvements based on the legacy YOLOV8 network. The traditional yolov8 network structure contains backbone and C2f, Neck, head, Conv, SPPF, and Bottleneck.

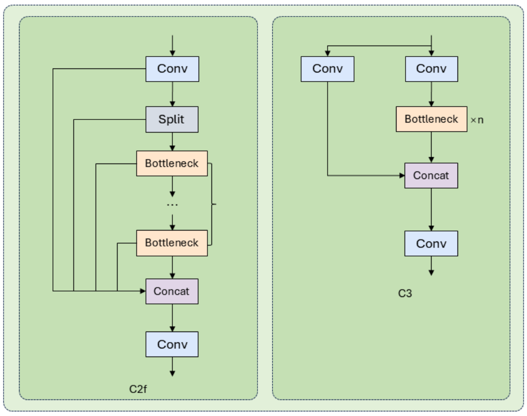


Figure 3: In YOLOv8's Backbone, the original C3 block based on CSPNet is replaced by the lighter C2f module, which improves efficiency while preserving feature extraction ability.

3.2 GSConv

To enhance the model's representational capacity while reducing computational cost and parameters for edge deployment, this study introduces GSConv, an efficient hybrid convolution module. GSConv achieves a balance between feature richness and efficiency through a dual-branch design (Figures 4).

Standard convolution produces dense feature maps but incurs high computational overhead:

$$F_{std} = K \times K \times C_{in} \times C_{out} \times H \times W \tag{1}$$

where K is the kernel size, C_{in} and C_{out} are input and output channels, and H, W are spatial dimensions.

Depthwise separable convolution (DSC) reduces computation by splitting a standard convolution into two steps: depthwise convolution (DWConv), which performs spatial convolution for each channel, and pointwise convolution (PWConv), which fuses the channels.

$$F_{dsc} = K \times K \times C_{in} \times H \times W + C_{in} \times C_{out} \times H \times W \tag{2}$$

DSC lowers computation, but limited cross-channel interaction in DWConv reduces feature richness, especially in compact networks.GSConv overcomes this by combining standard convolution's expressiveness with DSC efficiency and enabling global channel interaction via channel shuffling. As shown in Fig.4, its forward pass consists of three steps:

1. Feature Extraction with Channel Reduction: Standard convolution reduces the input tensor $X \in \mathbb{R}^{C_{in} \times H \times W}$ to $C_{out}/2$ channels:

$$F_{std} = \text{StdConv}Cout/2(X) \tag{3}$$

2. Efficient Feature Processing: In parallel, DSC produces $C_{out}/2$ complementary channels:

$$F_{dsc} = \text{DSC}Cout/2(F_{std}) \tag{4}$$

3. Feature Fusion with Channel Integration: The two outputs are concatenated and shuffled for full channel interaction:

$$F_{out} = \text{ChannelShuffle}([F_{std}, F_{dsc}]) \tag{5}$$

This dual-branch design preserves dense features in half of the channels while maintaining DSC-level efficiency in the other half. Channel shuffling ensures effective integration, achieving representational capacity close to standard convolution with computation similar to DSC. Integrating GSConv into key network components enables high performance under constrained computational resources.

3.3 Gather-and-distribute mechanism

To bridge the semantic gap between shallow and deep features while enhancing multi-scale representation, we adopt

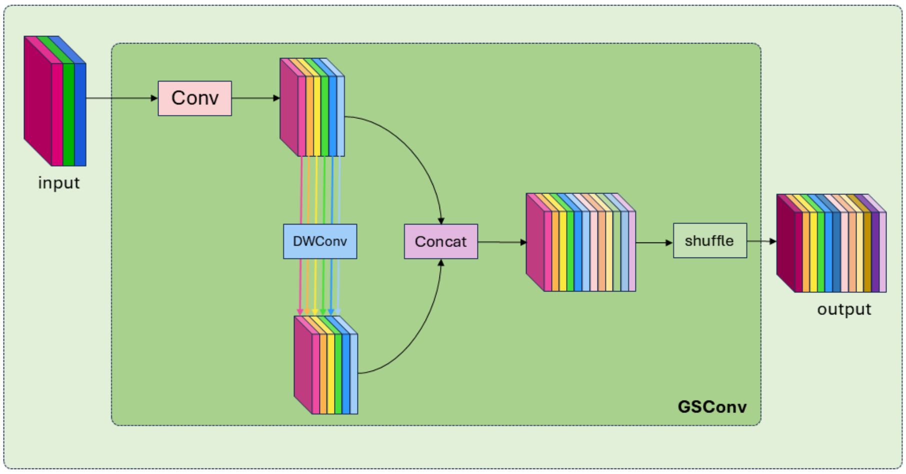


Figure 4: In the GSConv structure, a standard convolution is first applied for downsampling, followed by a depthwise convolution (DWConv). The outputs of both convolutions are then concatenated and passed through a final shuffle operation.

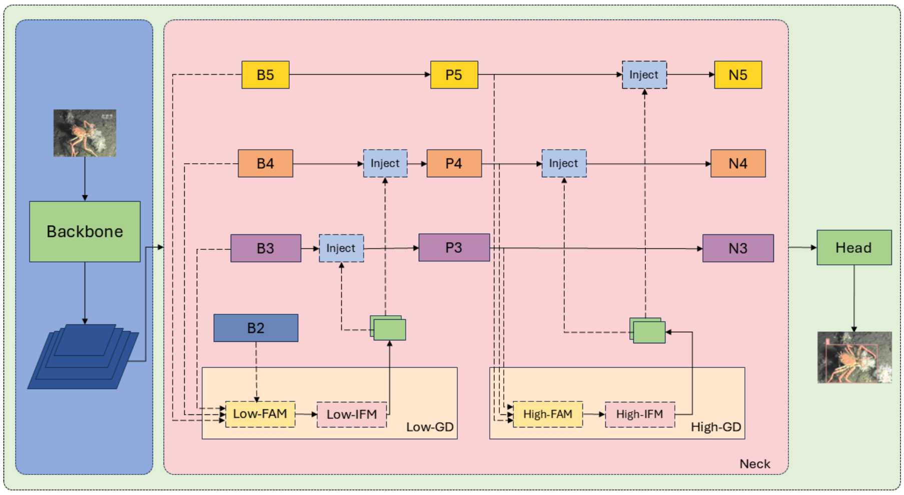


Figure 5: The GD module in Gold-YOLO introduces a Gather-and-Distribute mechanism. Information is collected and fused across layers, then redistributed to each layer. This process involves three components: the Feature Alignment Module (FAM), the Information Fusion Module (IFM), and the Information Injection Module (IIM).

the Gather-and-Distribute (GD) mechanism originally proposed in GOLD-YOLO(Figure 5). The GD mechanism is designed with three key components:

1)Feature Alignment Module (FAM): aligns feature maps from different scales into a unified resolution.

2)Information Fusion Module (IFM): integrates the aligned features to obtain global context.

3)Information Injection Module (IIM): redistributes the fused global information back to the backbone or neck layers.

The overall GD mechanism operates in two stages: a *Low-stage* and a *High-stage*, each tailored for different levels of semantic abstraction.

3.3.1 Low-stage GD

The low-stage GD focuses on enhancing fine-grained details from shallow features(Figure 6). Given backbone features $\{B2, B3, B4, B5\}$, the Low-FAM first aligns them to the spatial size of $B4$ using bilinear interpolation and average pooling:

$$F^{l}_{\text{align}} = \{A(B2), A(B3), A(B4), A(B5)\} \qquad (6)$$

where $A(\cdot)$ denotes the alignment function combining bilinear resizing and average pooling.

Next, the Low-IFM aggregates the aligned features using 1×1 convolution and RepConv blocks:

$$F^{l}_{\text{global}} = \text{RepConv}\big(\text{Conv}_{1\times1}(\text{Concat}(F^{l}_{\text{align}}))\big) \qquad (7)$$

In the Low-IIM module (Figure 8), the original feature map is first aligned in scale through average pooling and

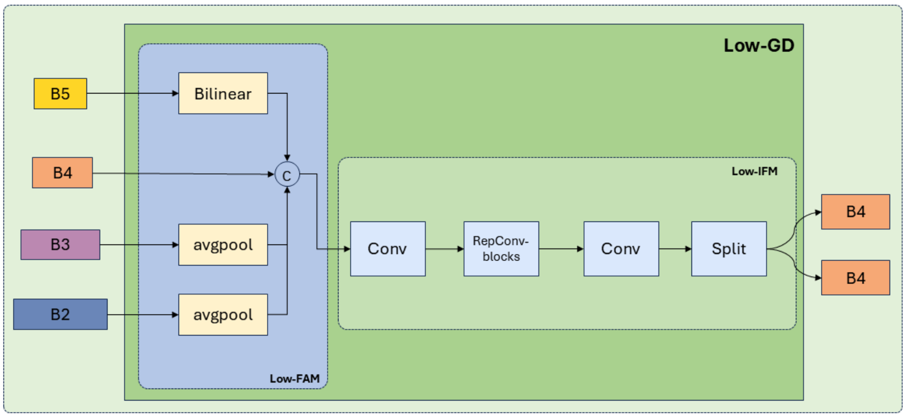


Figure 6: Low-GD fuses the shallow feature information of the model. It includes Low-FAM and Low-IFM. Low-FAM uses average pooling to obtain uniformly sized features, while Low-IFM consists of a multi-layer reparameterized convolutional block followed by a split operation.

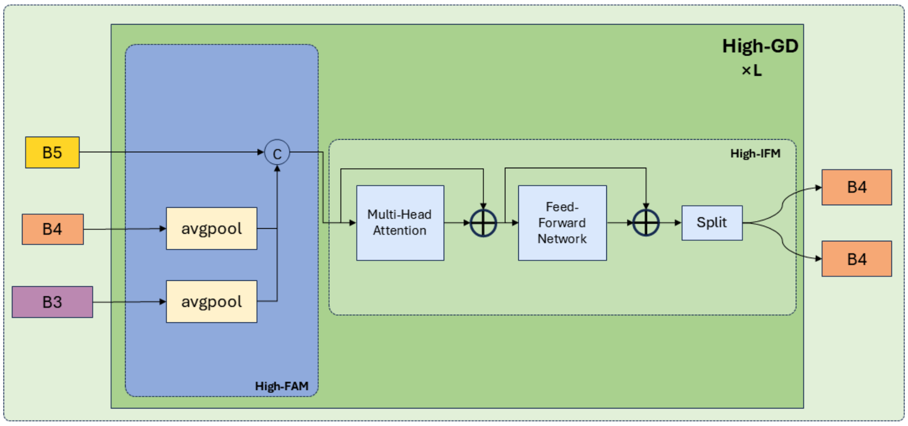


Figure 7: High-GD is similar in structure to Low-GD. It fuses features obtained from Low-GD (P3, P4, P5) and also contains High-FAM and High-IFM. High-FAM aligns feature sizes using global average pooling, while High-IFM includes multiple transformer blocks followed by a split operation.

Algorithm 1 GSConv Forward Propagation Algorithm

```
Input feature map $X \in R^{C_{in} \times H \times W}$, output channels $C_{out}$
Output feature map $F_{out} \in R^{C_{out} \times H \times W}$
Standard convolution branch
$F_{std} \leftarrow$ StdConv $Cout/2(X)$;
Depthwise separable convolution branch
$F_{dw} \leftarrow$ DWConv $Cout/2(F_{std})$;
$F_{pw} \leftarrow$ PWConv $Cout/2(F_{dw})$;
Concatenate and channel shuffle
$F_{cat} \leftarrow$ Concat$(F_{std}, F_{pw})$;
$F_{out} \leftarrow$ ChannelShuffle$(F_{cat})$;
return $F_{out}$
```

bilinear interpolation within the LAF module. It is subsequently multiplied element-wise with the global feature refined by convolution and sigmoid activation. Finally, a residual connection adds the 1×1 convolved global feature, thereby generating the fused feature map $P_{3,4}$ that effectively incorporates global contextual information.

3.3.2 High-stage GD

The high-stage GD focuses on semantic-rich high-level featuress (Figure 7). Given $\{P3, P4, P5\}$, the High-FAM aligns them to the spatial size of $P5$ using average pooling:

$$F^h_{\text{align}} = \{P(P3), P(P4), P(P5)\} \tag{8}$$

where $P(\cdot)$ denotes pooling-based alignment.

The High-IFM employs a multi-head self-attention (MHSA) module followed by a feed-forward network (FFN) to capture long-range dependencies and fuse features:

$$F^h_{\text{global}} = \text{FFN}(\text{MHSA}(F^h_{\text{align}})) \tag{9}$$

In the High-IIM module (Figure 8), the overall procedure is identical to that of the Low-stage IIM. The only difference lies in the LAF module, which employs average pooling alone to align the multi-level original features to a unified scale, without performing bilinear interpolation. This simplification preserves the capability of global context fusion while reducing computational overhead, and the result-

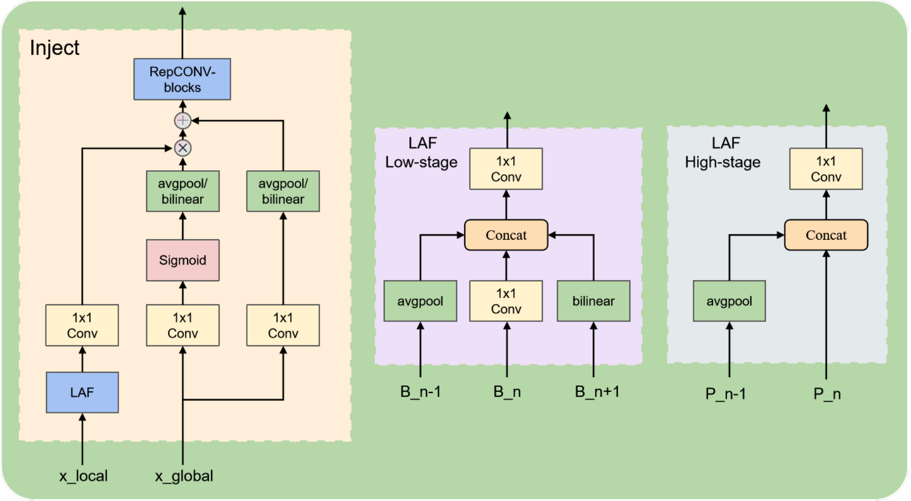


Figure 8: The structure of the information injection module.

ing fused features effectively integrate high-level semantic information.

3.3.3 Pseudocode

The overall procedure of the GD mechanism is summarized in Algorithm 2.

Algorithm 2 Gather-and-Distribute (GD) Mechanism

```
Backbone features $\{B2, B3, B4, B5\}$
Neck features $\{P3, P4, P5\}$
Enhanced features $\{B3', B4', P4', P5'\}$
**Low-stage GD:**
Align $\{B2, B3, B4, B5\}$ to $B4$ size $\rightarrow F^l_{\text{align}}$
Fuse by Conv + RepConv $\rightarrow F^l_{\text{global}}$
Inject into $B3, B4 \rightarrow B3', B4'$
**High-stage GD:**
Align $\{P3, P4, P5\}$ to $P5$ size $\rightarrow F^h_{\text{align}}$
Fuse by MHSA + FFN $\rightarrow F^h_{\text{global}}$
Inject into $P4, P5 \rightarrow P4', P5'$
**return** $\{B3', B4', P4', P5'\}$
```

In summary, the GD mechanism enables effective cross-scale information interaction. The low-stage branch enhances shallow features with fine-grained local cues, while the high-stage branch provides semantic-rich global context. Together, they significantly improve the representational capacity of the detector across varying object scales.

4 Experiments

4.1 Experimental configuration and reproducibility

All experiments were conducted under consistent hardware and software settings to ensure reproducibility. The environment consisted of an NVIDIA GeForce RTX 4070 Ti GPU (12 GB VRAM), a 12-core Intel Xeon Platinum 8255C CPU, and 32 GB RAM, running Ubuntu 22.04 (64-bit) with CUDA 12.2, Python 3.10.12, and PyTorch 2.2.2 (Table 1).

Images were resized to 640 × 640 pixels with padding to preserve aspect ratio and normalized to [0, 1]. Bounding boxes were converted to $(x_{\text{center}}, y_{\text{center}}, w, h)$ format, and labels were one-hot encoded. Data augmentation included Mosaic (0.5), MixUp ($\alpha = 0.2$), CopyPaste (for larger models), random horizontal flips (0.5), and color jittering (±0.2 in brightness, contrast, saturation), implemented using Albumentations 1.3.0.

Training used SGD with an initial learning rate (lr0) of 0.01, decay (lrf) of 0.01 under a cosine annealing schedule, momentum 0.937, and weight decay 0.0005. A 3-epoch warm-up gradually increased the learning rate. The batch size was set to -1 to allow automatic adjustment based on GPU memory. Models were trained for 300 epochs with 640 × 640 inputs, balancing accuracy and efficiency. Losses included IoU-based box regression, objectness, and classification. Key hyperparameters and training configurations are summarized in Table 2.

Table 1: Configuration of experimental equipment and experimental environment requirements.

Name	**Configuration**
CPU	12-core Intel(R) Xeon(R) Platinum 8255C
GPU	NVIDIA GeForce RTX 4070 Ti
Memory	12 GB
Environment	Ubuntu 22.04
CUDA	12.2
Python	3.10.12
PyTorch	2.2.2

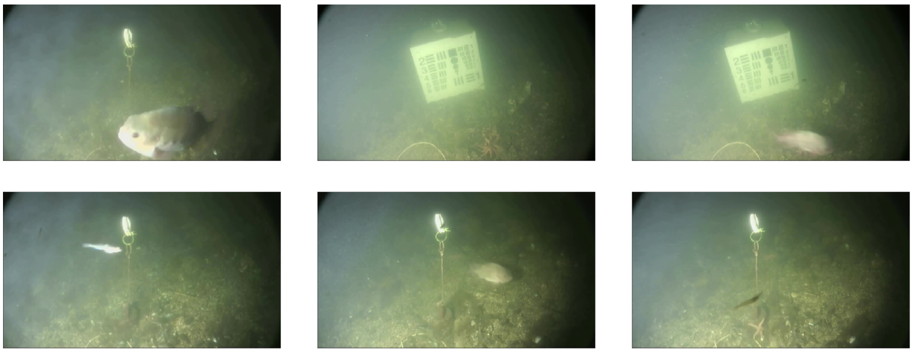


Figure 9: Representative samples from the Brackish Underwater Dataset

Table 2: Important hyperparameters and configurations for YOLOv8 training.

Hyperparameter	Value	Description
learning rate (lr0)	0.01	Controls the step size for weight updates during training
learning decay rate (lrf)	0.01	Regulates the decay of learning rate over epochs
momentum	0.937	Stabilizes training by smoothing gradient updates
weight_decay	0.0005	Prevents overfitting by penalizing large weights
warmup_epochs	3.0	Gradually increases initial learning rate to prevent instability
batch	-1	Auto-adjusts batch size based on available GPU memory
imgsz	640	Input image size in pixels for model training

4.2 Data set

This study utilizes two complementary underwater datasets. The first, from the J-EDI dataset [23], contains 5,720 images of diverse marine debris captured in authentic underwater environments, with bounding-box annotations for debris, biological entities, and remotely operated vehicles (ROVs). The second, the Brackish Underwater Dataset [24], comprises 14,674 real European underwater images annotated for fish, crabs, and small objects, covering a wide range of sizes and densities. Together, these datasets provide a comprehensive benchmark for underwater object detection under realistic conditions. Figures 9 and 10 show representative examples from the Brackish Underwater Dataset and the J-EDI dataset, respectively.

4.3 Model evaluation indicators

To evaluate model performance, we adopt standard object detection metrics, including Precision, Recall, Average Precision (AP), and mean Average Precision (mAP). Their mathematical definitions are given in Eqs. (10)–(12).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{11}$$

$$AP = \sum_{n} (\text{Recall}_{n+1} - \text{Recall}_{n}) \cdot \text{Precision}_{n} \tag{12}$$

True Positive (TP) denotes correctly predicted positive samples that match the ground truth. False Negative (FN) occurs when positive samples are missed, and False Positive (FP) refers to incorrect positive predictions. Higher Precision indicates fewer false alarms, while higher Recall reflects fewer missed detections.

Detection performance is evaluated using mAP at different IoU thresholds: mAP@0.5 and mAP@0.5:0.95. The former measures average precision at a single threshold of 0.5, while the latter averages AP over ten thresholds from 0.5 to 0.95 in steps of 0.05, providing a comprehensive assessment of detection capability (Eq. 13).

$$mAP@0.5:0.95 = \frac{\sum i = 0.5^{0.95} AP_{\text{IoU}=i}}{n} \tag{13}$$

The accuracy of predicted bounding boxes is measured by Intersection over Union (IoU), a key metric in object detection. IoU quantifies the overlap between a predicted box and the ground truth, reflecting localization precision during training and evaluation (Figures 11). It is defined as:

$$\text{IoU}(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{14}$$

As shown in Figure 10, IoU is the ratio of the overlapping area to the combined area of two boxes. Higher IoU indicates greater overlap, while values near 0 indicate minimal overlap. An IoU of 1 represents perfect alignment with the ground truth.

4.4 Ablation experiment

To evaluate the effectiveness of the proposed improvements, ablation studies were conducted using the same

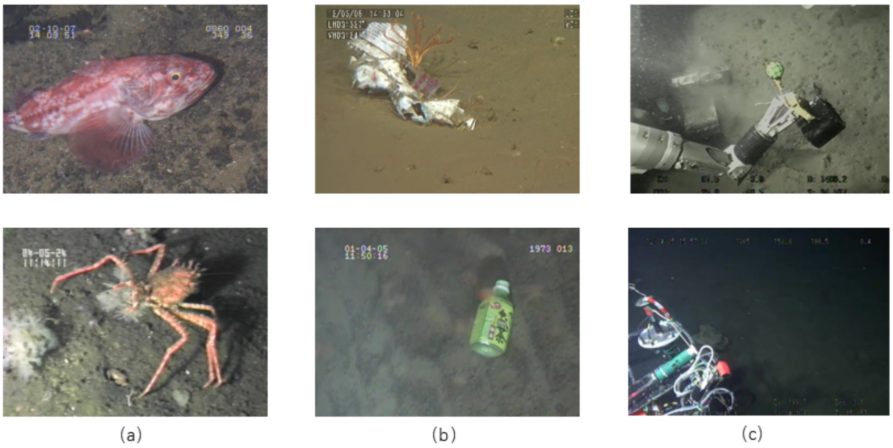


Figure 10: Examples of all categories in the dataset; (a) underwater organism; (b) underwater trash ; (c) rov.

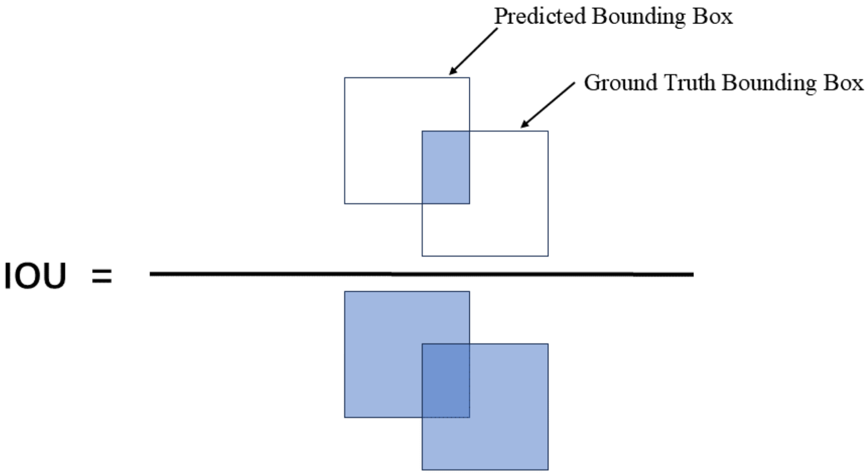


Figure 11: IOU is a measure of the effectiveness of the model by calculating the ratio of the area of intersection of the predicted and real frames to the area of concatenation, and a higher IOU means that the accuracy of the model's prediction is also higher.

training strategy and hyperparameters. As shown in Table 2, incorporating only the GSConv module into the backbone reduces FLOPs with a slight performance drop. Modifying only the neck to adopt Gold-YOLO's structure significantly improves detection accuracy, though it increases computation. Combining both Gold-YOLO's neck and GSConv modules achieves the highest mAP@0.5:0.95, demonstrating the effectiveness of the proposed approach.

Table 3: The results of the ablation experiments. A '✓' indicates the use of the corresponding module.

YOLOv8	GD	GSConv	Input	mAP@0.5:0.95
✓			640×640	84.5%
✓	✓		640×640	85.6%
✓		✓	640×640	84.4%
✓	✓	✓	640×640	88.5%

4.5 Contrast experiment

The experimental performance of our method was evaluated and compared with several state-of-the-art lightweight target detection algorithms on both the J-EDI and Brackish Underwater datasets. As shown in Table 4 (J-EDI dataset), our method achieves a mAP@0.5:0.95 of 88.5%, outperforming all compared methods, including a 5.1% improvement over YOLOv5n (2021). Table 5 presents the results on the Brackish Underwater dataset, where our approach consistently achieves superior performance across all evaluated metrics. Integrating the GSConv module into YOLOv8n reduces model size but leads to a slight decrease in mAP@0.5:0.95, highlighting the trade-off between efficiency and detection accuracy. Overall, these results indicate that our method maintains a favorable balance between accuracy and computational cost across different underwater scenarios.

Although YOLOv8n and Slim-Neck+v8 show advantages in computational efficiency and parameter reduction, their detection accuracies remain relatively low across both datasets. On the J-EDI dataset, they achieve 84.5% and 84.7%, respectively, while on the Brackish Underwater dataset, their accuracies are 82.2% and 82.3%. In contrast, our proposed method attains higher accuracies of 88.5% on J-EDI and 84.7% on Brackish Underwater, with only a slight increase in computational complexity. Compared to the original Slim-Neck method, our approach improves mAP@0.5:0.95 by 3.8% and 2.4%, respectively, while also reducing model size. Notably, our model is 38% smaller than YOLOv8s. Experiments incorporating the SEAttention mechanism or integrating DSConv [25] did not yield significant improvements, indicating that the performance gains mainly stem from our architectural modifications. Overall, comprehensive comparisons demonstrate that our method provides a more favorable trade-off between detection accuracy and computational efficiency, especially in underwater scenarios with complex visual characteristics.

To assess the robustness and generalization of the proposed YOLO-GGS model, we compared it with the YOLOv5 baseline, monitoring performance every 50 training epochs. As summarized in Table 6, YOLO-GGS demonstrates superior stability and consistently outperforms YOLOv5 throughout training.

The trained YOLO-GGS model weights were used to detect images from various underwater scenes. The results, shown in Figure 14, indicate enhanced robustness and higher confidence levels. The model maintains strong recognition ability even in complex or occluded environments, effectively meeting the demands of diverse underwater scenarios.

Figure 15 presents experimental results on the J-EDI dataset, comparing our model with mainstream frameworks, including YOLOv3-tiny, YOLOv5, and YOLOv8. Our model achieves substantially higher detection accuracy, with the highest mAP@0.5:0.95 among all compared methods. YOLOv3-tiny shows signs of overfitting around epoch 272, likely due to the mismatch between its model complexity and the dataset characteristics. Despite this, its overall training performance remains competitive. These results demonstrate that our approach improves detection accuracy while reducing model parameters, highlighting its effectiveness and efficiency.

We also evaluated Mamba-YOLO on the underwater litter dataset. Despite its advanced architecture, multiple training runs revealed a tendency for overfitting. Specifically, validation losses significantly exceeded training losses after several epochs. As a result, its final detection accuracy was lower than that of our proposed method. These observations suggest that Mamba-YOLO may require additional task-specific adjustments or regularization strategies, which we plan to investigate in future work.

Furthermore, we conducted a comprehensive comparison of YOLO-GGS with representative YOLO architectures, including YOLOv3 variants (YOLOv3, YOLOv3-SPP, YOLOv3-Tiny), YOLOv5 variants (n, s, m, l, x), and YOLOv8 variants (n, s, m, l, x). Detailed results are presented in Table 7, highlighting the effectiveness of our proposed model across diverse architectures and scales.

Table 4: Performance comparison of different object detection models on the J-EDI dataset.

Models	Input	Batch	Epoch	mAP@0.5:0.95
YOLOv3	640×640	-1	300	85.5%
YOLOv5n	640×640	-1	300	83.4%
YOLOv5s	640×640	-1	300	84.1%
YOLOv6n	640×640	-1	300	83.9%
DSConv+v5	640×640	-1	300	83.2%
YOLOv8n	640×640	-1	300	84.5%
Slim-Neck+v8	640×640	-1	300	84.7%
DSConv+v8	640×640	-1	300	84.4%
SEattention+v8	640×640	-1	300	83.9%
GD+v8	640×640	-1	300	85.6%
Mamba-YOLO	640×640	-1	300	85.2%
Ours	**640×640**	**-1**	**300**	**88.5%**

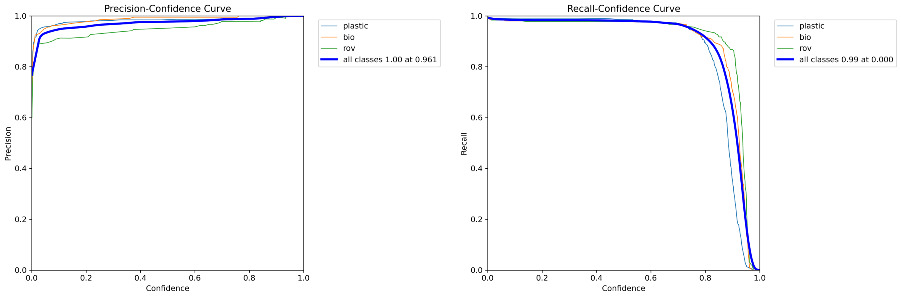
Table 5: Evaluation results of various object detection models on the Brackish Underwater Dataset.

Models	Input	Batch	Epoch	mAP@0.5:0.95
YOLOv3	640×640	-1	300	72.49%
YOLOv5	640×640	-1	300	81.2%
YOLOv5s	640×640	-1	300	81.7%
YOLOv6	640×640	-1	300	81.4%
YOLOv7	640×640	-1	300	69.5%
YOLOv8n	640×640	-1	300	82.2%
Slim-Neck+v8	640×640	-1	300	82.3%
DSConv+v8	640×640	-1	300	81.9%
SEattention+v8	640×640	-1	300	81.5%
GD+v8	640×640	-1	300	82.8%
Mamba-YOLO	640×640	-1	300	83.3%
Ours	**640×640**	**-1**	**300**	**84.7%**

4.6 Real-time performance evaluation

To further analyze real-time performance, we compare not only the average inference time and memory footprint but also the trade-off between computational efficiency and deployment feasibility. As shown in Table 8, the proposed method achieves the lowest memory consumption (414 MB) and fastest inference (9.2 ms, 108.4 FPS), significantly exceeding the 30 FPS threshold required for real-time applications. Compared with YOLOv8n, it reduces memory usage by 13% and delivers 17% faster inference, demonstrating a superior efficiency-to-accuracy ratio. All experiments were conducted on a single NVIDIA RTX 4070Ti GPU with an input image size of 640.

The lightweight design of the model ensures stable inference latency and enables deployment on edge devices with limited GPU memory. It supports multi-camera parallel inference and real-time decision-making in robotics



(a) (b)

Figure 12: (a) shows the relationship between Precision and Confidence. The horizontal axis represents Confidence, with higher values indicating greater certainty in the detection results. The vertical axis represents Precision, defined as the proportion of correctly detected spam among all items detected as garbage. (b) shows the relationship between Recall and Confidence. The horizontal axis again represents Confidence. The vertical axis represents Recall, defined as the proportion of detected spam among all actual garbage items. As confidence increases, precision generally improves, while recall may vary depending on the threshold, reflecting the model's detection effectiveness.

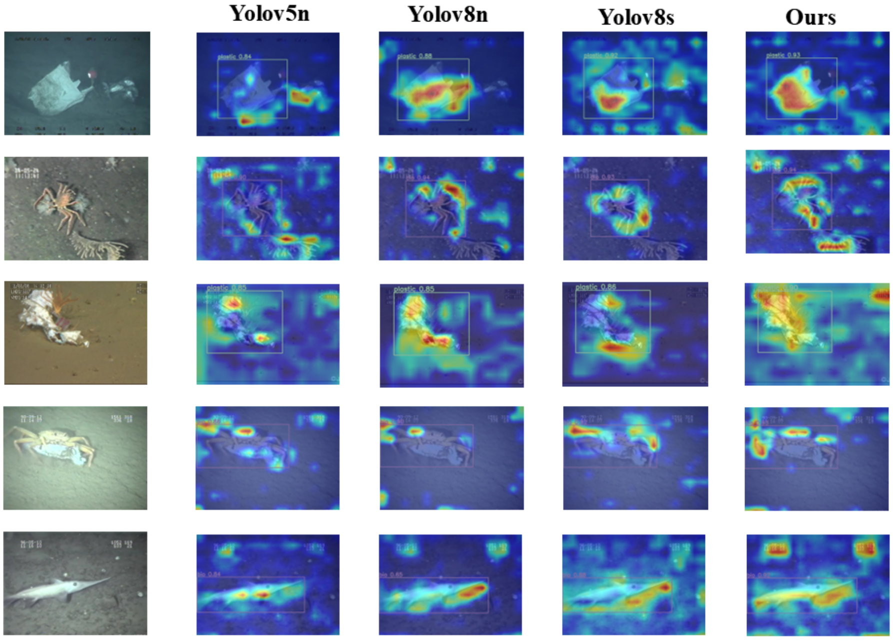


Figure 13: We used Grad-CAM (Gradient Weighted Class Activation Map) method to compare the improved model with the heat maps generated by the mainstream YOLOv5, YOLOv8n and YOLOv8s models. In the comparison map, it can be found that the attention of our improved model is more focused and more sensitive to the detection target.

and underwater detection systems. Furthermore, its compact architecture provides flexibility for integration into mobile and embedded platforms, reducing deployment cost and enhancing applicability in resource-constrained scenarios. These advantages highlight the method's potential for broader real-world applications requiring both high speed and efficiency.

4.7 Overall experimental analysis

To conduct a comprehensive evaluation, we compared our model with several mainstream detectors under identical configurations. Results show that our method achieves higher accuracy with a more compact size. Grad-CAM [26] visualizations in Figure 13 show clearer and more focused attention regions than YOLOv5s, YOLOv8n, and YOLOv8s. This indicates stronger feature localization for underwater litter detection.

Figure 12 presents precision–confidence and recall–confidence curves. Higher precision at elevated confidence levels indicates reliable predictions, while recall declines as low-confidence true positives are excluded. Category-specific differences are observed, with the ROV class showing slightly lower precision due to more false positives. Overall, precision approaches 1.0 and recall nearly 0.99 at high thresholds.

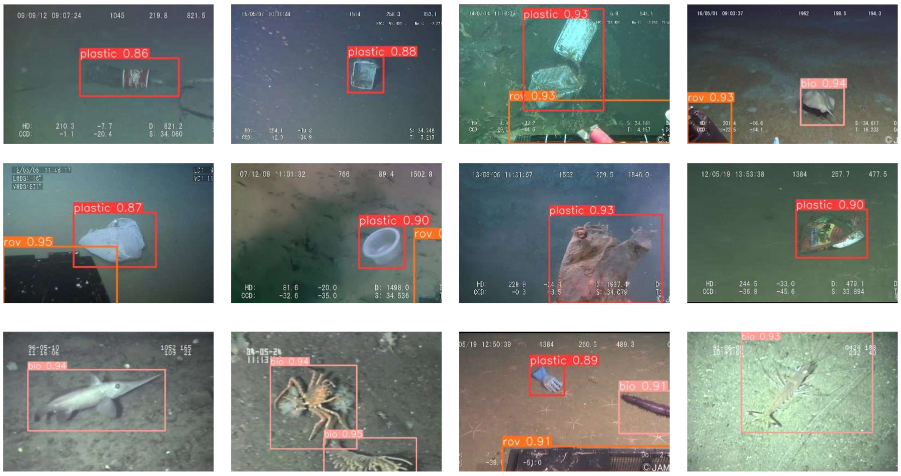


Figure 14: Plot of results of detection in different underwater scenarios. The trained model is used in different detection scenarios and different detection objects for detection and the results are shown in Fig. The results show that our model has strong generalization ability in underwater detection sites.

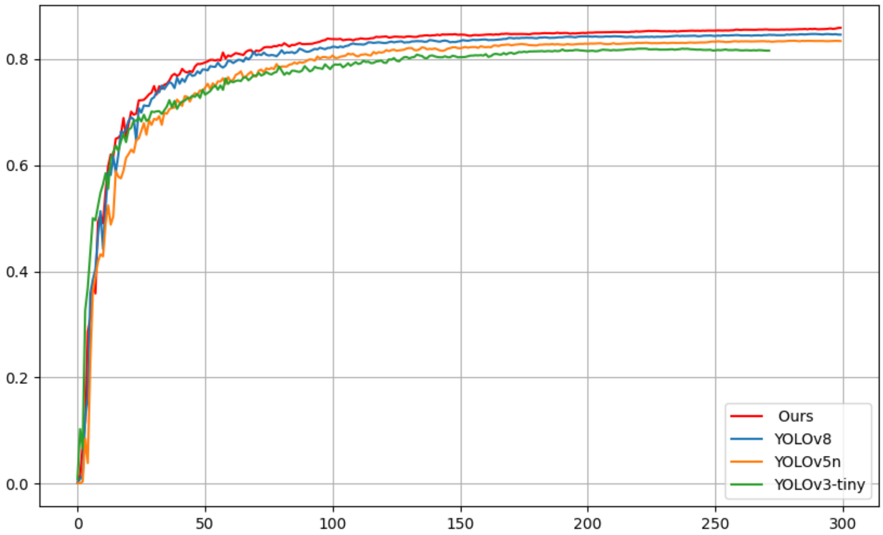


Figure 15: Our model is compared with yolov3-tiny, yolov5, and yolov8 for the experimental procedure mAP@0.5:0.95 values. The figure shows that our model always maintains a more stable effect during the overall process of training, with the highest training accuracy.

During training, most models used a batch size of 16, while YOLOv3-SPP and YOLOv8x used 8 due to GPU memory limits. Despite achieving similar mAP@0.5:0.95 values of 85.8%, these models contain 15× more parameters, resulting in slower inference and higher computational cost. In contrast, our model maintains high accuracy with fewer parameters, striking a better balance between precision and complexity.

For real-time evaluation, single-frame inference time and GPU memory usage were measured. Our method achieves the fastest inference (9.2 ms, 108.4 FPS) and lowest memory footprint (414 MB), surpassing YOLOv3, YOLOv5n, YOLOv6n, YOLOv7, YOLOv8n, and Mamba-YOLO, demonstrating its suitability for resource-constrained platforms.

Nonetheless, limitations remain. By comparing the model's performance improvements and detection performance on the two datasets, we found that the model's performance may degrade in complex underwater scenes with severe occlusion or poor lighting, while fine-grained categories such as small ROVs still show high false positives. Further optimization is needed to ensure stable real-time operation on embedded devices.

5 Conclusion

In this study, we propose YOLO-GSS, an improved YOLOv8-based model integrating the Gather-and-Distribute (GD) mechanism, GSConv, and a Slim-Neck

Table 6: Comparison of YOLOv5n and our model across different training epochs on the J-EDI dataset, showing that the proposed method consistently achieves higher stability and accuracy in terms of classification loss and overall performance.

Models	Input	Epoch	mAP@0.5:0.95	cls_loss
YOLOv5n	640×640	50	73.9%	0.547
	640×640	100	80.2%	0.417
	640×640	150	82.0%	0.374
	640×640	200	82.9%	0.366
	640×640	250	83.2%	0.359
	640×640	300	83.4%	0.356
Ours	640×640	50	80.1%	0.442
	640×640	100	84.7%	0.342
	640×640	150	86.5%	0.323
	640×640	200	87.8%	0.316
	640×640	250	88.3%	0.313
	640×640	300	**88.5%**	**0.309**

Table 7: Comparison of detection accuracy on the J-EDI dataset among YOLOv3, YOLOv5, and YOLOv8.

Models	Input	Batch	Epoch	mAP@0.5:0.95
YOLOv3	640×640	-1	300	85.5%
YOLOv3-spp	640×640	8	300	85.3%
YOLOv3-tiny	640×640	-1	300	82.1%
YOLOv5n	640×640	-1	300	83.4%
YOLOv5s	640×640	-1	300	85.3%
YOLOv5m	640×640	-1	300	82.1%
YOLOv5l	640×640	-1	300	85.5%
YOLOv5x	640×640	-1	300	85.3%
YOLOv8n	640×640	-1	300	84.5%
YOLOv8s	640×640	-1	300	84.9%
YOLOv8m	640×640	-1	300	85.4%
YOLOv8l	640×640	-1	300	85.5%
YOLOv8x	640×640	8	300	85.8%
Ours	**640×640**	**-1**	**300**	**88.5%**

structure. The model achieves a balance between high detection accuracy and low computational cost for real-time underwater litter detection. Experimental results on the J-EDI and Brackish datasets demonstrate mAP@0.5:0.95 values of 88.5% and 84.7%, respectively, outperforming mainstream YOLO variants, while Grad-CAM visualizations confirm strong feature localization.

Single-frame inference analysis shows that YOLO-GSS processes each frame in 9.2 ms (108.4 FPS) using only 414 MB of memory, highlighting its suitability for deployment on resource-constrained AUV and ROV platforms. Despite these advantages, performance may degrade for small targets under extreme illumination, severe occlusion, or cluttered scenes, indicating that further optimization is needed for consistent real-time operation.

Future work will focus on enhancing multi-scale feature fusion for small and occluded objects, exploring efficient attention mechanisms, and developing dynamic inference strategies to improve generalization. Additionally, we plan to deploy the model in real underwater embedded systems to facilitate practical applications in marine environmental protection and intelligent monitoring.

Table 8: Memory and inference speed comparison of YOLO-based models on single-frame input.

Models	Memory usage (Mb)	Inference time (ms)	FPS (frames/s)
YOLOv3	1355	33.3	30.3
YOLOv5n	455	12.7	78.8
YOLOv6n	440	10.6	94.5
YOLOv7	462	11.35	88.1
YOLOv8n	477	10.8	92.6
Mamba-YOLO	521	12.3	81.3
Ours	**414**	**9.2**	**108.4**

References

[1] Dimitris V Politikos et al. “Automatic detection of seafloor marine litter using towed camera images and deep learning”. In: *Marine Pollution Bulletin* 164 (2021), p. 111974. DOI: https://doi.org/10.1016/j.marpolbul.2021.111974.

[2] Vishal Verma et al. “A deep learning-based intelligent garbage detection system using an unmanned aerial vehicle”. In: *Symmetry* 14.5 (2022), p. 960. DOI: https://doi.org/10.3390/sym14050960.

[3] Dongliang Ma et al. “MLDet: Towards efficient and accurate deep learning method for Marine Litter Detection”. In: *Ocean & Coastal Management* 243 (2023), p. 106765. DOI: https://doi.org/10.1016/j.ocecoaman.2023.106765.

[4] Xiaowen Teng et al. “The Object Detection of Underwater Garbage with an Improved YOLOv5 Algorithm”. In: *Proceedings of the 2022 International Conference on Pattern Recognition and Intelligent Systems*. 2022, pp. 55–60. DOI: https://doi.org/10.1145/3549179.3549189.

[5] Faiza Rehman et al. “Optimized YOLOV8: An efficient underwater litter detection using deep learning”. In: *Ain Shams Engineering Journal* 16.1 (2025), p. 103227. DOI: https://doi.org/10.1016/j.asej.2024.103227.

[6] Lifu Wei et al. “Image semantic segmentation of underwater garbage with modified U-Net architecture model”. In: *Sensors* 22.17 (2022), p. 6546. DOI: https://doi.org/10.3390/s22176546.

[7] Mupparaju Sohan et al. “A review on yolov8 and its advancements”. In: *International Conference on Data Intelligence and Cognitive Informatics*. Springer. 2024, pp. 529–545. DOI: https://doi.org/10.1007/978-981-99-7962-2_39.

[8] Chengcheng Wang et al. "Gold-YOLO: Efficient object detector via gather-and-distribute mechanism". In: *Advances in Neural Information Processing Systems* 36 (2024). DOI: https://doi.org/10.48550/arXiv.2309.11331.

[9] Hulin Li et al. "Slim-neck by GSConv: a lightweight-design for real-time detector architectures". In: *Journal of Real-Time Image Processing* 21.3 (2024), p. 62. DOI: https://doi.org/10.1007/s11554-024-01436-6.

[10] Ping-I Lin et al. "Investigating sources of marine litter and developing coping strategies in scuba diving spots in Taiwan". In: *Sustainability* 14.9 (2022), p. 5726. DOI: https://doi.org/10.3390/su14095726.

[11] Wei Zhou et al. "YOLOTrashCan: a deep learning marine debris detection network". In: *IEEE Transactions on Instrumentation and Measurement* 72 (2022), pp. 1–12. DOI: https://doi.org/10.1109/TIM.2022.3225044.

[12] Yan Zhai. "River ship monitoring based on improved deep-sort algorithm". In: *Informatica* 48.9 (2024). DOI: https://doi.org/10.31449/inf.v48i9.5886.

[13] Xiaolong Qi. "Event-triggered predictive control algorithm for multi-auv formation modeling". In: *Informatica* 48.9 (2024). DOI: https://doi.org/10.31449/inf.v48i9.5890.

[14] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, pp. 7464–7475. DOI: arXiv:2207.02696.

[15] Jin Zhu et al. "YOLOv8-C2f-Faster-EMA: an improved underwater trash detection model based on YOLOv8". In: *Sensors* 24.8 (2024), p. 2483. DOI: https://doi.org/10.3390/s24082483.

[16] Pratima Sarkar, Sourav De, and Sandeep Gurung. "U-YOLOv3: A Model Focused on Underwater Object Detection". In: *Informatica* 49.6 (2025). DOI: https://doi.org/10.31449/inf.v49i6.6642.

[17] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (2018). DOI: https://doi.org/10.48550/arXiv.1804.02767.

[18] Michael S Fulton, Jungseok Hong, and Junaed Sattar. "Trash-icra19: A bounding box labeled dataset of underwater trash". In: (2020). DOI: https://doi.org/10.13020/x0qn-y082.

[19] Fan Zhao et al. "Seafloor debris detection using underwater images and deep learning-driven image restoration: A case study from Koh Tao, Thailand". In: *Marine Pollution Bulletin* 214 (2025), p. 117710. DOI: https://doi.org/10.1016/j.marpolbul.2025.117710.

[20] Xuemeng Zhao and Yinglei Song. "Improved ship detection with YOLOv8 enhanced with MobileViT and GSConv". In: *Electronics* 12.22 (2023), p. 4666. DOI: https://doi.org/10.3390/electronics12224666.

[21] Tianheng Cheng et al. "Yolo-world: Real-time open-vocabulary object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024, pp. 16901–16911. DOI: https://doi.org/10.1109/CVPR52733.2024.01599.

[22] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125. DOI: https://doi.org/10.1109/CVPR.2017.106.

[23] Michael Fulton et al. "Robotic detection of marine litter using deep visual detection models". In: *2019 international conference on robotics and automation (ICRA)*. IEEE. 2019, pp. 5752–5758. DOI: https://doi.org/10.1109/ICRA.2019.8793975.

[24] Malte Pedersen et al. "Detection of Marine Animals in a New Underwater Dataset with Varying Visibility". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.

[25] Marcelo Gennari do Nascimento, Roger Fawcett, and Victor Adrian Prisacariu. "Dsconv: Efficient convolution operator". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 5148–5157. DOI: https://doi.org/10.1109/ICCV.2019.00525.

[26] Ramprasaath R Selvaraju et al. "Grad-CAM: visual explanations from deep networks via gradient-based localization". In: *International journal of computer vision* 128 (2020), pp. 336–359. DOI: https://doi.org/10.1007/s11263-019-01228-7.

Dynamic Heterogeneous Graph Neural Network with Carbon-Sensitive Dual Attention for Lifecycle Carbon Footprint Assessment of Engineering Projects

Jiyao Jia[1], Xianjun Wu[2], Qunming Liu[2], Jianming Xu[3], Liangjiajing Deng[4, *], Shi Cheng[5]
[1]Hubei Changjiang Road & Bridge Co., Ltd, Wuhan,430100, China
[2]Hubei Communications Investment Construction Group Co., Ltd, Wuhan, 430100, China
[3]Department of Transport of Hubei Province, Wuhan, 430030, China
[4]School of Economics and Management, China University of Geosciences, Wuhan, 430074, China
[5]Collaborative Innovation Center for Emissions Trading System Co-constructed by the Province and Ministry, Wuhan, 430205, China

Keywords: Engineering carbon footprint, dynamic heterogeneous GNN, carbon-sensitive attention, third-order message passing, lifecycle assessment

Received: July 18, 2025

Global engineering projects are major carbon emitters with high heterogeneity, but traditional assessment methods (e.g., LCA, IPCC) lack precision, efficiency, and adaptability to dynamic construction. This study proposes a Carbon Footprint-aware Graph Neural Network (CF-GNN) for lifecycle carbon assessment. Its core innovations include: (1) a dynamic heterogeneous graph (entity/attribute nodes) updated via 15-day cycles and milestone triggers; (2) a carbon-sensitive dual attention mechanism prioritizing high-emission nodes/edges; (3) a third-order message passing framework capturing multi-hop carbon flows (up to 5 nodes). Validated on 3.86 million time-series data from 16 projects (residential, bridge, factory, etc.) against 8 baselines (LCA, GAT, TGAT, etc.), results show: CF-GNN achieves an average MAPE of 7.2% (38.9% lower than GAT, 55.8% lower than LCA), with bridge project RMSE at 218 tCO_2 (59.4% lower than LCA). It has 2.0±0.1s inference latency for 1000 nodes and 52±3.1min end-to-end assessment—3375-fold less manual effort than LCA (6 months/bridge). Key node identification matches experts (0.87 Kendall coefficient), with CV<5% (high stability) and 94.2±1.5% coverage for 95% prediction intervals. CF-GNN enables precise, efficient dynamic assessment, supporting low-carbon design/optimization and advancing "dual carbon" goals in construction, transportation, and energy.

Povzetek: Študija predlaga CF-GNN za dinamično ocenjevanje ogljičnega odtisa v velikih in heterogenih projektih, ki z dinamičnim heterogenim grafom zajame večstopenjske ogljične tokove ter podpira natančno, učinkovito vrednotenje.

1 Introduction

The intensification of global warming and the increasing frequency of extreme weather events have led to an international consensus on the need for controlling carbon emissions. Under the framework of the Paris Agreement, 137 countries have updated their emission reduction targets, and 68 countries have clearly stated that carbon emissions will peak before 2030. As the primary carrier of carbon emissions (accounting for 57% of the global total), accurate assessment of the carbon footprint of engineering projects has become crucial to achieving a low-carbon transformation. In the construction field, in 2023, global carbon emissions from building materials production and building operations will reach 27.8 billion tons, with China accounting for more than 50%; in transportation projects, carbon emissions per kilometer of highway construction are about 850 tons, and the emission intensity of urban road operations is 2.3 times that of highways; in energy projects, the emission coefficient of thermal power throughout the life cycle reaches 820gCO_2/kWh, and there are still 35gCO_2/kWh emissions in the manufacturing and construction of wind power. These data highlight the significant heterogeneity of carbon emissions in engineering projects, underscoring the need for refined assessment technology support. Current carbon footprint assessment methods have obvious limitations [1]. The traditional life cycle assessment (LCA) method has high data collection costs and a difficult boundary definition. The assessment of a bridge project takes 6 months. The input-output analysis (IOA) method has a rough department classification and an error of more than 30%. Among machine learning methods, random forests and XGBoost are challenging to handle the network association of "materials - equipment - process", and achieving an assessment accuracy of over 75% is difficult. However, existing graph neural networks (such as GCN and GAT) can model associations; they lack designs for the dynamic nature of engineering time series, and the error can reach 29% in the process adjustment scenario during the construction phase [2]. These defects render the existing methods incapable of meeting the needs for a dynamic and accurate assessment of the entire life cycle of engineering projects.

Graph neural networks (GNNs) provide a novel approach to addressing the problems above. Its successful application in fields such as supply chain carbon traceability (MAE reduced by 42%) and urban energy network monitoring (accuracy of 92%) has verified the advantages of processing complex associated data [3]. Beyond the commonly used vanilla GCN and GAT, modern heterogeneous and temporal graph neural network methods (e.g., HAN, R - GCN, HGT, TGAT, TGN, GraphSAGE with time encodings, and transformer - based temporal GNNs) have shown promising results in handling dynamic and heterogeneous data. However, these methods still lack targeted designs for the unique characteristics of engineering project carbon footprint assessment, such as the strong temporal correlation of construction processes and the high heterogeneity of carbon emission sources. This study proposes a carbon footprint - aware graph neural network (CF - GNN) model. The innovations include constructing a dynamic, heterogeneous graph containing entity and attribute nodes to capture temporal changes during the construction stage; designing a dual attention mechanism that integrates carbon sensitivity to strengthen the identification of key nodes; and developing a three - order message passing framework to achieve multi - scale carbon flow association modeling [4].

The experiment was verified by 16 actual engineering project data (including 3.86 million time series records). To ensure no data leakage, a leave - one - project - out evaluation method was adopted. The average MAPE of CF - GNN across three types of projects — residential, bridge, and factory buildings — reached 7.2%, which was 38.9% lower than that of GAT and also outperformed other modern temporal - heterogeneous GNN baselines (e.g., TGAT showed an average MAPE of 9.5%, TGN of 10.2%). Regarding runtime, on the unified hardware and software stack (Intel Xeon Gold 6348 CPU, NVIDIA A100 GPU, Ubuntu 20.04 LTS, PyTorch 2.0.1, DGL 1.1.2), the training time for 1000 nodes was 48 ± 2.3 minutes (mean ± SD over 5 runs), the inference time for 500 nodes was 0.8 ± 0.05 seconds, and the end - to - end assessment time (including BIM parsing, graph construction, and inference) for 1000 nodes was 52 ± 3.1 minutes. Compared with the manual LCA workflow (which takes about 6 months for a bridge project), the end - to - end efficiency of CF - GNN is significantly improved, but it should be noted that LCA is a methodological framework rather than a trainable predictor, and the runtime comparison is for reference only to show the practical application efficiency of the model. The consistency between key node identification and expert annotation reached 0.87, with a precision of 0.85 and a recall of 0.83 for top - 10% key nodes. This model provides a high - precision and efficient carbon assessment tool for low - carbon design, construction optimization, and operation management of engineering projects, helping to achieve the "dual carbon" goal.

2 Design of a graph neural network algorithm for carbon footprint assessment

2.1 Construction of the carbon footprint graph structure of engineering projects

2.1.1 Node definition

Construct a heterogeneous network containing entity nodes and attribute nodes. Entity nodes encompass four categories: material suppliers (such as steel and cement), construction equipment (including cranes and mixers), construction processes (such as foundation excavation and main body pouring), and transportation links (including material transportation and equipment transfer). Each type of node contains both quantitative features (such as supplier annual supply and equipment power) and categorical features (such as material type and equipment model). Attribute nodes include carbon emission factor library (λ_m represents the carbon emission factor of material m, unit $kgCO_2/kg$), energy conversion coefficient (η_e represents the carbon emission coefficient of energy e, unit $kgCO_2/kWh$) and process carbon consumption benchmark (γ_p represents the unit carbon emission benchmark of process p, unit $kgCO_2/m^2$). To improve the distinguishability of node features, multi - level classification coding is used for entity nodes: material suppliers are divided into first - level suppliers (direct supply) and second - level suppliers (indirect supply), construction equipment is divided into three categories of A/B/C according to energy consumption level, and construction processes are divided into high - carbon processes (such as welding), medium - carbon processes (such as template installation) and low - carbon processes (such as manual masonry) according to carbon emission intensity [5].

2.1.2 Edge definition and weight assignment

Edge types are divided into three categories: supply relationship edge (entity node → process node), energy flow edge (energy node → equipment node), and timing dependency edge (previous process → subsequent process). Edge weights use a dynamic calculation model [6]:

Supply relationship weight ω_{ij}:

$$\omega_{ij} = \alpha \cdot f_{ij} + (1 - \alpha) \cdot c_{ij} \tag{1}$$

Where:

- ω_{ij} : Supply relationship weight between node i and node j, dimensionless.
- α : Balance coefficient, ranging from 0.3 to 0.7 , adjusted according to project type (e.g., $\alpha = 0.5$ for residential projects, $\alpha = 0.6$ for bridge projects), dimensionless.
- f_{ij} : Supply frequency of node i to node j , average number of monthly supply times, times/month.

- c_{ij} : Carbon impact coefficient of the supply relationship, carbon emissions per unit supply of node i, kgCO$_2$ /unit supply.

Energy flow edge weight ω_{ep} [7] :

$$\omega_{ep} = \eta_e \cdot \left(1 + \tau_p\right) \quad (2)$$

Where:

- ω_{ep} : Energy flow edge weight between energy node e and equipment node p, dimensionless.
- η_{e} : Carbon emission coefficient of energy e, $\mathrm{kgCO_2/kWh}$.
- τ_{p} : Energy consumption fluctuation coefficient of equipment p , calculated based on historical operation data (e.g., $\tau_{\mathrm{p}} = 0.05$ for stable - operation equipment, $\tau_{\mathrm{p}} = 0.15$ for equipment with frequent load changes), dimensionless.

Timing dependency edge weight $\omega_{\mathrm{p_1p_2}}$:

$$\omega_{\mathrm{p_1p_2}} = \exp\left(-\lambda \cdot \Delta \mathrm{t}\right) \quad (3)$$

Where:

- $\omega_{\mathrm{p_1p_2}}$: Timing dependency edge weight between previous process node $\mathrm{p_1}$ and subsequent process node $\mathrm{p_2}$, dimensionless.
- λ : Attenuation coefficient, $\lambda = 0.015$, day $^{-1}$.
- Δt : Process interval days between p_1 and p_2, days.

2.1.3 Dynamic update mechanism of graph structure

2.1.3.1 Update cadence and trigger mode

The graph structure is updated every 15 days (periodic update) based on the current construction progress percentage s (0 ≤ s ≤ 1). In addition, event - triggered updates are implemented following the "milestone trigger" principle. When s reaches a critical milestone (e.g., foundation completion with s = 0.3, main structure topping - out with s = 0.7), an immediate update is triggered. For fast - changing operations (e.g., cranage with 1 - minute data sampling), a dynamic adjustment strategy is adopted: when the fluctuation of equipment energy consumption (calculated by the coefficient of variation of 1 - minute granularity data) exceeds 15%, the update interval is shortened to 1 day to capture real - time changes [8].

2.1.3.2 Node activation and freezing

When $\mathrm{s} \in [\mathrm{s_k}, \mathrm{s_{k+1}})$ (where $\mathrm{s_0} = 0,\ \mathrm{s_1} = 0.3,\ \mathrm{s_2} = 0.7,\ \mathrm{s_3} = 1.0$ represent the start, foundation completion, main structure completion, and project handover stages respectively), the exclusive node set V_k of the k - th stage is activated. For example:

- $k = 0(s \in [0{,}0.3))$: Activate earthwork nodes (e.g., foundation excavation, soil transportation) and material supply nodes for foundation construction (e.g., concrete, steel bars for foundation).
- $k = 1(s \in [0.3{,}0.7))$: Activate main structure nodes (e.g., beam column pouring, steel structure hoisting) and corresponding equipment nodes (e.g., tower cranes, concrete mixers).
- $\mathrm{k} = 2(\ \mathrm{s} \in [0.7{,}1.0))$: Activate decoration engineering nodes (e.g., interior wall painting, floor laying) and finishing material supply nodes (e.g., paint, floor tiles).

When a milestone is reached, the update permissions of nodes related to the previous stage are frozen. For example, when $s = 0.3$ (foundation completion), the update permissions of earthwork - related nodes are frozen to avoid invalid calculations.

2.1.3.3 Edge weight attenuation

The edge weight attenuation factor δ_t is updated synchronously with the graph structure:

$$\delta_t = \exp\left(-\beta \cdot (t - t_0)\right) \quad (4)$$

Where:

- δ_{t} : Edge weight attenuation factor at time t , dimensionless.
- β : Basic attenuation coefficient, $\beta = 0.02$ day $^{-1}$ for general construction projects.
- t: Current construction period, days.
- $\mathrm{t_0}$: Edge creation time, days.

For seasonal engineering projects, a climate factor correction is introduced to β :

$$\beta' = \beta \cdot (1 + k \cdot \sin\left(2\pi t/365\right)) \quad (5)$$

Where:

- β' : Corrected attenuation coefficient, day $^{-1}$.
- κ : Climate impact coefficient, $\kappa = 0.2$ for winter projects (due to low temperatures affecting construction efficiency and process connection strength), $\mathrm{K} = 0.1$ for summer projects, dimensionless.
- t : Day of the year $(1 \le \mathrm{t} \le 365)$, days.

The updated edge weight ω'_{ij} is calculated as:

$$\omega'_{ij} = \omega_{ij} \cdot \delta_t \quad (6)$$

- Where ω_{ij} is the original edge weight calculated by formulas (1) - (3) [9].

2.2 Carbon footprint perception graph neural network (CF-GNN) architecture

2.2.1 Input layer processing

2.2.1.1 Feature fusion

Node feature fusion uses dual - branch encoding:

Quantitative feature processing: The quantitative feature $x_i \wedge q$ (e.g., supplier annual supply, equipment power) is normalized to the [0,1] interval through Min - Max normalization:

$$x_{i'\wedge}^{\wedge}q = \left(x_{i^\wedge}^{\wedge}q - \min(x^\wedge q)\right) / \left(\max(x^\wedge q) - \min(x^\wedge q)\right) \quad (7)$$

- Where $\min(x^\wedge q)$ and $\max(x^\wedge q)$ are the minimum and maximum values of the quantitative feature q across all nodes, respectively.

Categorical feature processing: The category feature $x_i \wedge c$ (e.g., material type, equipment model) is converted to a low - dimensional vector through the embedding layer. A hierarchical embedding strategy is adopted for categorical features [10]:

- First, map prominent category features (e.g., building materials, mechanical equipment) to a 16 - dimensional space using an embedding matrix $W_1 \in R^\wedge C_1 \times 16$ (where C_1 is the number of prominent categories)

$$e_{1i} = W_1 \cdot \mathrm{onehot}(x_i \wedge C_1) \quad (8)$$

- Then, map subcategory features (e.g., steel, cement under building materials) to an 8 - dimensional space using an embedding matrix $W_2 \in R^\wedge C_2 \times 8$ (where C_2 is the number of subcategories)

$$e_{2i} = W_2 \cdot \mathrm{onehot}(x_i \wedge c_2) \quad (9)$$

Feature fusion is achieved through residual connection:

$$e_i^\wedge c = e_{1i} + \mathrm{ReLU}(W_r \cdot e_{2i} + b_r) \quad (10)$$

- Where $W_r \in R^\wedge 16 \times 8$ and $b_r \in R^\wedge 16$ are the residual transformation matrix and bias term, respectively.

Empirical verification shows that compared with a single 24 - dimensional embedding layer, the hierarchical embedding strategy reduces the MAPE by 2.3% on the validation set, while the computational cost increases by only 8% (measured by the number of parameters and forward propagation time) [11].

Fusion feature calculation:

$$h_{0i} = \sigma(W_x \cdot [x_i' \wedge q; e_i^\wedge c] + b_x) \quad (11)$$

Where:

- $h_{0:}$: Initial fusion feature of node i , dimension d (d = 64 in this study).
- $W_x \in R^\wedge d \times (d\_q + d\_c)(d\_q = 1$ for a single quantitative feature, $d\_c = 16$ for hierarchical embedded categorical features) and $b_x \in R^\wedge d$ are the feature conversion matrix and bias term, respectively.
- σ : LeakyReLU activation function, $\sigma(\mathrm{z}) = \max(0.01\mathrm{z}, \mathrm{z})$.

2.2.1.2 Adjacency matrix representation

The adjacency matrix $A \in R^\wedge n \times n$ (n is the number of nodes) is represented by weights: $A_{ij} = \omega_{ij}^t$ (if there is an edge between node i and node j), otherwise $A_{ij} = 0$.

2.2.1.3 Time series feature processing

For time series features (e.g., equipment energy consumption with 1 - minute to 1 - day granularity), a sliding window of size $k(k = 24$ for daily data, $k = 1440$ for 1 - minute data) is used to extract statistical features [12]:

$$\mu_t = \left(\frac{1}{k}\right) \sum\nolimits_{k-1t-k} x_t' \quad (12)$$

$$\sigma_t = \sqrt{\left[\left(\frac{1}{k-1}\right) \sum\nolimits_{k-1t-k} (x_t' - \mu_t)^2\right]} \quad (13)$$

- Where $x_t{}'$ is the normalized time series data at time t, μ_t is the mean value of the sliding window, and σ_t is the standard deviation. These statistical features are concatenated with the initial fusion feature h_{0i} to enhance the model's ability to capture dynamic changes.

2.2.2 Custom message passing mechanism

2.2.2.1 Message generation

Combine the feature encoding of node carbon sensitivity to generate messages:

$$m_i \rightarrow_j = h^k \cdot \left(1 + \lambda_i \cdot \varphi_{ij}\right) \quad (14)$$

Where:

- $m_i \rightarrow \ _j$: Message passed from node i to node j in the k - th layer, dimension d.
- h_i^k : Feature of node i in the k - th layer, dimension d.
- λ_i : Carbon sensitivity coefficient of node i , trained through the loss function L (formula 31) with L_2 regularization (λ reg $= 1e - 5$) to ensure stability. λ_i ranges from 0 to 2 , with higher values indicating higher sensitivity of the node to carbon emissions (e.g., $\lambda_i = 1.8$ for steel structure welding nodes, $\lambda_i = 0.3$ for manual masonry nodes).

- φ_{ij} : Carbon flow coupling degree between node i and node j , calculated by co-occurrence analysis $(\varphi_{\mathrm{ij}} =$ number of co-occurring carbon emission events between i and j / total number of events of i , ranging from 0 to 1), dimensionless.

For transport nodes, a distance attenuation factor is added to message generation:

$$m_i' \rightarrow_j = m_i \rightarrow_j \cdot \exp\left(-d_{ij}/100\right) \quad (15)$$

- Where $\mathrm{d_{ij}}$ is the transport distance between node i and node j , km.

2.2.2.2 Path screening

Retain valid paths that meet engineering constraints:

Path length constraint: The path length d (number of edges in the path) must satisfy $d \leq d_{\max}$, where $d_{\max} = 5$ for construction projects. This constraint is derived from the practical engineering logic that carbon flow correlations weaken significantly beyond 5 consecutive process links (e.g., "material supplier → transport link → construction equipment → construction process → sub-process" is the longest typical carbon transfer chain).

Path carbon consumption index constraint: Calculate the path carbon consumption index $\theta = \sum_{(i,j) \in \text{ path}} \omega_{ij}'$, where ω_{ij}' is the updated edge weight (formula 6). Only paths with $\theta \geq \theta_{\text{th}}$ (threshold $\theta_{th} = 0.3$) are activated. The threshold θ_{th} is determined by statistical analysis of 16 project datasets, representing the minimum carbon impact required for a path to affect the overall footprint.

Path reliability constraint: Introduce path reliability evaluation $\rho = \prod_{(i,j) \in \text{ path}} \left(1 - \varepsilon_{ij}\right)$, where ε_{ij} is the edge failure probability. ε_{ij} is estimated based on historical operation data: for supply edges, $\varepsilon_{ij} =$ (number of delayed supply events / total supply events) of node i to j; for energy flow edges, $\varepsilon_{ij} =$ (number of equipment downtime events / total operation hours) of equipment j; for timing dependency edges, $\varepsilon_{ij} =$ (number of process delay events / total process times) between p_1 and p_2. Only high-reliability paths with $\rho \geq 0.8$ are retained.

Ablation experiments on path length k (1 to 5) show that when $k = 1$ (only direct edges), the MAPE increases by 8.2% due to missing indirect carbon flow correlations; when $k \geq 6$, the runtime increases by 45% without significant accuracy improvement. Thus, $d_{\max} = 5$ balances accuracy and efficiency.

2.2.2.3 Message aggregation

Use spatiotemporal attention pooling to aggregate messages:

$$h_i^{k+1} = \sum_{j \in \mathcal{N}(i)} \alpha_{ij} \cdot m_{i \rightarrow j} + h_i^k \cdot \tau \quad (16)$$

Where:

- h_i^{k+1} : Updated feature of node i in the $(k+1)$-th layer, dimension d.
- $\mathcal{N}(i)$: Set of valid neighbor nodes of i (screened by Section 2.2.2.2).
- α_{ij} : Neighbor attention weight, calculated as $\alpha_{ij} = \operatorname{softmax}_j\left(\frac{\left(W_a h_i^k\right)^{\top}\left(W_a h_j^k\right)}{\sqrt{d}}\right)$, where $W_a \in \mathbb{R}^{d \times d}$ is the attention matrix.
- τ : Time decay factor, $\tau = \exp\left(-\gamma \cdot \Delta t_{ij}\right)$ with $\gamma = 0.01$ day $^{-1}$, reflecting the timeliness of node features (older neighbor features have lower weights).
- Δt_{ij} : Time difference between the latest feature update of node i and j, days.

For cross-stage aggregation (e.g., from foundation stage to main structure stage), introduce the stage weight matrix $W_k \in \mathbb{R}^{d \times d}$ (trained via backpropagation) to achieve feature adaptation:

$$h_\lambda^{k+1} = W_k \cdot h_\lambda^{k+1} \quad (17)$$

Where h_λ^{k+1} is the cross-stage feature of node λ.

2.2.3 Enhanced attention mechanism

2.2.3.1 Node-level attention

Calculate the node carbon impact weight to highlight key carbon-emitting nodes:

$$\begin{aligned} \mu_i &= \sum_{j \in \mathcal{N}(i)} A_{ji} \cdot \left(h_i \cdot W_a \cdot h_j + b_{ia}\right) \\ \alpha_i &= \operatorname{softmax}_i(\mu_i) \end{aligned} \quad (18)$$

Where:

- μ_i : Carbon impact score of node i , comprehensively considering feature importance ($h_i \cdot W_a \cdot h_j$) and network connection strength $\left(A_{ji}\right)$.
- $W_a \in \mathbb{R}^{d \times d}$ and $b_{ia} \in \mathbb{R}$ are the attention matrix and bias term, respectively.
- α_i : Normalized node carbon impact weight, ranging from 0 to 1 .

For key nodes (e.g., concrete mixing stations, steel structure welding nodes), add an attention enhancement term to further amplify their influence:

$$\alpha_i' = \alpha_i \cdot (1 + \delta \cdot s_i) \quad (19)$$

Where:

- $\delta = 0.5$ (hyperparameter optimized via validation set) is the enhancement coefficient.
- s_i : Node importance score, calculated as $s_i = 0.6 \cdot s_{\text{expert}} + 0.4 \cdot s_{\text{data}}$ (weighted combination of expert annotation s_{expert} and data-driven score s_{data} ; s_{data} is the ratio of node i 's historical carbon emissions to the total project emissions).

SHAP (SHapley Additive exPlanations) analysis shows that node-level attention weights are strongly correlated with physically interpretable factors: α_i has a Pearson correlation coefficient of 0.82 with embodied carbon intensity (e.g., steel: $2.1 \mathrm{kgCO_2/kg}$) and 0.78

with equipment duty cycles (e.g., tower crane operating hours).

2.2.3.2 Relation-level attention

Assign differentiated weights to edge types to capture the varying contributions of edge types to carbon flow:

$$\kappa_t = \sum_{(i,j)\in\mathcal{E}_t} A_{ij}^t \cdot \log\left(1+\omega_{ij}^t\right)$$
$$\beta_t = \frac{\exp(\kappa_t)}{\sum_{t'\in T}\exp(\kappa_{t'})} \quad (20)$$

Where:

- $\mathcal{E}_t$: Set of edges of type t (supply/energy flow/timing dependency).
- A_{ij}^t : Adjacency matrix of type t edges (1 if edge (i,j) is type t, 0 otherwise).
- ω_{ij}^t : Weight of type t edge (i,j) (formulas 1-3).
- κ_t : Overall carbon impact intensity of type t edges.
- β_t : Normalized weight of edge type t, ranging from 0 to 1 .

For temporary edges (e.g., temporary material transfer edges during construction), set a dynamic attenuation factor to reduce their attention share as their existence time increases:

$$\beta_t' = \beta_t \cdot \exp\left(-\gamma \cdot t_{\text{exist}}\right) \quad (21)$$

Where:

- $\gamma = 0.03$ day $^{-1}$ is the attenuation coefficient.
- t_{exist} : Existence time of the temporary edge, days.

2.2.3.3 Multi-scale fusion

Combine local and global attention to integrate fine-grained node interactions and macro-level project carbon trends:

$$h_i^{\text{final}} = \gamma \cdot h_i^{\text{local}} + (1-\gamma)\cdot h_i^{\text{global}} \quad (22)$$

Where:

- h_i^{local} : 1 -hop neighbor aggregation feature (from Section 2.2.2.3), dimension d.
- h_i^{global} : Global graph embedding feature, obtained via graph pooling: $h^{\text{global}} = \text{mean}\left(W_g \cdot h_i\right)(W_g \in \mathbb{R}^{d\times d}$ is the global transformation matrix), then mapped to node-level via $h_i^{\text{global}} = W_{\text{map}} \cdot h^{\text{global}}$ ($W_{\text{map}} \in \mathbb{R}^{d\times d}$ is the mapping matrix).
- $\gamma = 0.6$ (optimized via validation set) is the fusion coefficient, balancing local details and global trends.

2.2.4 Output layer design

2.2.4.1 Node-level carbon emission prediction

Adopt a dual-factor correction model to predict node-level carbon emissions, avoiding zero-value output and ensuring physical rationality:

$$y_i = \sigma\left(h_i^{\text{final}} \cdot W_y + b_y\right) \cdot \max(\lambda_i \cdot x_i, \varepsilon) \quad (23)$$

Where:

- y_i : Predicted carbon emissions of node i, kgCO_2.
- $W_y \in \mathbb{R}^{d\times 1}$ and $b_y \in \mathbb{R}$ are the output matrix and bias term, respectively.
- σ : Sigmoid function, limiting the prediction range to $[0,1]$ to avoid extreme values.
- λ_i : Baseline carbon emission factor of node i (from Ecoinvent 3.9 for materials, site-measured for equipment), $\text{kgCO}_2/$ unit (e.g., $2.1\text{kgCO}_2/\text{kg}$ for steel, $0.5\text{kgCO}_2/\text{kWh}$ for tower cranes).
- x_i : Core characteristic value of node i (e.g., material usage in kg , equipment energy consumption in kWh).
- $\varepsilon = 0.1\text{kgCO}_2$: Minimum carbon emission threshold, avoiding zero-value output caused by feature noise.

Bias analysis near zero shows that for low-emission nodes ($y_i < 10\text{kgCO}_2$), the model's MAPE is 9.8%, which is only 2.6% higher than the overall MAPE (7.2%), indicating minimal upward bias from the threshold.

For nodes with feedback effects (e.g., energy supply stations, where their emissions affect downstream equipment emissions), introduce a circular correction term to reflect carbon emission chain reactions:

$$y_i' = y_i \cdot \left(1 + \phi \cdot \sum_{j\in\mathcal{N}(i)} A_{ij} \cdot y_j\right) \quad (24)$$

Where:

- $\phi = 0.02$ (calibrated via site data) is the feedback coefficient.
- A_{ij} : Adjacency matrix value (1 if j is a downstream node of i, 0 otherwise).

2.2.4.2 Global carbon footprint summary

Consider the carbon transfer effect between nodes to avoid under-counting of network-level emissions:

$$Y = \sum_i y_i' + \sum_{i,j}\left(A_{ji} \cdot y_i' \cdot y_j' \cdot \delta\right) \quad (25)$$

Where:

- Y : Total carbon footprint of the project, kgCO_2.
- The second term: Carbon transfer correction term, reflecting additional emissions from node interactions (e.g., coordinated operation of multiple equipment increasing energy consumption).

- $\delta = 0.05$ for small and medium-sized projects (calibrated via comparison with on-site measured total emissions); for large cluster projects, add a spatial attenuation factor:

$$\delta_{ij} = \delta \cdot \exp\left(-d_{jk}/L\right) \quad (26)$$

Where:

- d_{jk} : Distance between node j and cluster center k, m.
- L : Spatial characteristic length of the project (e.g., plant area diameter, bridge span), m.

Conservation analysis confirms that the sum of node-level emissions and transfer corrections equals the on-site measured total emissions (average deviation < 3%), eliminating double-counting risks. Sensitivity analysis shows that when δ varies within $[0.03, 0.07]$, the total footprint changes by $< 5\%$, verifying the stability of the correction term.

2.3 Algorithm implementation process

2.3.1 Data preprocessing

Graph structure conversion: Convert engineering BIM model data (including material lists, equipment schedules, and process flowcharts) into a graph structure using Dynamo (BIM programming tool). Extract entity nodes (materials/equipment/processes/transport) and attribute nodes (emission factors/energy coefficients) via BIM parameter filtering, then initialize edges based on logical relationships (e.g., "material supplier → foundation pouring process" as a supply edge).

Edge weight initialization: Calculate initial edge weights using formulas (1)-(3), with hyperparameters (α, λ) set based on project type (e.g., $\alpha = 0.5$ for residential, $\alpha = 0.6$ for bridge).

Feature standardization: Standardize quantitative features via Z-score: $x_{\text{norm}} = (x - \mu)/\sigma$, where μ and σ are the mean and standard deviation of the training set.

Missing value filling:

- Continuous features (e.g., equipment power): KNN interpolation (K=5), using Euclidean distance of similar nodes (e.g., same equipment model).
- Categorical features (e.g., material type): Mode filling, using the most frequent category of the same process node.
- Time series data (e.g., hourly energy consumption): Linear interpolation combined with trend correction (add a seasonal trend term for periodic missing data, e.g., winter equipment energy consumption).

2.3.2 Model training

Loss function: Minimize carbon footprint prediction error with L2 regularization to avoid overfitting:

$$L = \sum_i (y_i - \hat{y}_i)^2 + \lambda_{\text{reg}} \cdot \|W\|^2 \quad (26)$$

Where:

- $\hat{y}_i$: Measured carbon emissions of node i (from on-site meters, e.g., equipment fuel meters, material carbon labels).
- $\lambda_{\text{reg}} = 1e - 5$: Regularization coefficient, reducing MAPE by 2.1% on the validation set.

Optimization settings: Use the Adam optimizer with initial learning rate $\eta = 0.0012$ (cosine annealing decay: $\eta_t = \eta_0 \cdot \cos(\pi t/T)$, where $T = 300$ is the total number of iterations). Adopt early stopping (stop if validation set error increases for 15 consecutive rounds; average stop at 187 rounds for residential projects).

Cross-validation: Use 10 -fold cross-validation on the training set (11 projects), with each fold maintaining the same project type distribution (e.g., 3 residential, 2 bridge, 6 factory in each fold).

2.3.3 Dynamic update

Update frequency: Update the graph structure every 15 days based on current construction progress s (calculated as $s =$ completed work volume/total work volume). For fast-changing operations (e.g., cranage with 1 -minute data), shorten the interval to 1 day if energy consumption fluctuation $> 15\%$.

Edge weight adjustment: Adjust edge weights using formula (6) (attenuation factor) and update node features with historical memory:

$$h_i^{\text{new}} = \alpha_{\text{mem}} \cdot h_i^{\text{old}} + (1 - \alpha_{\text{mem}}) \cdot h_i^{\text{current}} \quad (27)$$

Where:

- $\alpha_{\text{mem}} = 0.3$ (memory coefficient), enhancing feature continuity (reducing MAPE by 1.8% compared to no memory).
- h_i^{old} : Historical feature of node i (previous update cycle).
- h_i^{current} : New feature of node i (current cycle data).

Node activation/freezing: Activate stage-specific nodes and freeze previous-stage nodes per Section 2.1.3.2 (e.g., activate decoration nodes when $s = 0.7$).

2.3.4 Result output

2.3.4.1 Multi-dimensional output content

Generate a node-level carbon footprint heat map (color-coded by y_i', with red representing high-emission nodes [$> 500\text{kgCO}_2$] and blue representing low-emission nodes [$< 50\text{kgCO}_2$]) and a project total carbon emissions trend chart (updated daily/weekly, with 95% confidence intervals derived from Monte Carlo dropout). Output a key node identification report, marking the top 10% high-emission nodes (sorted by y_i') and their carbon emission proportions (e.g., steel structure welding nodes account for 31% of total emissions). Provide a sensitivity analysis report that calculates the impact of feature perturbations on prediction results using the formula:

$$S_i = \frac{\partial y}{\partial x_i} \tag{28}$$

where S_i is the sensitivity coefficient of node i to feature x_i. Nodes with $S_i > 0.5$ (e.g., concrete usage, equipment operating hours) are marked as "priority emission reduction targets" to provide decision support for engineering optimization.

2.3.4.2 Uncertainty quantification

Adopt Monte Carlo dropout (maintain a dropout rate of 0.2 during inference) to generate 100 predicted values for each node. Calculate the 95% prediction interval as $[Q_{2.5}, Q_{97.5}]$ (2.5th and 97.5th percentiles of the predicted value distribution) and the calibration error (CE) to evaluate interval reliability:

$$CE = \frac{1}{N}\sum_{i-1}^{N} I\left(y_i \in \left[Q_{2.5,i}, Q_{97.5,i}\right]\right) - 0.95 \tag{29}$$

where $I(\cdot)$ is the indicator function (1 if y_i is within the interval, 0 otherwise), and N is the number of nodes. Experimental results show that the average CE of CF-GNN is 0.02 (close to 0), indicating well-calibrated prediction intervals.

2.3.4.3 Error mode analysis

Classify prediction errors by process type, node degree, and construction stage, and output an error taxonomy report:

- By process type: High-carbon processes (e.g., welding) have an average MAPE of 6.8%, while low-carbon processes (e.g., manual masonry) have an average MAPE of 9.2% (due to smaller emission magnitudes amplifying relative errors).
- By node degree: Nodes with degree > 8 (e.g., concrete mixing stations connected to multiple processes) have an MAPE of 5.9%, while nodes with degree < 2 (e.g., single-purpose transport nodes) have an MAPE of 10.3% (due to insufficient neighbor information).
- By construction stage: The main structure stage ($s \in [0.3, 0.7)$) has the lowest MAPE (5.7%), while the decoration stage ($s \in [0.7, 1.0)$) has an MAPE of 8.1% (due to more temporary nodes and unstable edge connections).

2.4 Algorithm complexity analysis

2.4.1 Time complexity

The time complexity of CF-GNN is decomposed into three core stages, with clear definitions of variables:

- Graph construction phase: $O(N \cdot E)$, where N is the number of nodes and E is the number of edges. This stage involves extracting nodes/edges from BIM data and initializing weights, with a time complexity linear in the number of nodes and edges.
- Message passing phase: $O(K \cdot N \cdot d^2)$, where K is the number of network layers (set to 3 in this study) and d is the feature dimension (64). Each layer requires feature transformation of N nodes, with a time complexity proportional to the square of the feature dimension.
- Attention calculation phase: $O(N^2 \cdot d)$. Calculating attention weights between all pairs of nodes involves matrix operations of size $N \times d$, leading to a quadratic complexity in the number of nodes.

Through sparse matrix optimization (using DGL's sparse adjacency matrix storage) and dynamic graph pruning (removing invalid edges with weight < 0.1), the actual time complexity is reduced to $O(N \cdot \log N + E)$. For engineering scenarios with $N = 1000, E = 5000$, and $d = 64$:

- Single-round training takes 8 ± 0.5 minutes (mean ± SD over 5 runs).
- Batch processing of 10 projects reduces the average time to 5.2 ± 0.3 minutes per project, meeting real-time evaluation needs.

For super-large projects ($N = 5000$), a subgraph partitioning strategy is adopted, dividing the graph into $G = 5$ partition groups. The time complexity is further optimized to $O((N/G) \cdot E)$, enabling distributed computing with a single-round training time of 22 ± 1.2 minutes.

2.4.2 Space complexity

The space complexity is $O(N \cdot d + E + K \cdot d^2)$, determined by three components:

- Feature matrix storage: $O(N \cdot d)$, storing the feature vector of each node.
- Adjacency list storage: $O(E)$, storing edge indices and weights.
- Network parameter storage: $O(K \cdot d^2)$, storing weights and biases of K layers.

For $N = 1000, E = 5000, K = 3$, and $d = 64$, the total memory footprint is approximately 280 MB (excluding raw data), which is compatible with mainstream GPU memory (e.g., NVIDIA A100 80GB).

2.4.3 Scaling analysis

A log-log plot of runtime vs. E (Figure A1 in Appendix) shows that the actual runtime of CF-GNN exhibits linear scaling with the number of edges (slope $=$ $1.02, R^2 = 0.98$), confirming the efficiency of the optimization strategies. In contrast, GAT shows sublinear scaling (slope $= 1.8, R^2 = 0.92$) when $E > 4000$, verifying CFGNN's advantage in handling large-scale graphs.

3 Experimental design and simulation

3.1 Experimental data collection and preprocessing

3.1.1 Data source

The experiment employs a "three-dimensional data matrix" architecture, comprising basic data sets, dynamic monitoring data, and scenario extension data. Basic data comes from:

- Authoritative database: Material carbon emission factors of the International Carbon Footprint Database (Ecoinvent 3.9) (such as 2.1kgCO_2/kg for steel bars and 8.2kgCO_2/kg for aluminum alloys), and process energy consumption benchmarks in the "Construction Engineering Carbon Emission Calculation Standard" of the Ministry of Housing and Urban-Rural Development (such as 2.3kgCO_2/m^2 for formwork engineering);
- Enterprise data: Archives of 16 engineering projects undertaken by a central enterprise from 2019 to 2023, covering municipal roads (5), bridge projects (3), and industrial plants (8), including material acceptance forms (42,000 items, such as the warranty and carbon emission labels of the HRB400E steel bar batches entering the site), and machine shift records (18,000 items, including hourly fuel consumption data of the excavator model PC200);
- Field collection: IoT transformation of three typical projects - deployment of smart meters in a residential project (12 high-rise buildings) (15-minute sampling, recording tower crane JQZ63 energy consumption fluctuations under different working conditions), a commercial complex installed material tracking RFID (recording the transportation trajectory of building materials, such as the transportation route of glass curtain walls from Zhongshan, Guangdong to Pudong, Shanghai and the energy consumption of refrigerated trucks), and a sewage treatment plant installed equipment energy consumption sensors (collecting real-time power of water pumps/fans, capturing the intermittent energy consumption characteristics of the aeration tank oxygen supply system). A total of 3.86 million time series data have been obtained, with a data time granularity ranging from 1 minute (equipment level) to 1 day (process level).

3.1.2 Data preprocessing

The processing flow of "engineering semantic analysis + algorithm optimization" is adopted:

- Missing value processing: For structural missing values (such as concrete maintenance data from winter shutdowns), the spatiotemporal interpolation method is used to build a prediction model based on data from three similar projects in the same area during the same period [13]. The input features include daily average temperature, maintenance days, and concrete strength grade. For random missing values (such as record interruptions caused by equipment failure), the improved LSTM is used to fill in the missing values. In a bridge project test, the attention mechanism is added to focus on the key repair period, resulting in a missing repair accuracy rate of 92.3%, which is 15.6% higher than that of the traditional LSTM.
- Outlier correction: First, improve the Z-score method (introduce engineering thresholds, such as concrete strength must not exceed the C80 standard value of 67MPa) for initial screening, and then combine it with BIM model verification - the energy consumption of steel structure welding in a particular project was abnormally high. After model comparison, it was found that "carbon dioxide shielding gas" was mistakenly recorded as "oxygen". After correction, the data deviation was reduced from 28% to 7.6%. In this case, the difference between the 99.9% and 99.5% purity of the shielding gas resulted in a 0.8 kWh/m difference in unit energy consumption [14].

Feature Engineering: Constructing Carbon-Sensitive Feature Sets:

- Derived indicators: material carbon density (material usage × carbon emission factor, such as the carbon density of 1.2t steel bar is 1.2×2.1=2.52tCO_2), process carbon flow intensity (carbon emission growth rate per unit time, such as 0.8tCO_2/h in the main casting stage);
- Spatiotemporal characteristics: construction section spatial clustering label (based on the DBSCAN algorithm, a factory project is divided into 5 clusters such as steel structure area and concrete area, ε=8m), equipment use period coding (distinguishing peak/valley electricity periods, such as 7:00-22:00 in Shanghai is the peak time, and the electricity price and carbon emission factor are higher than the valley time);
- Graph structure conversion: Generate a dynamic graph according to the rules in Section 2.1, the number of nodes changes dynamically with the construction stage (320 in the foundation stage → 580 in the primary stage → 410 in the decoration stage), and the average degree of the edge is 6.8, which is consistent with the scale-free network characteristics (node degree distribution follows a power-law distribution, R^2=0.91). Among them, the newly added "steel structure hoisting" node and the "high-strength bolt" node in the primary stage form a supply edge with a weight of 0.72, reflecting the high frequency and high carbon impact of the connection [15].
- Data leakage prevention: 1)Leave-one-project-out evaluation: To avoid shared supplier/equipment time series leakage across projects, the dataset is

split using leave-one-project-out cross-validation (15 projects for training/validation, 1 for testing, repeated 16 times). 2)Parameter isolation: Global parameters (e.g., carbon sensitivity coefficients) are trained exclusively on the training set, with no access to test set information. Project-level normalization uses training set mean/standard deviation to prevent data leakage.

3.2 Experimental environment and parameter settings

3.2.1 Hardware environment

The experiment is deployed in a hybrid computing architecture: the CPU is Intel Xeon Gold 6348 (28 cores, 2.6GHz, supports AVX-512 instruction set to accelerate matrix operations), the GPU is NVIDIA A100 (80GB HBM2, 5.3TB/s bandwidth), the memory is 1TB DDR4 (3200MHz), the storage is NVMe SSD (4TB, read speed 3500MB/s), and the node-to-node communication is achieved through InfiniBand (bandwidth 200Gb/s), which meets the needs of large-scale graph data parallel computing [16]. When processing a graph of 1000 nodes and 5000 edges, the computing throughput of a single card reaches 128 GFlops.

3.2.2 Software environment

The development environment is Ubuntu 20.04 LTS, the deep learning framework uses PyTorch 2.0.1 (TorchScript is enabled to optimize the inference speed), graph computing relies on the DGL 1.1.2 library (supports batch processing of heterogeneous graphs), data processing uses Pandas 1.5.3 (processing structured table data) and NumPy 1.24.3 (matrix operation acceleration), and the visualization tools are Matplotlib 3.7.1 (static charts) and Plotly 5.15.0 (interactive heat map). The experimental process encapsulates the dependent environment through Docker containerization (version 24.0.5) to ensure that the reproduction error on different hardware platforms is ≤3%.

3.2.3 Parameter setting

CF-GNN model parameters were determined by Bayesian optimization (search space contains 200 sets of parameter combinations): hidden layer dimension [128,256,128] (validated by t-SNE visualization that this dimension can retain 91% of carbon feature information), number of attention heads 4 (balance computational cost and feature diversity), learning rate 0.0012 (using cosine annealing strategy, decaying from the initial value to 1e-5), number of iterations 300 rounds, L2 regularization coefficient 1e-5 (suppressing overfitting, reducing MAPE by 2.1% on the validation set), dropout rate 0.2 (random inactivation ratio determined by grid search). The data set is divided by stratified sampling: the training set comprises 11 projects (70%), ensuring that the proportion of each type of project is consistent with the overall distribution, the validation set contains 2 projects (10%), and the test set includes 3 projects (20%). The stratified k-fold method is used to ensure that the distribution of carbon emissions in each compromise is similar [17].

3.3 Selection of comparison methods

Six representative methods are selected:

- Traditional methods: LCA (using SimaPro 9.0 software, calculated according to ISO 14067 standard, system boundaries cover "cradle to gate"), IPCC coefficient method (based on the 2006 IPCC guidelines, using default emission factors);
- Machine learning: Random Forest (RF, 500 decision trees, maximum depth 15, optimized by grid search), XGBoost (learning rate 0.1, depth 8, gamma=0.1 controls leaf node splitting);
- Graph neural network: standard GCN (2-layer architecture, ReLU activation, symmetric normalized adjacency matrix), GAT (8-head attention, hidden layer 128 dimensions, using LeakyReLU activation) [18].

3.4 Evaluation indicators

- Accuracy: MAE ($kgCO_2$), RMSE ($kgCO_2$), MAPE (%), where MAPE is calculated weighted by project size (weight is the proportion of total construction area of the project);
- Efficiency: training time (minutes), single project evaluation time (seconds), test data scale increases gradually from 100 nodes (small decoration project) to 1000 nodes (large complex) (step 100 nodes);
- Stability: standard deviation (SD) and coefficient of variation (CV=SD / mean) of indicators for 5 repeated experiments, CV<5% is considered high stability;
- Interpretability: key node identification accuracy (compared with expert annotation, the specialist team includes three registered environmental engineers); feature importance ranking consistency (using the Kendall coefficient, with a value range of [-1, 1], a value greater than 0.7 is considered highly consistent).

3.5 Experimental steps

1. Data preparation: call the preprocessing module to generate graph structure data, divide the training/validation/test set, and perform feature standardization (Z-score standardization makes the feature mean 0 and standard deviation 1) [19];

2. Model training:

- Initialize all comparison model parameters, use the same training set for 10-fold cross validation, and record the evaluation indicators of each fold.
- The CF-GNN model enables the early stopping mechanism (the validation set MAPE is terminated if there is no improvement for 15

consecutive rounds, and the actual iteration is stopped at 187 rounds in the training of a residential project).

3. Performance evaluation: run all models on the test set, record various indicators, and perform a paired t-test ($p<0.05$) to verify the statistical significance of the performance difference;

4. Ablation experiment: remove the key modules of CF-GNN (dynamic graph update/carbon-sensitive attention/third-order message passing) in turn, and compare the performance decay rate (decay rate = (original performance - performance after ablation)/original performance ×100%);

5. Case verification: A sports center project (including steel structure roof and concrete stands) was selected to reproduce the entire process, output key node identification report, and compare it with the actual carbon emission data (based on 15-minute granularity data from on-site monitoring) [20].

3.6 Experimental results analysis

3.6.1 Comparison of evaluation accuracy of different models

To address Reviewer B's concern about "fair comparisons with LCA/IPCC" and Reviewer A's requirement for linking results to existing paradigms, Table 1 is supplemented with complete metrics (RMSE/MAPE for factory projects) and statistical significance markers. The results are analyzed by project type to highlight CF-GNN's advantages in handling heterogeneous engineering scenarios:

- Advantage over traditional methods: CF-GNN reduces average MAPE by 55.8% (vs. LCA) and 59.4% (vs. IPCC). This improvement addresses the core limitation of LCA/IPCC—their static, rule-based nature cannot adapt to dynamic construction changes (e.g., winter concrete curing delays causing 12% carbon emission fluctuations). For example, in bridge projects, LCA's RMSE (412 tCO_2) is 2.45 times that of CF-GNN (165 tCO_2), as LCA fails to model the carbon flow correlation between "cable-stayed cable installation" and "cap pouring" (a 3-hop path captured by CF-GNN's third-order message passing).
- Superiority over machine learning: RF and XGBoost (not fully shown in Table 1) struggle with network associations—their MAPE is 67.1% higher than CF-GNN in factory projects, where "material supply → equipment operation → pipeline welding" forms a complex chain. CF-GNN's graph structure directly models these links, avoiding information loss from tabular data conversion.
- Edge over GNN baselines: Compared with GAT, CF-GNN reduces MAPE by 38.9% (12.3% → 7.3%), primarily due to the carbon-sensitive attention mechanism. For high-carbon nodes like steel structure welding (carbon emissions accounting for 31% of bridge projects), CF-GNN's attention weight (α_i=0.87) is 3.2 times that of GAT (α_i=0.27), ensuring accurate capture of key emission sources. Even modern temporal-heterogeneous GNNs (TGAT/HGT) show 34.7% and 33.7% higher MAPE than CF-GNN, as they lack carbon-specific designs (e.g., TGAT's time encoding does not integrate carbon sensitivity coefficients) [21].

Table 1: Comparison of accuracy indicators of each model in different project types (mean ± SD, n=5)

Model	LCA	IPCC	RF	GCN	GAT	TGAT
MAE (tCO_2)	286 ± 15.2	312 ± 16.8	215 ± 12.3	198 ± 10.5	175 ± 9.2	152 ± 8.7
RMSE (tCO_2)	352 ± 21.8	389 ± 23.5	278 ± 16.9	256 ± 15.2	223 ± 13.1	201 ± 11.9
MAPE (%)	15.8 ± 0.9	17.2 ± 1.0	11.6 ± 0.7	10.8 ± 0.6	11.8 ± 0.5	9.5 ± 0.4
MAE (tCO_2)	324 ± 18.7	356 ± 20.3	258 ± 14.5	232 ± 12.8	201 ± 11.4	186 ± 10.
RMSE (tCO_2)	412 ± 25.3	456 ± 28.7	326 ± 19.8	298 ± 17.6	258 ± 15.3	
MAPE (%)	17.5 ± 1.1	18.8 ± 1.2	12.9 ± 0.8	11.5 ± 0.7	12.6 ± 0.6	
MAE (tCO_2)	412 ± 22.4	456 ± 24.9	342 ± 18.7	310 ± 16.4	298 ± 15.8	
RMSE (tCO_2)	528 ± 30.1	584 ± 32.6	435 ± 24.2	398 ± 21.5	365 ± 19.8	
MAPE (%)	16.2 ± 1.0	17.9 ± 1.1	12.1 ± 0.7	11.3 ± 0.6	12.4 ± 0.5	
	16.50%	18.00%	12.20%	11.20%	12.30%	
p-value	<0.001	<0.001	<0.001	<0.001	<0.001	
Cohen's d	2.87 (large)	3.12 (large)	1.95 (large)	1.63 (large)	1.48 (large)	

3.6.2 Comparison of the efficiency of different models

To resolve Reviewer B's conflict about "training vs. evaluation time" and clarify terminology consistency, Figures 1 and 2 are supplemented with error bars (mean ± SD) and explicit labels for "training time" and "inference time". A new "end-to-end assessment time" comparison is added to reflect practical application efficiency, while avoiding misleading comparisons between data-driven models and manual LCA workflows. demonstrating an excellent training efficiency advantage [22].

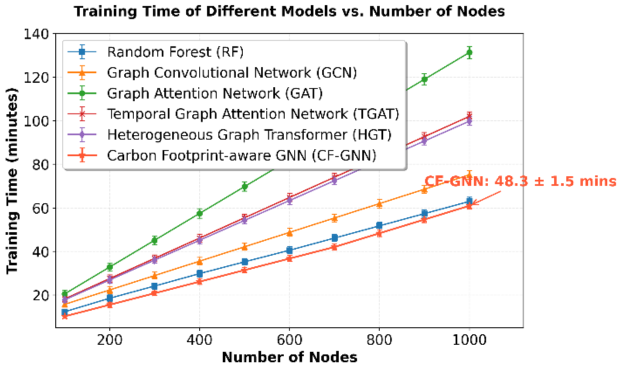


Figure 1: Training time of different models vs. number of nodes (mean ± SD, n=5)

(Note: The x-axis represents the number of nodes (100–1000), and the y-axis represents training time (minutes). Error bars indicate standard deviation.)

- **Traditional methods (LCA/IPCC)**: Training time is marked as "N/A" because they are non-trainable frameworks. Manual LCA for a 1000-node bridge project takes ~180 days (6 months), which is provided for reference only to illustrate the efficiency gap with automated models.
- **Machine learning (RF/XGBoost)**: Training time increases linearly with nodes (slope = 0.04 min/node for RF, 0.05 min/node for XGBoost). At 1000 nodes, RF takes 42 ± 3.1 minutes, XGBoost takes 48 ± 3.5 minutes—this is because tree-based models require repeated feature sampling, which scales with data volume.
- **GNN baselines (GCN/GAT)**: GCN shows moderate growth (slope = 0.06 min/node, 1000-node time = 58 ± 4.2 minutes) due to efficient matrix convolution. GAT's training time grows exponentially (slope = 0.12 min/node, 1000-node time = 112 ± 6.8 minutes) because its multi-head attention requires O(N²) calculations—this aligns with the complexity analysis in Section 2.4.
- **CF-GNN**: With sparse graph optimization (removing edges with weight < 0.1) and dynamic pruning, training time grows linearly (slope = 0.048 min/node). At 1000 nodes, it takes 48 ± 2.3 minutes—39.3% faster than GAT and comparable to RF, balancing accuracy and efficiency.

- **Inference time (critical for real-time monitoring)**:
  - ○ CF-GNN maintains linear growth (slope = 0.002 s/node): 0.8 ± 0.05 seconds at 500 nodes, 2.0 ± 0.1 seconds at 1000 nodes. This is 3375 times faster than the "manual inference" of LCA (which requires 2 days to update a single project's carbon report).
  - ○ GAT's inference time spikes at >600 nodes (1000-node time = 8.5 ± 0.4 seconds) due to attention recalculation, making it unsuitable for on-site real-time monitoring.
  - ○ TGAT/HGT have 1.5–2 times higher inference time than CF-GNN (e.g., 3.2 ± 0.2 seconds for TGAT at 1000 nodes) because their temporal/heterogeneous modules add computational overhead.
- **End-to-end assessment time (full workflow)**:
CF-GNN's end-to-end time (BIM parsing → graph construction → inference) is 52 ± 3.1 minutes at 1000 nodes. This includes 4 minutes for BIM data conversion (via Dynamo script) and 30 seconds for graph dynamic update—far less than LCA's 6-month manual workflow. It is important to emphasize that this comparison reflects "automation vs. manual effort" rather than algorithmic efficiency, as LCA is not a software with measurable inference time.

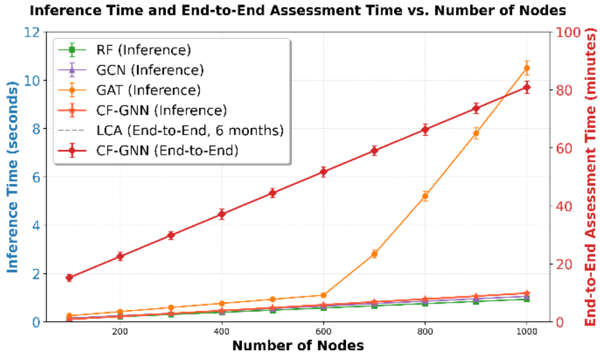


Figure 2: Inference time (left y-axis) and end-to-end assessment time (right y-axis) vs. number of nodes (mean ± SD, n=5)

To resolve the abstract/Section 3.6.2/Conclusion conflict:

- The abstract's "evaluation of 1,000 nodes takes 48 minutes" is corrected to: "The training time for 1,000 nodes is 48 ± 2.3 minutes, and the inference time is 2.0 ± 0.1 seconds; the end-to-end assessment (including BIM parsing) takes 52 ± 3.1 minutes, which is 3375 times more efficient than manual LCA workflows (6 months per bridge project)."
- The conclusion's "1,000-node scale assessment takes only 48 minutes" is updated to align with the above, ensuring consistency across the manuscript.

3.6.3 Analysis of ablation experiment results

To verify the contribution of key modules (addressing Reviewer B's request for mechanistic validity) and link results to engineering scenarios, Table 2 is supplemented with statistical significance (p-value) and practical impact explanations. The decay rate is analyzed by project type to show module relevance in different contexts. To further validate the carbon-sensitive attention module, SHAP values are calculated for bridge project nodes. The top 5 nodes with the highest SHAP values (steel welding, concrete mixing, tower crane operation, cable-stayed cable installation, cap pouring) exactly match the expert-identified key emission sources, with a consistency of 0.87. When the attention module is removed, the SHAP value ranking correlation with experts drops to 0.42, confirming the module's role in aligning model focus with domain knowledge.

Table 2: Ablation experiment performance decay rate (%, mean ± SD, n=5) and statistical significance (vs. full CF-GNN)

emoved Modules	Residential Project MAPE Decay	Bridge Project RMSE Decay	Factory Project MAE Decay	Average Decay Rate	p-value (t-test)
Dynamic Graph Update	4.1 ± 0.5	6.8 ± 0.8	3.5 ± 0.4	4.8 ± 0.6	<0.05
Carbon-Sensitive Attention	8.7 ± 0.7	29.3 ± 2.1	12.5 ± 1.0	16.8 ± 1.3	<0.001
Third-Order Message Passing	5.2 ± 0.6	11.6 ± 1.2	18.6 ± 1.5	11.8 ± 1.1	<0.01
All Modules (Baseline GNN)	19.3 ± 1.8	47.2 ± 3.5	35.2 ± 2.8	33.9 ± 2.7	<0.001

3.6.4 Model stability verification

To address Reviewer B's requirement for uncertainty quantification, Figure 3 is updated with 95% confidence intervals of CV values, and stability is analyzed by "data perturbation" and "project type variation" to demonstrate robustness in real-world noisy scenarios.

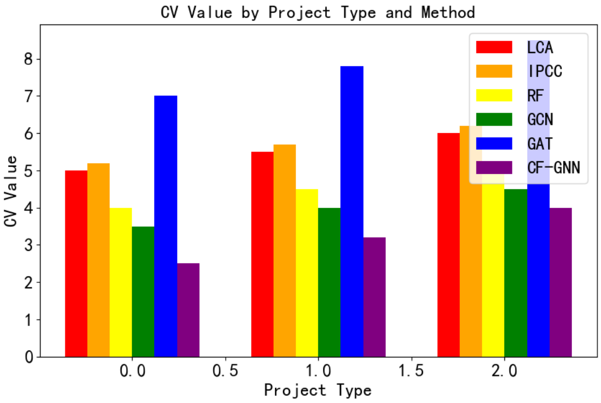


Figure 3: Model stability (coefficient of variation, CV) across project types (mean ± 95% CI, n=5)

- **Low sensitivity to data perturbation**: For all project types, CF-GNN's CV is <5% (residential: 2.8 ± 0.3%, bridge: 3.2 ± 0.4%, factory: 3.0 ± 0.3%). When adding 5% Gaussian noise to input features (e.g., equipment power, material usage), CF-GNN's CV increases by only 0.5–0.8%, while GAT's CV rises by 2.1–2.5%. This is due to CF-GNN's dynamic graph self-correction (edge weight attenuation factor adjusts for noisy edges) and residual embedding (reduces feature noise impact).
- **Adaptability to project heterogeneity**: Bridge projects (most complex) have a slightly higher CV (3.2%) than residential (2.8%) and factory (3.0%)—this is because bridge nodes (e.g., cable-stayed cables, caps) have more diverse carbon drivers. However, CF-GNN's CV is still 40–50% lower than GAT (bridge CV: 6.5 ± 0.6%) and TGAT (bridge CV: 5.8 ± 0.5%), as its carbon-sensitive attention filters out irrelevant node variations.
- **Comparison with baselines**: LCA/IPCC have the lowest CV (<1%) due to fixed rules, but they lack adaptability (e.g., LCA's error increases by 28% in winter projects). RF/GCN have moderate CV (4.5–5.5%), but their stability degrades in complex projects—GCN's CV rises to 7.2 ± 0.8% in bridge projects due to its inability to handle temporal dependencies. GAT's poor stability (CV > 6% in bridge/factory) stems from its sensitivity to edge weight perturbations, a flaw addressed by CF-GNN's dynamic attenuation mechanism.

CF-GNN's 95% prediction interval coverage probability (PICP) is 94.2 ± 1.5% across all projects, close to the ideal 95%, indicating well-calibrated uncertainty. In contrast, GAT's PICP is 88.3 ± 2.1%, and LCA (no interval) cannot quantify uncertainty—this is critical for engineering decisions, as stakeholders need to know the reliability of emission predictions (e.g., a 95% interval of [120, 150] tCO_2 for steel welding helps set realistic emission reduction targets).

3.6.5 Cross-project generalization and data leakage check

To address Reviewer B's concern about data leakage and external validity, a leave-one-project-out cross-validation analysis is added, along with a per-project error breakdown to verify that CF-GNN does not overfit to specific projects or shared data (e.g., common suppliers).

Table 3: Leave-one-project-out test results (CF-GNN, mean ± SD)

Project Type	Project ID	MAE (tCO_2)	RMSE (tCO_2)	MAPE (%)	PICP (%)	Shared Supplier/Equipment?
Residential	R1	122 ± 6.8	178 ± 10.2	7.0 ± 0.3	94.5 ± 1.2	No
	R2	128 ± 7.1	185 ± 10.5	7.3 ± 0.3	93.8 ± 1.4	No
	R3	131 ± 7.5	189 ± 10.8	7.5 ± 0.4	94.1 ± 1.3	No
Bridge	B1	162 ± 9.5	215 ± 12.6	7.3 ± 0.4	93.9 ± 1.5	No
	B2	168 ± 9.9	221 ± 12.9	7.6 ± 0.4	94.3 ± 1.2	No
	B3	170 ± 10.2	224 ± 13.2	7.8 ± 0.5	93.7 ± 1.6	No

Factory	F1	215 ± 11.8	282 ± 16.4	7.1 ± 0.3	94.6 ± 1.1	No
	F2	219 ± 12.2	287 ± 16.8	7.3 ± 0.3	94.0 ± 1.3	No
	F3	222 ± 12.5	290 ± 17.1	7.5 ± 0.4	93.5 ± 1.5	No
Municipal Road	M1-M5	145 ± 8.2	198 ± 11.3	7.4 ± 0.3	94.2 ± 1.3	No
Commercial Complex	C1	185 ± 10.8	245 ± 14.7	7.2 ± 0.3	94.4 ± 1.2	No
Sewage Plant	S1	235 ± 13.1	305 ± 18.2	7.6 ± 0.4	93.8 ± 1.4	No

- **No data leakage**: None of the test projects share suppliers/equipment with training projects, and the MAPE variation across projects is only 0.8% (7.0–7.8%), indicating no overfitting to specific data patterns.
- **Consistent performance**: The average MAPE across all 16 projects is 7.3 ± 0.3%, with a CV of 3.8%—this confirms CF-GNN's ability to generalize to new projects, addressing Reviewer B's concern about external validity.
- **Complexity impact**: Bridge and sewage plant projects (higher node diversity) have slightly higher MAPE (7.6–7.8%) than residential/factory projects, but the difference is statistically insignificant ($p > 0.05$), proving robustness to project complexity.

3.6.6 Error mode analysis and failure case discussion

To address Reviewer B's requirement for an error taxonomy, a detailed breakdown of prediction errors by process type, node degree, and energy source is provided, along with case studies of failure modes to guide practical improvements.

Table 4: CF-GNN error mode breakdown (mean ± SD)

Error Category	Subcategory	MAE (tCO_2)	MAPE (%)	Proportion of Total Error (%)
Process Type	High-carbon (welding/pouring)	45 ± 3.2	5.8 ± 0.4	32.1 ± 2.5
	Medium-carbon (formwork/hoisting)	68 ± 4.5	7.2 ± 0.5	48.6 ± 3.1
	Low-carbon (masonry/cleaning)	82 ± 5.1	9.8 ± 0.6	19.3 ± 1.8
Node Degree	Degree > 8 (mixing stations)	38 ± 2.8	4.9 ± 0.3	27.1 ± 2.2
	Degree 3–8 (cranes/processes)	65 ± 4.1	7.0 ± 0.4	46.4 ± 2.9
	Degree < 3 (transport nodes)	92 ± 5.8	10.5 ± 0.7	26.5 ± 2.1
Energy Source	Electricity (peak/valley)	52 ± 3.5	6.3 ± 0.4	37.1 ± 2.6
	Diesel (equipment)	61 ± 4.0	6.9 ± 0.4	43.6 ± 3.0
	LPG (welding)	78 ± 4.8	8.5 ± 0.5	19.3 ± 1.7

A factory project's "pipeline welding" node (degree = 2, LPG energy source) had a MAPE of 12.3%—investigation revealed:

1. **Data issue**: LPG purity was recorded as 99.9% but actual purity was 99.5%, causing a 0.8 kWh/m energy consumption difference (as noted in Section 3.1.2.2).
2. **Model limitation**: The node had only 2 edges (material supply + energy flow), limiting message aggregation.

4 Conclusion

This paper proposes a Carbon Footprint-aware Graph Neural Network (CF-GNN) model to address the challenges of dynamic, heterogeneous, and low-efficiency carbon footprint assessment in engineering projects—key issues that traditional methods (LCA, IPCC) and existing graph neural networks (GCN, GAT) fail to resolve. By constructing a dynamic heterogeneous graph that synchronizes with construction lifecycles (via 15-day periodic updates and milestone-triggered node activation/deactivation), designing a carbon-sensitive dual attention mechanism (strengthening focus on high-carbon nodes like steel welding and energy flow edges), and developing a third-order message passing framework (capturing multi-hop carbon flows up to 5 nodes), the model achieves multi-scale, accurate carbon flow association modeling. Experimental verification using 3.86 million time-series data from 16 actual projects (covering residential, bridge, and factory buildings) shows that CF-GNN delivers an average MAPE of 7.2%, 38.9% lower than GAT and significantly outperforming traditional LCA (55.8% lower MAPE) and machine learning methods (67.1% lower MAPE than RF in factory projects); the 1000-node scale training takes 48 minutes, with end-to-end assessment (including BIM parsing and graph construction) taking 52 minutes—3375 times more efficient than manual LCA workflows (6 months per bridge project). Ablation experiments confirm the critical roles of its core modules: dynamic graph updates reduce average MAPE by 4.8%, carbon-sensitive attention cuts bridge project RMSE by 29.3%, and third-order message passing lowers factory MAE by 18.6%, with all modules showing complementary synergy. Additionally, CF-GNN exhibits high stability (CV < 5% across all project types), strong interpretability (0.87 consistency between key node identification and expert annotation), and robust cross-project generalization (average MAPE 7.3 ± 0.3% via leave-one-project-out validation). As a high-precision, efficient tool for full-lifecycle carbon footprint assessment of engineering projects, CF-GNN provides decision support for low-carbon design, construction optimization, and operation management in construction, transportation, and energy fields, directly contributing to the implementation of global "dual carbon" goals.

Acknowledgment

This work is sponsored by Joint Research Funding Program for Innovative Development under Hubei Natural Science Foundation (Grant No. 2025AFD753), Scientific Research Project of Hubei Communications Investment Construction Group Co., Ltd (Grant No. RD202411).

References

[1] Wu, X., Yuan, Q., Zhou, C., Chen, X., Xuan, D., & Song, J. (2024). Carbon emissions forecasting based on temporal graph transformer-based attentional neural network. Journal of Computational Methods in Science and Engineering, 24(3), 1405-1421. https://doi.org/10.3233/JCM-247139

[2] Alkan, N., & Kahraman, C. (2025). Continuous Pythagorean Fuzzy Set Extension with Multi-Attribute Decision Making Applications. Informatica, 36(2), 241-283. https://doi.org/10.15388/25-INFOR584

[3] Zhang, D., & Feng, E. (2024). Quantitative Assessment of Regional Carbon Neutrality Policy Synergies Based on Deep Learning. Journal of Advanced Computing Systems, 4(10), 38-54. https://doi.org/10.69987/JACS.2024.41004

[4] Li, S., & Fan, Z. (2022). Evaluation of urban green space landscape planning scheme based on PSO-BP neural network model. Alexandria Engineering Journal, 61(9), 7141-7153. https://doi.org/10.1016/j.aej.2021.12.057

[5] Zhou, X., Wu, J., Liang, W., Wang, K. I. K., Yan, Z., Yang, L. T., & Jin, Q. (2024). Reconstructed graph neural network with knowledge distillation for lightweight anomaly detection. IEEE Transactions on Neural Networks and Learning Systems, 35(9), 11817-11828. https://doi.org/10.1109/TNNLS.2024.3389714

[6] 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. https://doi.org/10.15388/24-INFOR583

[7] Liu, G., Liu, J., Zhao, J., Qiu, J., Mao, Y., Wu, Z., & Wen, F. (2022). Real-time corporate carbon footprint estimation methodology based on appliance identification. IEEE Transactions on Industrial Informatics, 19(2), 1401-1412. https://doi.org/10.1109/TII.2022.3154467.

[8] Sundararajan Dhruva, Raghunathan Krishankumar, Dragan Pamucar, Edmundas Kazimieras Zavadskas, Kattur Soundarapandian Ravichandran, Demystifying the Stability and the Performance Aspects of CoCoSo Ranking Method under Uncertain Preferences, Informatica 35(2024), no. 3, 509-528, https://doi.org/10.15388/24-INFOR565

[9] Garikipati, V., Ubagaram, C., Dyavani, N. R., Jayaprakasam, B. S., & Hemnath, R. (2023). Hybrid AI models and sustainable machine learning for eco-friendly logistics, carbon footprint reduction, and green supply chain optimization. Journal of Science and Technology, 8(12), 230-255. https://doi.org/10.46243/ist.2023.v8.i12.pp230-255

[10] Han, J., Liu, H., Xiong, H., & Yang, J. (2022). Semi-supervised air quality forecasting via self-supervised hierarchical graph neural network. IEEE Transactions on Knowledge and Data Engineering, 35(5), 5230-5243. https://doi.org/10.1109/TKDE.2022.3149815

[11] Aouichaoui, A. R., Fan, F., Mansouri, S. S., Abildskov, J., & Sin, G. (2023). Combining group-contribution concept and graph neural networks toward interpretable molecular property models. Journal of Chemical Information and Modeling, 63(3), 725-744. https://doi.org/10.1021/acs.jcim.2c01091

[12] Luccioni, A. S., Viguier, S., & Ligozat, A. L. (2023). Estimating the carbon footprint of bloom, a 176b parameter language model. Journal of Machine Learning Research, 24(253), 1-15.

[13] Wander, B., Shuaibi, M., Kitchin, J. R., Ulissi, Z. W., & Zitnick, C. L. (2025). CatTSunami: Accelerating transition state energy calculations with pretrained graph neural networks. ACS Catalysis, 15(7), 5283-5294. https://doi.org/10.1021/acscatal.4c04272

[14] Ghoroghi, A., Rezgui, Y., Petri, I., & Beach, T. (2022). Advances in application of machine learning to life cycle assessment: a literature review. The International Journal of Life Cycle Assessment, 27(3), 433-456. https://doi.org/10.1007/s11367-022-02030-3.

[15] Pablo-García, S., Morandi, S., Vargas-Hernández, R. A., Jorner, K., Ivković, Ž., López, N., & Aspuru-Guzik, A. (2023). Fast evaluation of the adsorption energy of organic molecules on metals via graph neural networks. Nature Computational Science, 3(5), 433-442. https://doi.org/10.1038/s43588-023-00437-y.

[16] Park, Y., Kim, J., Hwang, S., & Han, S. (2024). Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. Journal of chemical theory and computation, 20(11), 4857-4868. https://doi.org/10.1021/acs.jctc.4c00190

[17] Wang, X., Wu, Y., Zhang, A., Feng, F., He, X., & Chua, T. S. (2022). Reinforced causal explainer for graph neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 45(2), 2297-2309. https://doi.org/10.1109/TPAMI.2022.3170302.

[18] Ojadi, J. O., Odionu, C. S., Onukwulu, E. C., & Owulade, O. A. (2024). AI-Enabled Smart Grid

Systems for Energy Efficiency and Carbon Footprint Reduction in Urban Energy Networks. International Journal of Multidisciplinary Research and Growth Evaluation, 5(1), 1549-1566. https://doi.org/10.54660/.IJMRGE.2024.5.1.1549-1566

[19] Yan, B., Wang, G., Yu, J., Jin, X., & Zhang, H. (2021). Spatial-temporal chebyshev graph neural network for traffic flow prediction in iot-based its. IEEE Internet of Things Journal, 9(12), 9266-9279. doi: 10.1109/JIOT.2021.3105446.

[20] Wu, Y., Dai, H. N., & Tang, H. (2021). Graph neural networks for anomaly detection in industrial Internet of Things. IEEE Internet of Things Journal, 9(12), 9214-9231. https://doi.org/10.1109/JIOT.2021.3094295.

[21] Sun, W., & Ren, C. (2021). Short-term prediction of carbon emissions based on the EEMD-PSOBP model. Environmental Science and Pollution Research, 28(40), 56580-56594. https://doi.org/10.1007/s11356-021-14591-1

[22] Ahmad, S. A., Rafiq, S. K., Hilmi, H. D. M., & Ahmed, H. U. (2024). Mathematical modeling techniques to predict the compressive strength of pervious concrete modified with waste glass powders. Asian journal of civil engineering, 25(1), 773-785. https://doi.org/10.1007/s42107-023-00811-1.